

Sergey Subbotin
Editor

Computer Modeling and Intelligent Systems

Proceedings of The Seventh International Workshop
CMIS-2024



Zaporizhzhia, Ukraine
May 3, 2024

Subbotin S., (Ed.): The Seventh International Workshop on Computer Modeling and Intelligent Systems (CMIS-2024). Zaporizhzhia, Ukraine, May 3, 2024, CEUR-WS.org, online

This volume represents the proceedings of the Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2024), held in Zaporizhzhia, Ukraine, in May, 2024. It comprises 40 contributed papers that were carefully peer-reviewed and selected from 60 submissions.

Copyright © 2024 for the individual papers by the papers' authors.
Copying permitted only for private and academic purposes.
This volume is published and copyrighted by its editors.

Preface

It is my pleasure to present you the proceedings of The Seventh International Workshop on Computer Modeling and Intelligent Systems (CMIS-2024), held in Zaporizhzhia, Ukraine on 3 of May 2024.

The CMIS workshop was held in 2024 for the seventh time at the National University "Zaporizhzhia Polytechnic" (<http://www.zp.edu.ua>), which is the largest and oldest regional center for science and higher education in the Zaporizhzhia region of South-Eastern Ukraine.

The main purpose of the CMIS-2024 workshop is a discussion of the recent research results in all areas of Computer Sciences and Information Technologies.

The workshop is soliciting literature review, survey and research papers including, whilst not limited to, the following areas of interest:

- Artificial Intelligence;
- Computer Science Methods for Modeling;
- Info-Communication Technologies for Data Analysis and Applications for Engineering, Medicine and Education.

There CMIS-2024 workshop countries of PC members and authors of submitted papers are: Azerbaijan, Germany, India, Israel, Kazakhstan, Kyrgyzstan, Morocco, Netherlands, Pakistan, Poland, Portugal, Slovak Republic, Spain, Ukraine, United States of America, and Uzbekistan. The number of countries of CMIS-2024 authors and PC members is 16.

The language of CMIS-2024 Workshop is English. The workshop took the form of oral presentations of peer-reviewed regular papers. The video of presentations is available at <https://www.facebook.com/groups/cmishorkshop/>.

The CMIS-2024 Organizing Committee have received 60 paper submissions, out of which 40 were accepted for presentation as a regular papers in the result of reviewing process, where the editor, program committee members, external and peer reviewers sought to ensure a high scientific level, as well as a high-quality presentation of the selected papers. These papers were published in this volume of CMIS-2024 proceedings.

The workshop would not have been possible without the support of many people. I would like to thank the Workshop Program Committee Members, organization staff, and external reviewers for their excellent and tireless work. I sincerely wish that all attendees benefited scientifically from the workshop and wish them every success in their research. It is the humble wish of the workshop organizers that the professional dialogue among the researchers, scientists, and engineers continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

May, 2024

Sergey Subbotin, Dr. Sc., Prof.,
CMIS-2024 PC Chair and Proceedings Editor
Head of the Department of Software Tools,
National University "Zaporizhzhia Polytechnic"

Programme Committee

Chair

Prof. **Sergey Subbotin**, National University "Zaporizhzhia Polytechnic", Ukraine

Members

Prof. **Gustavo Alves**, University of Porto, Portugal

Prof. **Yevgeniy Bodyanskiy**, Kharkiv National University of Radio Electronics, Ukraine

Prof. **Karsten Henke**, Ilmenau University of Technology, Germany

Prof. **Igor Korobiichuk**, Warsaw University of Technology, Poland

Prof. **Vitaly Levashenko**, University of Zilina, Slovak Republic

Prof. **David Luengo**, Universidad Politecnica de Madrid, Spain

Prof. **Andrii Oliinyk**, National University "Zaporizhzhia Polytechnic", Ukraine

Prof. **Nataliya Pankratova**, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine

Prof. **Mykhailo Poliakov**, National University "Zaporizhzhia Polytechnic", Ukraine

Prof. **Anatoliy Sachenko**, Western-Ukrainean National University, Ukraine

Prof. **Natalya Shakhivska**, Lviv Polytechnic National University, Ukraine

Prof. **Galyna Tabunshchyk**, National University "Zaporizhzhia Polytechnic", Ukraine

Prof. **Elena Zaitseva**, University of Zilina, Slovak Republic

Dr. **Ivan Izonin**, Lviv Polytechnic National University, Ukraine

Dr. **Serhii Leoshchenko**, National University "Zaporizhzhia Polytechnic", Ukraine

Dr. **Anzhelika Parkhomenko**, National University "Zaporizhzhia Polytechnic", Ukraine

Dr. **Alexei Sharpanskykh**, Delft University of Technology, Netherlands

Dr. **Thomas (Tom) Trigano**, Shamoon College of Engineering, Israel

Dr. **Heinz-Dietrich Wuttke**, Ilmenau University of Technology, Germany

Organizers



National University "Zaporizhzhia Polytechnic",
Zaporizhzhia, Ukraine

<http://www.zp.edu.ua>



Department of Software Tools
of the National University "Zaporizhzhia Polytechnic",
Zaporizhzhia, Ukraine

<http://pz.zntu.net>

Local Organizational Committee

Prof. **Sergey Subbotin**, National University "Zaporizhzhia Polytechnic", Ukraine

Ph.D **Serhii Leoshchenko**, National University "Zaporizhzhia Polytechnic", Ukraine

Eng **Maksym Andreiev**, National University "Zaporizhzhia Polytechnic", Ukraine

Partners

Co-funded by the
Erasmus+ Programme
of the European Union



WORK4CE

International project "Cross-domain competences for healthy and safe work in the 21st century"(WORK4CE) of Erasmus+ Programme co-funded by the European Union (Ref. No. 619034-EPP-1-2020-1-UA-EPPKA2-CBHE-JP)

DAAD



ViMaCs

Virtual Master Cooperation
Data Science

International project "Virtual Master Cooperation Data Science" (ViMaCs) of DAAD

DAAD

EU-ViMUK

EuroPIM Virtual
Master School Ukraine

International project "EuroPIM Virtual Master School Ukraine" (EU-ViMUK) of DAAD

Reviewers

Name	Email	Number of reviews
Alexey Yurko	yurkoalexe@gmail.com	4
Alina Shafronenko	alina.shafronenko@nure.ua	4
Alina Yanko	al9_yanko@ukr.net	4
Andrey Kupin	kupin.andrew@gmail.com	4
Andrii Nyzhnyk	andrij1510@gmail.com	4
Andrii Tkachuk	andru_tkachuk@ukr.net	4
Andriy Goncharenko	andygoncharenco@yahoo.com	4
Anna Kozlova	ak.koann@gmail.com	1
Artem Andreiev	a.a.andreev@casre.kiev.ua	1
Artem Trukhov	artem.trukhov@gmail.com	4
Ayman Aljarbough	ayman.aljarbough@ucentralasia.org	4
Denys Khanin	denys.o.khanin@lpnu.ua	4
Dmytro Fedasyuk	dmytro.v.fedasyuk@lpnu.ua	4
Dmytro Zubov	dzubov@ieee.org	4
Eugene Fedorov	fedorovee75@ukr.net	4
Igor Povkhan	igor.povkhan@uzhnu.edu.ua	4
Ihor Pilkevych	igor.pilkevich@meta.ua	4
Ihor Prots'Ko	ihor.o.protsko@lpnu.ua	4
Inna Polishchuk	inna.polishchuk@uzhnu.edu.ua	1
Ivan Dyyak	ivan.dyyak@lnu.edu.ua	4
Ivan Muzyka	musicvano@gmail.com	4
Ivan Opirskyy	iopirsky@gmail.com	4
Khrystyna Lipianina-		
Honcharenko	xrustya.com@gmail.com	3
Maksym Boiko	mboiko25@gmail.com	2
Marina Polyakova	marinapolyakova943@gmail.com	4
Miroslav Kelemen	miroslav.kelemen@tuke.sk	4

Name	Email	Number of reviews
Moskalenko Vasiliovich	v.moskalenko@cs.sumdu.edu.ua	3
Mykola Romanchuk	romannik@ukr.net	4
Nataliia Zanko	zankon@i.ua	4
Nazarii Burak	nazar.burak@ukr.net	4
Nickolay Rudnichenko	nickolay.rud@gmail.com	4
Oksana Mulesa	Oksana.Mulesa@uzhnu.edu.ua	4
Oleg Bisikalo	obisikalo@gmail.com	3
Oleh Pihnastyi	pihnastyi@gmail.com	3
Oleh Vietrov	o.vietrov@donnu.edu.ua	4
Oleksandr Zhukov	oleksanrei@gmail.com	3
Oleksandra Bulgakova	sashabulgakova2@gmail.com	4
Oleksii Vodka	oleksii.vodka@gmail.com	4
Olena Naumchak	olenanau@gmail.com	4
Olga Nechyporenko	olne@ukr.net	4
Olga Oliynyk	oleinik_o@ukr.net	4
Ruslan Yakovliev	director.klk.hnuvs@gmail.com	4
Sergiy Prykhodko	sergiy.prykhodko@nuos.edu.ua	8
Serhii Vladov	ser26101968@gmail.com	4
Tetiana Neskorodieva	tvnesk1@gmail.com	4
Tetiana Vakaliuk	tetianavakaliuk@gmail.com	4
Tetyana Marusenkova	tetyana.marus@gmail.com	4
Valentyn Yanchuk	v.yanchuk@gmail.com	4
Valerii Dudykevych	valerii.b.dudykevych@lpnu.ua	4
Viacheslav Zosimov	zosimovvv@gmail.com	4
Victor Krasnobayev	v.a.krasnobaev@gmail.com	3
Victoria Vysotska	victoria.a.vysotska@lpnu.ua	4
Vira Shendryk	v.shendryk@cs.sumdu.edu.ua	4
Vladimir Vuchuzhanin	icst_nuop@ukr.net	4
Volodymyr Polishchuk	v.polishchuk87@gmail.com	4

Name	Email	Number of reviews
Volodymyr Rusyn	rusyn_v@ukr.net	3
Vyacheslav Moskalenko	v.moskalenko@cs.sumdu.edu.ua	1
Yaroslav Dorogyy	yaroslav.dorohyi@donntu.edu.ua	4
Yuliia Parfenenko	yuliya_p@cs.sumdu.edu.ua	2
Yuriy Lakh	yurii.v.lakh@lpnu.ua	4

Sponsors

Andreev Artem Andriyovych
Boyko Maksym Volodymyrovych
Epic Maryna Oleksandrivna
Horbenko Vitaly Ivanovich
Kupin Andriy Ivanovich
Lakh Yuriy Volodymyrovych
Martsenyuk Yevhenii Vitaliyovych
Maslova Nataliya Oleksandrivna
Naumchak Olena Mykhaylivna
Nemkova Olena Anatoliivna
Nyzhnyk Andriy Olegovich
Pilkevich Ihor Anatoliyovych
Polishchuk Volodymyr Volodymyrovych
Rudenko Diana Oleksandrivna
Semenov Maria Rudolphivna
Tanyanskyi Oleksiy Serhiyovych
Ursul Ivan Zinoviovych
Vychuzhanin Oleksii Volodymyrovych
Zhukov Oleksandr Oleksiyovych

Medical Subsystem for Blood Tests Evaluation Based on E-commerce Solution

Tetiana A. Vakaliuk^{1,2,3}, Valentyn Yanchuk¹, Andrii Tkachuk¹ and Anna Humeniuk¹

¹ Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine

² Institute for Digitalization of Education of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine

³ Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine

Abstract

Cloud computing, in combination with distributed data aggregation, helps humankind solve the problem of data collection and processing near the source of collection, supporting the possibility of distributed systems processing data in the nearest node and further aggregating the data into bigger sources for normalisation and application of the data models and data statistical processing. Medical data are well known for their exceeding sizes and huge dependencies on specific needs; however, applying contemporary models, combining different subsystems, and applying analytic approaches help resolve serious automation problems. Data aggregation, combined with data processing and visualisation, enables medical business owners to decide how to plan the budget for the upcoming period. The study foresees the usage of data collection, processing, and visualisation of medical test data and the projection of wasted percentages with the feedback loop for service providers integrated with an e-commerce solution. The system composition, lifecycle, and process transitions are developed according to the business requirements of the companies and their sales, returns, waste analysis documents, and corresponding processes. Data-receiving models and transformations are mentioned but not described due to commercial secrets, which do not impact the approach of system development and can be applied to similar or other industries.

Keywords

Processing at Cloud Services, distributed data aggregation, medical subsystems, interoperability

1. Introduction

The current society has many challenges related to the medical and healthcare systems, which constantly change in the context of reform. However, medical servicing systems had a serious hit of activity related to the COVID-19 pandemic [1], which forced people to rethink their concept of interaction with medical institutions completely. With the number of patients exponentially increasing in the pandemic, the healthcare system requested utterly new approaches for logistics and support of medical institutions from the perspective of medical tests, the readiness of staff, and medical equipment to process an increasing number of patients and patients' medical records. At that very moment, the vast majority of medical institutions noticed the urgent need for modernisation of their laboratory bases, and more laboratory networks came forward and served as certified centres of blood sampling and processing for medical institutions. It enabled the organisations to support more patients, enforce consultancy rather than concentrate on laboratory investigations to make expert conclusions on each patient's case, reference the disease's course, and validate the treatment results [2]. The supply chain of such organisations is still based on e-commerce solutions of specific types that can work with enterprise resource planning ERP systems. Besides standard solutions for e-commerce solutions satisfying the requirements of standard users, we see more and more e-commerce applications creating additional subsystems to automate specific fields of solutions where e-commerce is being applied.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ tetianavakaliuk@gmail.com (T. A. Vakaliuk); v.yanchuk@gmail.com (V. Yanchuk); andru_tkachuk@ukr.net (A. Tkachuk); gum_ann@ztu.edu.ua (A. Humeniuk)

ORCID iD 0000-0001-6825-4697 (T. A. Vakaliuk); 0000-0002-6715-4667 (V. Yanchuk); 0000-0003-2466-6299 (A. Tkachuk); 0000-0002-5744-4599 (A. Humeniuk)



© 2024 Copyright for this paper is held by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Theoretical background

Recent research proves that information and computer technologies (ICT) in healthcare may greatly improve the level of service, support, and maintenance [3]. Some evaluation techniques for effectiveness are measurable, some are subjective, and some can be identified via supplementary parameters, models, and techniques, which contemporary businesses must invent to keep the usability and parameter collection as simple as possible.

The effectiveness of the healthcare system ensures human rights to healthcare and satisfies such a need. It is essential to investigate how we can develop a subsystem that incorporates contemporary technological advantages to optimise the process of evaluation, approximation, and identification of supplied material volumes for upcoming periods.

Among the advantages is substantial research on applying cloud computing to process calculations and evaluations at specific nodes, which has been effectively applied in different industries[4]. E-commerce reacted immediately, transitioning the application's most significant data and process-loaded segments to the cloud computing nodes or edge computing by transferring the solution to cloud services and implementing additional services.

Data aggregation, with the application of cloud computing possibilities [5], allows the process of distributing data and visualising them with contemporary reports and enables all that to the medical networks to perform ordering, logistics, inventory update, waste registration, and forecasting, a real benefit of an automated system for international companies, mitigating the risks of insufficient tests, liquids, and supplies. That is why different medical institutions have switched to automated devices, which can lower the expenditure of different liquids and avoid overdosing. At the same time, such devices can collect statistics on used liquids or supplies that can be exported to external networks, which may be helpful for automated aggregated systems.

The timely availability of products guarantees uninterrupted work of blood withdrawal centres, blood test laboratories, and many other institutions taking samples, including reproduction centres and sometimes even home medical visits, which are necessary if the patient has limited movement activities, etc.

It is a hot topic if research is accelerated with the constantly increasing number of tests the person needs to perform to start the diagnostics. Just six years ago, the number of tests was dramatically low, and COVID and blood tests exploded the industry of medical supplies for various tests and applications. Even though the number of tests increased, the supplier's physical aspects did not change seriously; however, the medical devices for processing increased significantly.

The vast majority of medical laboratories are creating supply chains based on pharmaceutical companies and agencies that correspond to medical and local standards and can satisfy diligent logistics and sales agreements. E-commerce solutions can do many activities as they may have a network or distributed architecture and involve Cloud Technologies internally. Some of the elements of cloud solutions are logistic elements, which are part of almost every E-commerce solution. One of the arguments to apply and support the logistics idea of using alternative logistics routes may still support medical laboratories even with higher delivery requirements, including satisfaction of cooling and heating delivery requirements [6]. Regardless of payment and delivery constraints, the prominent role of the evaluation is to predict how many supplies should be purchased shortly and in the context of an upcoming year.

To analyse and build the contemporary subsystem aiming to support medical blood tests and international organisations with contemporary devices, it is suggested that the international companies' current operation activities and infrastructure ordering from the integrated e-commerce solutions be analysed. A complete set of business analysis tasks will be formed as separate work; however, the fundamental solutions suggested will be based on network analysis [7] applied to the solution domain. It will enable the analysis of two scenarios when a higher or lower level of automation is applied. Indeed, the data gathered from the equipment reduces manual entries and creates data slices for analytic reports. Some laboratories still prefer manual updates since only some of the complete set of equipment can be installed to automate the work entirely. Thus, manual solutions should still be supported. However, both approaches are aimed at validating the possibility of collecting a minimal set of parameters included in an automated framework assisting medical organisations with a closed cycle of ordering, analysis of remaining

items available, analysis of wastes, and informing the provider of possible waste percentages, which forms the set of tasks for the current paper.

Analysis of the current lifecycle with elements of cloud computing and data pre-processing with possible aggregation can be supported with edge computing as a model that optimises the cloud computing system by processing data closer to the region where it is collected and used, a kind of cloud instance [8].

The cloud computing primary edge computing definitions certainly include the type of IoT data retrieval from devices and sensors that provide aggregation and processing. However, computing and calculations can be enriching factors that may go hand-in-hand with the decentralisation of functions and may seriously decrease data latency [9]. Geographically distributed applications, like in the case of international companies [11], have many similarities with the edge network of source data and may easily cover the essential capabilities such as application, computing, networking, storage, and performing processing services satisfying the agility requirements. In our case, real-time business connection and data optimisation are the prominent factors for minimising possible development costs.

Cloud computing has a decentralised architecture; it empowers several features that can be quite effective, such as processing data closer to the user's recipient region or location [11] and interacting with allocated data, which is primarily gathered if IoT devices generate it. In combination with cloud services, this data can be aggregated and transferred to generalisation units, satisfying lower latency and acceptable computation power. These features are mission-critical in healthcare [12] and organisation and institution management [13].

Thus, in our case, we may consider an international company with branches in different countries in Northern and Latin America, Europe, Asia, and the Pacific region that validates how to satisfy the needs of specific branches or the needs of the whole company in a specific region with aggregation to the corporation in general.

To make this happen, the existing e-commerce system infrastructure should be extended with a subsystem that can regionally collect, process, aggregate, compute, and transfer data to the upper level.

Thus, the data collection problem, addressed by the decentralised database and aggregated per layer and its usage via service, delivering data to the users' computers to solve obscure or localised issues, is already solved with distributed systems. Therefore, the data for particular medical meta-data will be stored in the respective subsystem. The system layers should consider local and global levels to guarantee granularity.

Hence, the research aims to illuminate the suggested approach and highlight the main aspects and results of medical subsystem development and usage.

3. Methods

The comparative analysis of e-commerce systems is skipped here since we focus on e-commerce, which is based on ERP systems that simplify the logic and data processing. The module of industry-specific adaptation for data collection, processing, and additional functionalities added to the E-Commerce solution as an extension is provided and considered. As the current trend of many e-commerce systems is SaaS solutions, that part is also considered. It increases the coverage of use cases and the breadth of applications.

To focus mainly on additional components and go in-depth about applying the extended functionality, doing calculations, and aggregations, we consider that the e-commerce solution already uses cloud solutions, integrates cloud ERPs, and works with cloud web shops.

The solution should consider presenting reports that reflect the data and results of the model application (forecasting), which is crucial for decision-making for all four key process players. They need it mainly to keep the process transparent and effective. The solution should suggest both the automated one, where IoT elements will provide the data collected, and the manual one, which allows the end-user to enter the data for further model adjustment.

Use-case diagrams and application diagrams, along with screenshots of proposed solutions, will allow us to dive into the heart of the development process and the main concepts and solutions proposed. Models applied for forecasting are still part of the commercial NDA-protected signature, so they are not highlighted, as they are not the core part of the provided results. For

the reader's audience, the current prediction model takes the annual amount of purchased items in the year's projection, considering the already purchased items and the average monthly volumes purchased in the recent three years. This means that purchase volume should be at least the average of a specific month for the last three years. For each upcoming month, the trajectory is built based on the recently purchased amount in the upcoming period so as not to be lower than the average of an annual total per specific item. The math is relatively simple for implementation and will not slow down the Power BI report.

4. Results

The medical subsystem is designed to support users' roles with actual real-time report information and updated visualisation of data on supplies needed for the upcoming period. As part of an integrated e-commerce solution, a subsystem enriches additional elements, resulting in data queries fed to reports assembled based on collected data (Figure 1).

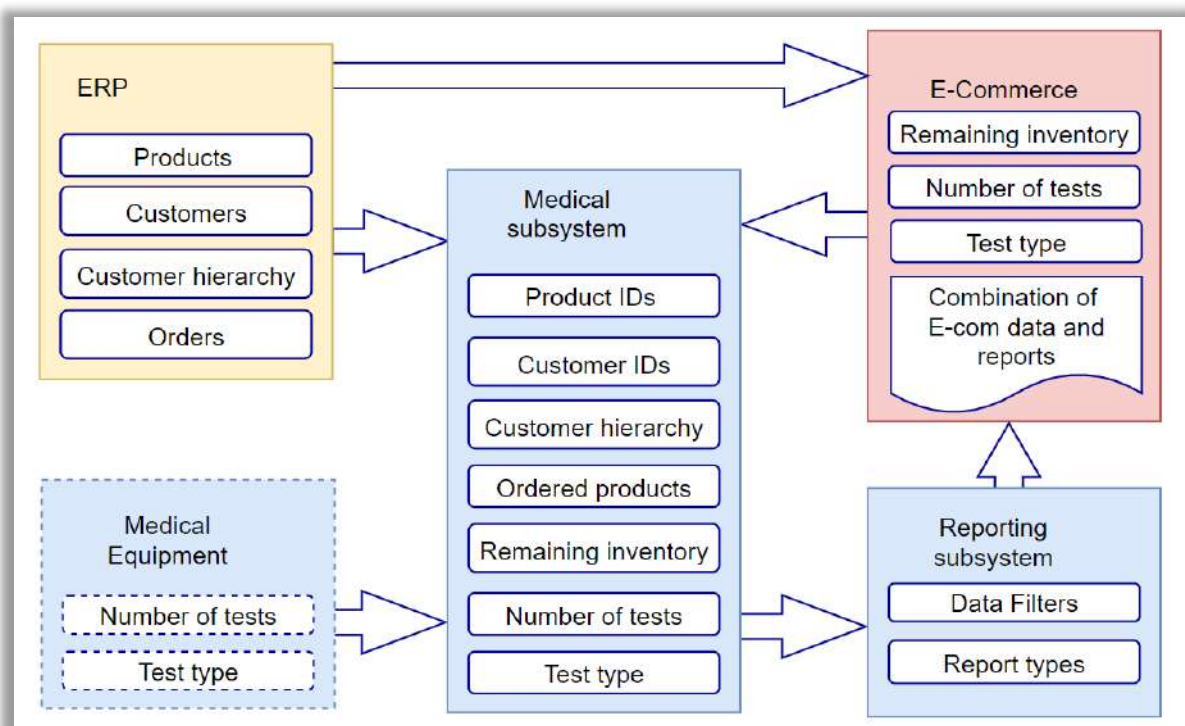


Figure 1: Data exchange model

Let us review the data schema for the proposed system and analyse the data flow in the context of the applicability of the cloud services and data layers for different roles.

The medical subsystem is sufficient to support medical institutions with up to 4 levels of hierarchy to support reporting. The number of levels may vary and impact the number of aggregation layers, which the supplier may outline if they can support such a hierarchy level. Most ERP-based e-commerce solutions support four levels. However, such systems may still install extensions, enabling more options. Reflection at the hierarchy level may impact the number of layers for data aggregation and the number of potential customers to be added to the system.

4.1. Architectural solution for the system

The general structure of the system proposed will be based on an ERP (for example, the SAP Hana Cloud) integrated with an e-commerce solution, as presented in Figure 2. Cloud ERP will ensure that every region accesses the data with minimal latency and consider data delivery to another subsystem.

The **proposed e-commerce** services will be based on cloud services that may guarantee presence in specific regions and data layers on the local level. Two servicing APIs, marked in blue,

will be provided for the front-end and back-end to control the usual e-commerce solution functions and the servicing functions of exporting necessary anonymised data to the medical subsystem.

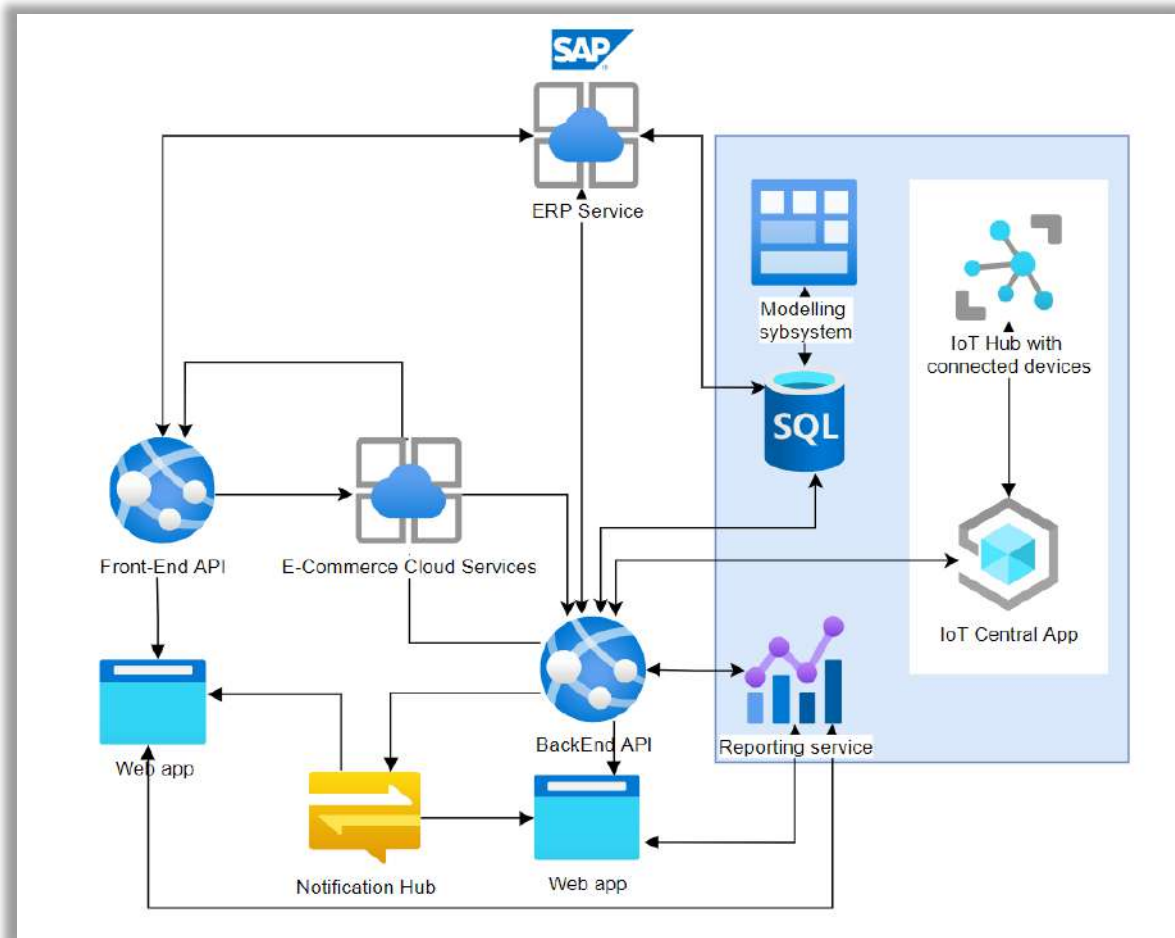


Figure 2: Architecture of the integration of the API with applications and services

Both APIs should have a respective control by Web application that brings ease in access allocation and control. Both applications will serve as entry points for groups of users, where the front-end and back-end users will have their levels of access and customer-type recognition. All the above units support node allocation and transfer of permissions to specific nodes of the allocated service the user works with.

Data collected from the nodes will be processed within the service call and stored in the medical subsystem node, which has storage and access to other aggregation layers. It will ensure that users will operate within their data layers and use the calculation power of the specific node.

The medical subsystem will have servicing functions to interact with reporting services and IoT devices and collect data from respective devices.

The reporting service will have a monitoring subunit that interacts with the system's back-end API to signal which report still needs to be updated. Such triggered notification requests will be made at the customer level. Aggregated data will trigger signals to the API to inform the managers, who should act and enforce their subordinate employees to provide reports regularly for correct data aggregation.

It is worth noting that the IoT may not be added to every organisation, which is why the e-commerce solution will still send the data via the e-commerce cloud service to the medical subsystem to store the data and feed it to the modelling subsystem, which is responsible for making analysis and predictive reports for the upcoming period.

4.2. Data transfer between the parts of the system

As long as the customer and product data will be stored in the ERP and the main focus of the customer, manager, administrator, and supplier will be interacting with the e-commerce solution, it is suggested to develop and implement the solution architecture and describe the use-cases for the medical subsystem that will be serving as the data storage, aggregation unit, and model application unit that will work for the local nodes as well as for the upper aggregated level for global statistics.

Review the data structure needed to develop the overall system data flow.

The **ERP** will provide a list of "product IDs" and "orders" that were explicitly purchased, along with the customer ID, "customer hierarchy position," and "customer type."

The above attributes are needed for proper reporting filtering and layer distribution for proper aggregation happening in the medical subsystem and have a projection to the reporting system. The customer data will be anonymised to satisfy the requirements of PCI compliance, which is a vital requirement for the General Data Protection Regulation according to Regulation (EU) 2016/679. Thus, the medical subsystem will work with the data only around specific customer IDs, which are fully anonymised and will solely use customer-respective attributes for reporting.

There is a specific medical equipment subunit of the system that has IoT sensors that may collect data and avoid manual entry of the number of tests performed and gathered: "Test registration number" reflects the total quantity of tests in a specific time range; "Test type" characterises the substances and supplies needed for this type of test.

The **Medical Equipment** sub-unit is a variable part that will support the data aggregation from the automated services. This is quite a novel part, reflecting this part of the solution. However, the medical laboratories still do not guarantee a similar unified type of equipment.

The **reporting subsystem** will differ slightly from the general medical system, as it may operate across the database and reflect pretty much any report supporting technological aspects.

The **e-commerce** part will enable the respective role to send the remaining inventory to the medical subsystem, ensuring the model is supported with proper data. If the Medical Equipment sub-unit is unavailable, the number of tests with respective test types will be manually entered from the e-commerce system.

Rendered reports will reflect the combination of medical subsystem data via the reporting subsystem unit, applying the report type and data filters to the data and accessibility layer and reading either the specific node data or the aggregated territory layer.

Indeed, the current version is an initial implementation that can be extended and updated; however, for the moment of research and implementation, it has already provided quite good results with the current setup.

The essential operations of the platform are based on the primary existing nodes, which will be integrated with new subunits following their requirements and data integrity to help the layers function, respectively.

4.3. Usability case

The current solution's foreseen use cases satisfy all 4 groups of roles using the system indicated in the use case diagram with the main actors of the process, Figure 3.

The leading role enables customers to maximise profit while maintaining reasonable supply satisfaction and uninterrupted service. The system provides notification services regularly, enforcing the data entries in due time. An automated service for collecting the test results avoids requiring the customer to enter the number of tests in automated or semi-automated mode.

The primary users of the subsystem are medical analysts (customers) and their senior managers (managers) who are observing state-of-the-art (receiving notifications about reports not submitted or updated, entry/checking of the inventory values), organisation administrators (review dashboards, adjustment of the reporting filters, validation of the data integrity), and industry representatives (review dashboards, purchased volumes, and projected orders in the upcoming future).

Indeed, the current diagram only focuses on the medical subsystem and excludes all e-commerce functionalities and supportive use cases that are covered by default ERP or cloud services. It must fully disclose the main point of embedding the multilevel possibilities for users to access the reports from the respective data layer.

Nonetheless, the diagram already indicates that two levels of users, customer and manager, may see different reports and get access to several layers of data produced for reports by different layers of aggregation. Customer operates with reporting only within their account, which is local, and using the locally aggregated data fed to the reporting system, where the manager may act as a customer and be able to switch to the level of customers observing, which will switch to a different aggregation level and feed the report with respective data. As the implementation is done within the real-time system, accessing the real-time e-commerce and ERP data will refresh the enrichment of the report.

A similar situation happens to the administrator, who may switch between different levels of aggregated data since the data validation and integrity check, customer hierarchy validation, and transformation still access different data sets and involve different system nodes.

From the supplier's perspective, no granularity of different layers of users is foreseen, only the corporate level with aggregated data, since the supplier is more interested in branch, geographical, and corporate levels rather than customer-specific.

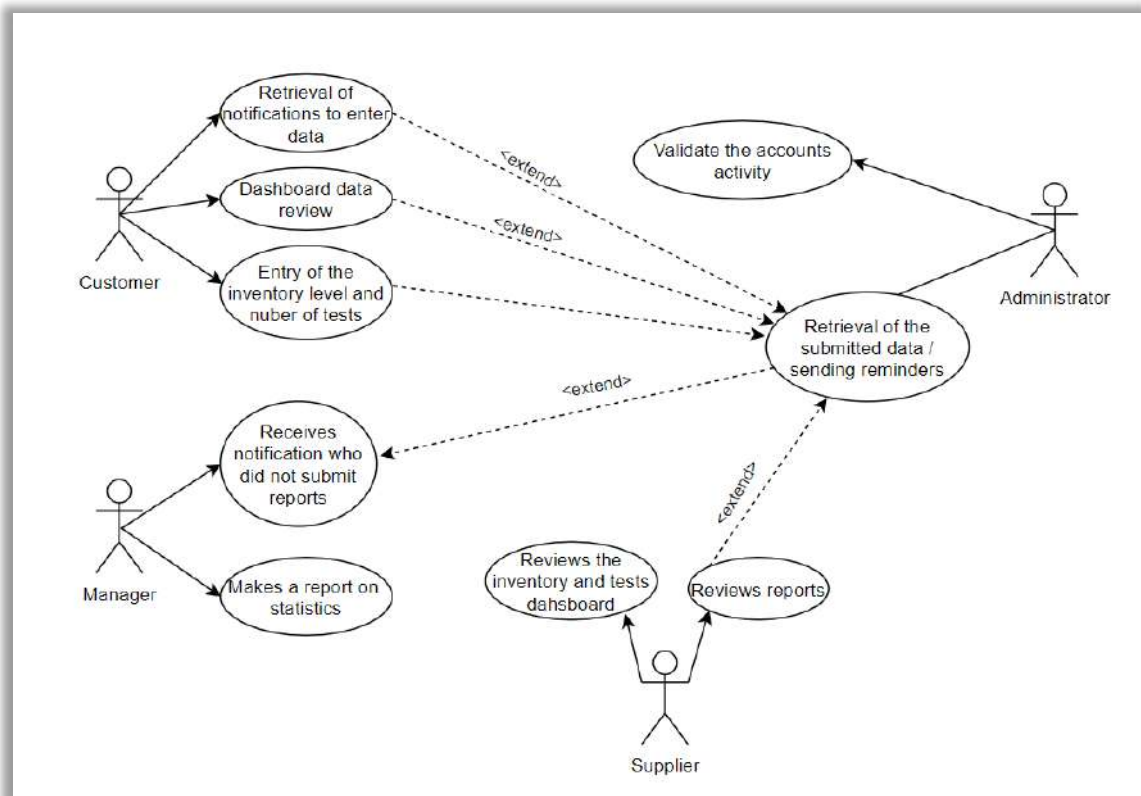


Figure 3: Use-case diagram with the main actors of the process

Therefore, the process goes further: providers submit the data, which is then validated by the system and other customer hierarchy levels involved. To keep the data up-to-date, the key players receive notifications on missing reports or overdue deadlines for report submission. Dashboards are accessible for crucial process players, supporting the overview process daily.

Every e-commerce solution already contains the full features and activities that satisfy the customers' need to find the product. However, the place where reporting is to be embedded should correspond to the requirements of minimal latency and keeping the requests to the real-time processing units at a minimum. Thus, the long lists and search results should be free from unnecessary calls.

Thus, the central reporting places of the e-commerce landscape should be the product detail pages, as they are not using batch processing models, where lists and comparison pages will have

a more significant number of requests to the medical subsystem. Embedded reports will be shown via a direct API call that renders the report on the page. This can be achieved in various ways; however, the report must read the aggregated data from the medical subsystem so that the data is reflected.

The usual start of an analytic report will be triggered by the button Analytic Report, which will start a request to report and send requested parameters, including the report time. So, the page is updated with specific segments reflecting data; see Figure 4. The initial load should not trigger a report since there is no need to bother the additional subsystem if the customer is just willing to read the product details.



Figure 4: Example of report rendered on the E-Commerce page with the price diagram

The customer should be able to choose between report types directly on the same page without reloading the page and minimising the impact on the current page load time, which is crucial for e-commerce users.



Figure 5: Example of report rendered on the webshop page with the prediction model results

Similar solutions for multiple ordered products can be embedded in the customer's "My Account" section, where the aggregated data for the most purchased products may be displayed to the customer and the manager.

Similar reports may be accessible to the supplier in a product or customer context.

5. Discussion

The solution architecture is structured to support distributed nodes and their coverage with network and cloud solutions. Data structures and solutions are relevant to representing taxonomies in the CIT polyhedron system [15].

Implementing a medical subsystem seriously extends the solution's applicability in general and extends the current integrated solution with subject-domain data compared to the existing standard e-commerce system. Collected data processed in the cloud services demonstrated good results from a latency perspective; however, it may need more extensive validation in other cases.

The infrastructure and data sets are arranged according to European integration and the main principles of findability, accessibility, interoperability, and reusability (FAIR). International standards require software developers to respect the European Union declaration of open data and usage in the medical and governmental sector regulated by Directive (EU) 2019/1024, which is crucial to obey the law and ensure data protection according to GDPR (General Data Protection Regulation according to Regulation (EU)) 2016/679 and Guidelines (EU) 2020/C 122 I/01.

The system functions based on different add-ons implemented as extensions that connect via APIs and services, supporting cloud technologies that enable horizontal scaling.

From the usability and applicability standpoint, the coverage of 4 roles of users, including two types of customers, corporate analysts and corporate managers, opens the broader analytical possibilities for the plans of ordering versus reporting of the tests performed. Administrative personnel support ensures the integrity and validation of the aggregated values on the level of operation within the e-commerce platform—the potential extension of the API calls for exporting more raw data for further adjustments. The accessibility of the supplier and the extension of their visibility of purchased volumes and data modelling for future periods makes it possible to project production volumes. As described in [14], the stakeholder's data should respect sufficient precision and have a layered structure.

Transparency plays a crucial role in e-commerce since the critical indicator of engagement of all stakeholders in the processes may be guaranteed with only available data and thoroughly validated reports. Current publication activity demonstrates the interest and importance of the subject [16], [17], and the whole importance of medical reporting [18], [19] with the application of contemporary technologies [20] in the post-COVID-19 period.

The biggest advantage of the proposed solution model is its scalability for different regions. It can be applied to various types of businesses where generalised operational data is most beneficial. There are many famous solutions for Admin Panel reports related to revenue, operational evaluations, etc. Front-end reports are beneficial for ongoing operation functionality.

Among the application's limitations is that the business-to-consumer (B2C) scenario should be excluded, leaving the power for business-to-business (B2B) scenarios. Power BI reports at the highest license level allow around 480 live connections per second, which may easily be abused by random users from the consumer segment.

From a calculation perspective, the ERP system's calculation power should not be utilised. Several calculations impact the routine operations that are called to serve the daily operations.

Figure 6 below represents the comparative analysis of the period recorded in analytics. Minding that the project went live in August 2023, the number of shop accounts created in the webshop seriously increased, proving usability. In contrast, the new shop account creations indicated spikes around the seasonal exhibitions and fares. The number of page views is stable; however, the number of orders created offline seriously dropped after migration to the newer architecture.

A stable trend of transactions increases with the usability and scalability of specific regions that have a stable reflection of the transaction amount and value of online transactions.

6. Conclusions

The paper proved that the extensibility of e-commerce solutions enables subject-oriented add-ons and components that may play a crucial role in the work processes of medical laboratories and supply organisations. The suggested approach allows for consolidating data and aggregating values to help laboratories make better purchase plans and evaluate supply chains more effectively.

The proposed model enabled many solutions to apply the technologies in combination with the software available to support a broader range of applications. Examples of medical devices indicated how visual and effective a combination of a reporting system with a data processing

subsystem is and how the approach of the fully automated solution may be extended with manual entries and still follow all processes of data integration and integrated system functioning daily.

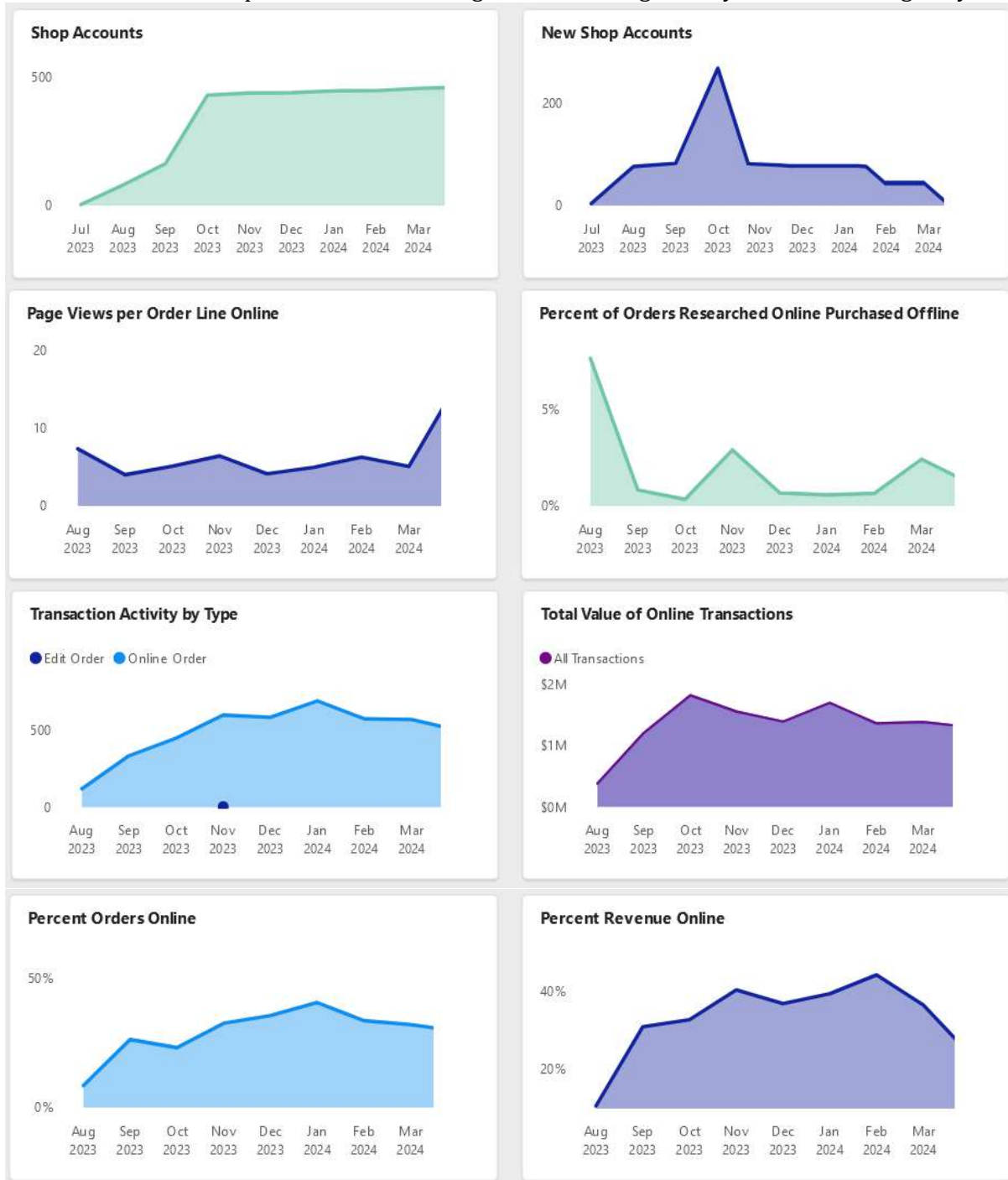


Figure 6: Comparative analysis of the leading critical indicators for the new platform implementation in August 2023 to date

Particular efficiency is found in the local validation segment for organisations and organisation branches since such organisations can represent each department and evaluate their particular level of purchases and potential purchase plan. Aggregating the data at the company level supports faster evaluation and reporting.

A wasted algorithm characterised by the number of tests is an essential part of the validation model and appears as a corrective factor for the solution from the customer angle as well as from the product angle.

Aggregated data at the company level allows the centralised allocation of the data for managers and the reports at different levels of granularity. Cloud computing ensures that the medical subsystem satisfies an acceptable level of productivity and calculation possibilities with

minimal latency. Even though the data on the local nodes is calculated and stored, the aggregated data across the regions will have two modes: server-side aggregation, which involves extracting data and building reports on the fly or creating pre-rendered reports for delivery to the end-user.

Adhering to the FAIR principles and the stipulations outlined in Directive (EU) 2019/1024 and Regulation (EU) 2016/679 is crucial in the context of medical information. Taxonomies can serve as a potent method for representing conference data. The digital transformation of medical information can contribute to establishing equitable and transparent conditions. These guidelines follow the Guidelines (EU) 2020/C 122 I/01.

The proposed approach can be applied in different application areas, propagating solutions combining hybrid systems with external services for aggregation visualisation and API integration.

References

- [1] Semerikov, S. Chukharev, S. Sakhno, A. Striuk, V. Osadchyi, V. Solovieva, T. Vakaliuk, P. Nechypurenko, O. Bondarenko, and H. Danylchuk, Our Sustainable Coronavirus Future, volume 166, EDP Sciences, 2020. doi:10.1051/e3sconf/202016600001.
- [2] Pessoa-Gonçalves, Yago & Jesus, Ana & Desiderio, Chamberttan & Minchio, Gabrielly & Louzada, Arthur & Shimano, Marcos & Oliveira, Carlo. (2023). Understanding the Relationship between Vaccine Supply Dead Space and Wasted COVID-19 Vaccine Doses. *Revista da Sociedade Brasileira de Medicina Tropical*, 56, 10.1590/0037-8682-0353-2023.
- [3] Klochko, V. Fedorets, O. Maliar, and V. Hnatuyk, The use of digital models of hemodynamics for the development of 21st-century skills as a component of healthcare competence of the physical education teacher, *E3S Web of Conferences* 166 (2020). doi:10.1051/e3sconf/202016610033.
- [4] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, The role of edge computing in the Internet of things, *IEEE Communications Magazine* 56 (2018) 110–115. doi:10.1109/MCOM.2018.1700906.
- [5] Huh, Y. Seo, Understanding edge computing: Engineering evolution with artificial intelligence, *IEEE Access* 7 (2019), 164229–164245. doi:10.1109/ACCESS.2019.2945338.
- [6] Yanchuk V M, Antoniuk D S, Tkachuk A G, Maestri E., and Vizghalov O 2020 Integration of delivery services business cases for improvement of business and environmental sustainability of e-commerce solutions Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization, and Knowledge Transfer. Volume II: Workshops, Kharkiv, Ukraine, October 6–10, 2020 (CEUR Workshop Proceedings, vol. 2732) Sokolov O, Zholtkevych G, Yakovyna V, Tarasich Y, Kharchenko V, Kobets V, Burov O, Semerikov S and Kravtsov H (CEUR-WS.org), pp. 363–376 URL: <http://ceur-ws.org/Vol-2732/20200363.pdf>
- [7] Gobov, D., Yanchuk, V.: Network Analysis Application to Analyse the Activities and Artifacts in the Core Business Analysis Cycle. In: 2021 2nd International Informatics and Software Engineering Conference (IISEC), pp. 1-6. IEEE, Ankara, Turkey (2021) <https://doi:10.1109/IISEC54230.2021.9672373>
- [8] J. Taheri, S. Deng (Eds.), Edge Computing: Models, technologies and applications, volume 33 of IET professional application of computing, Institution of Engineering and Technology, 2020. doi:10.1049/PBPC033E.
- [9] K. Cao, Y. Liu, G. Meng, Q. Sun, An Overview on Edge Computing Research, *IEEE Access* 8 (2020), 85714–85728. doi:10.1109/ACCESS.2020.2991734.
- [10] Z. Zhao, F. Liu, Z. Cai, and N. Xiao, Edge Computing: Platforms, Applications and Challenges, *Jisuanji Yanjiu yu Fazhan/Computer Research and Development* 55 (2018) 327–337. doi:10.7544/issn1000-1239.2018.20170228.
- [11] S. Munirathinam, Chapter Six: Industry 4.0: Industrial Internet of Things (IIOT), in P. Raj and P. Evangeline (Eds.), *The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases*, volume 117 of *Advances in Computers*, 2020. doi:10.1016/bs.adcom.2019.10.010.
- [12] R. Dave, N. Seliya, and N. Siddiqui, The Benefits of Edge Computing in Healthcare, Smart Cities, and IoT, *Journal of Computer Sciences and Applications* 9 (2021) 23–34. doi:10.

- [13] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, Edge-Computing-Enabled Smart Cities: A Comprehensive Survey, *IEEE Internet of Things Journal* 7 (2020), 10200–10232. doi:10.1109/JIOT.2020.2987070.
- [14] Nick Rozanski, Eóin Woods. *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Addison-Wesley, 2011, 704 p.
- [15] O. Stryzhak, V. Prykhodniuk, M. Popova, M. Nadutenko, S. Haiko, and R. Chepkov, Development of an Oceanographic Databank Based on Ontological Interactive Documents, in K. Arai (Ed.), *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 2, SAI 2021, Virtual Event, 15-16 July 2021, volume 284 of Lecture Notes in Networks and Systems*, Springer, 2021, pp. 97–114. doi:10.1007/978-3-030-80126-7_8.
- [16] S. Semerikov, S. Chukharev, S. Sakhno, A. Striuk, V. Osadchy, V. Solovieva, T. Vakaliuk, P. Nechypurenko, O. Bondarenko, and H. Danylchuk, Our sustainable coronavirus future, volume 166, *EDP Sciences*, 2020. doi:10.1051/e3sconf/202016600001.
- [17] O. Pursky, A. Selivanova, I. Buchatska, T. Dubovyk, T. Tomashevskaya, and H. Danylchuk, Features of learning motivation in the conditions of coronavirus pandemic (COVID-19), *Educational Technology Quarterly* 2021 (2021), 375–387. doi:10.55056/etq.31
- [18] Nikitchuk, T.M., Andreiev, O.V., Korenivska, O.L. and Medvediev, M.G., 2023. A model of an automated biotechnical system for analysing pulseograms as a kind of edge device. *Journal of Edge Computing* [Online], 2(1), pp. 64–83. Available from: <https://doi.org/10.55056/jec.627>
- [19] Nikitchuk, T.M., Vakaliuk, T.A., Chernysh, O.A., Korenivska, O.L., Martseva, L.A. and Osadchy, V.V., 2022. Non-contact photoplethysmographic sensors for monitoring students' cardiovascular system functional state in an IoT system. *Journal of Edge Computing* [Online], 1(1), pp. 17–28. Available from: <https://doi.org/10.55056/jec.570>.
- [20] Korenivska, O.L., Bedytskyi, V.B., Andreiev, O.V. and Medvediev, M.G., 2023. A system for monitoring the microclimate parameters of premises based on the Internet of Things and edge devices. *Journal of Edge Computing* [Online], 2(2), pp. 125–147. Available from: <https://doi.org/10.55056/jec.614>

Simulation of Intelligent Air Transportation Management System Based upon Entropy Approach

Andriy V. Goncharenko

National Aviation University, 1, Liubomyra Huzara Avenue, Kyiv, 03058, Ukraine

Abstract

The paper is devoted to the computer modeling for the optimal governing of the intelligent air transportation management system. The performed computer simulation is based upon the initial postulate of the subjective entropy maximum principle developed for the active systems control. The research and calculation experimentations are conducted for both losses (“harmfulness”) and utilities (“usefulness”) functions. It is discovered the important phenomenon unknown before, which is the shape of the phase diagrams for the preferences over the losses functions. Another novelty and significance of the findings related to the preference functions and entropy are that the conditional optimization of the subjective individuals’ preferences functions entropy in conjunction with the proposed hybrid combined relative pseudo-entropy function helps determine the relative certainty/uncertainty degree concerning prevailing/dominating subjective preferences functions.

Keywords

Entropy, preferences, uncertainty, air transport, optimization, intelligence, management, simulation, objective functional.

1. Introduction

Computer modeling of intelligent systems is an actual and important task for scientific research and investigations. The urgency for such kinds of study is dictated by the contemporary informative world and intensifying development of intelligent technologies. Since air transportation management systems are functioning in some complex operational situations, there must be some corresponding scientific approaches allowing assessing the circumstances of the occurred multi-alternativeness.

One part of the influential factors, requiring computer modeling, is the issues of the aircraft as whole (including its powerplants) proper maintenance and repair [1, 2] in order to keep the reliability of the aeronautical engineering and the flight safety of the entire aircraft itself [3, 4].

Elements of the intelligence, either natural or artificial, are present at the air transportation management system’s making governing decisions and operational alternatives choice. Such processes might be considered from the point of view of the utility theory [5, 6].

Anyway, the entropy paradigm formulated in [7 – 9], and used widely in science nowadays [10], with taking into account economic aspects [11], realized in the theory of active systems and subjective preferences [12], should be rather effectively implemented to the problems of the intelligent aviation radio equipment reliability parameters monitoring [13], revealing needed properties of new materials [14], neural networks of different kinds modeling [15, 16].

The essential feature is the application of such type of the entropy paradigm as used in [7 – 10, 12, 17 – 20] in order to model the properties of the intelligent air transportation management system’s optimal behavior in conditions of the available operational multi-alternativeness causing the uncertainty of the individuals’ subjective functions of their preferences.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

 andygoncharenko@yahoo.com (A. V. Goncharenko)



0000-0002-6846-9660 (A. V. Goncharenko)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Development of theoretical models with the elements of simulation

The modeling of the intelligent air transportation management system's optimal behavior requires a closer look into theoretical provisions. The uncertainty of the individuals' subjective preferences functions should be assessed with the subjective entropy [12]. The well-known Jaynes' principle of the entropy maximum [7 – 9] implemented into the field of aviation will help cope with the problems connected with the notorious human factor. First, let us consider a portion of the developments dedicated to the subjective preferences theory provisions.

2.1. Basic concept

It is proposed to apply the generalized model, taking into account the operational uncertainty with the help of the objective functional of the intelligent air transportation management system, in the view of [12]:

$$\Phi_{\pi}^{-} = - \sum_{i=1}^N \pi^{-}(\sigma_i) \ln \pi^{-}(\sigma_i) - \beta \sum_{i=1}^N \pi^{-}(\sigma_i) L(\sigma_i) + \gamma \left[\sum_{i=1}^N \pi^{-}(\sigma_i) - 1 \right], \quad (1)$$

where $\pi^{-}(\sigma_i)$ – corresponding subjective individual's preference function of the responsible decision making person distributed with respect to the negative qualities of the achievable for the person's goals alternatives σ_i ; β and γ – the values, introduced in the objective intelligent air transportation management system functional (1), in some respect, could be defined as endogenous parameters reflecting certain features and properties of psych, or internal parameters of the intelligent air transportation management system, the uncertain Lagrange multipliers, some coefficients or weight coefficients at the specific problem settings [12]; $L(\sigma_i)$ – the corresponding function of the personally estimated losses (“harmfulness”) related with the available alternatives; the individual distinguishes a certain one-sided attitude to the managerial process.

The first member in the intelligent air transportation management system objective functional (1) having the expression of

$$- \sum_{i=1}^N \pi^{-}(\sigma_i) \ln \pi^{-}(\sigma_i), \quad (2)$$

is the subjective entropy H_{π} , that is the measure of the operational multi-alternativeness uncertainty of the individuals' preferences functions of the effectiveness (in the given case study: losses $L(\sigma_i)$ related to the alternatives).

The last member in the intelligent air transportation management system objective functional (1) having the expression of

$$\sum_{i=1}^N \pi^{-}(\sigma_i) - 1, \quad (3)$$

is the normalizing condition.

On the contrary, formulating the objective functional as, [12]:

$$\Phi_{\pi}^{+} = - \sum_{i=1}^N \pi^{+}(\sigma_i) \ln \pi^{+}(\sigma_i) + \beta \sum_{i=1}^N \pi^{+}(\sigma_i) U(\sigma_i) + \gamma \left[\sum_{i=1}^N \pi^{+}(\sigma_i) - 1 \right], \quad (4)$$

where $\pi^{+}(\sigma_i)$ – corresponding positive subjective individual's preference function of the responsible decision making person; $U(\sigma_i)$ – the corresponding function of the personally estimated utility related with the available alternatives; the individual distinguishes a certain second-sided attitude to the managerial process.

Both formulation of (1) and (4) that is both the negative (losses, harmfulness) and positive (utility, usefulness) estimations are possible.

And the both-sided estimated managerial process of (1) and (4) is extremized with the use of the necessary conditions for the subjective preferences functions entropy conditional optimization in the view of

$$\frac{\partial \Phi_{\pi}^{-}}{\partial \pi^{-}(\sigma_i)} = 0, \quad \frac{\partial \Phi_{\pi}^{+}}{\partial \pi^{+}(\sigma_i)} = 0, \quad (\forall i \in \overline{1, N}). \quad (5)$$

For both senses of (1) and (4) the conditions of (5) yield the so-called canonical distributions of the subjective preferences functions [12].

In cases of the object (item, thing) subjective preferences these canonical distributions of the preferences functions are conventionally called: the subjective preferences functions distributions of the first kind [12].

The optimal distributions are as follows, [12]:

$$\pi^{-}(\sigma_i) = \frac{e^{-\beta_L L(\sigma_i)}}{\sum_{j=1}^N e^{-\beta_L L(\sigma_j)}}, \quad \pi^{+}(\sigma_i) = \frac{e^{\beta_U U(\sigma_i)}}{\sum_{j=1}^N e^{\beta_U U(\sigma_j)}}, \quad (6)$$

where β_L and γ_L – corresponding coefficients with the subscript for the objective intelligent air transportation management system functional of (1) having related with the negative sense alternatives σ_i effectiveness estimations; and β_U and γ_U – corresponding coefficients if the subjective attention is drawn to the positive features, in generally speaking terms to the same set of the alternatives.

Illustration of the theoretical speculations, described with the expressions of (1) – (6), is in [12] too; and it is shown in the Figure 1.

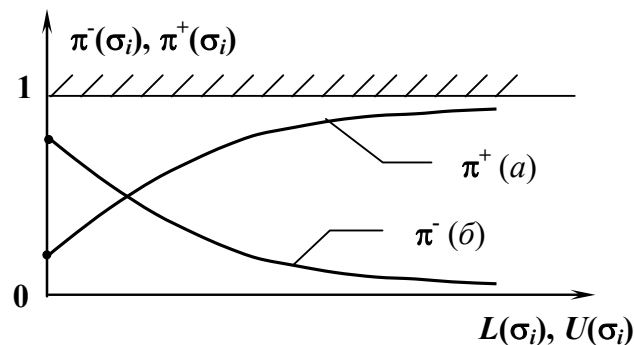


Figure 1: The shape of the preferences functions, [12]

Now, there arises a big conceptual question. The curve denoted as $\pi^{-}(\sigma_i)$ is monotonously decreasing as the corresponding harmfulness function increases, but the point is that, that this is not always so, because, if it, the preference function for some of the alternatives, decreases, it means that for the other alternatives it symmetrically increases (absolutely like $\pi^{+}(\sigma_i)$ preference shown in the Figure 1 as well).

Let us demonstrate this ideological collision with the simplest modeling.

2.2. Modeling

For the simplest modeling let us consider the linear increase of harmfulness (losses) functions with respect to a distinguishing parameter for a three-alternative case.

The computational data are as follow:

$$\sigma = -100 \dots 100, \quad \beta_L = 0.009, \quad L_1(\sigma) = 10\sigma, \quad L_2(\sigma) = 15\sigma, \quad L_3(\sigma) = 20\sigma. \quad (7)$$

The results of the simulation are shown in the Figures 2 – 5.

The diagrams plotted in the Figure 2 are for the losses (“harmfulness”) functions values computed by the last three equations of (7) correspondingly.

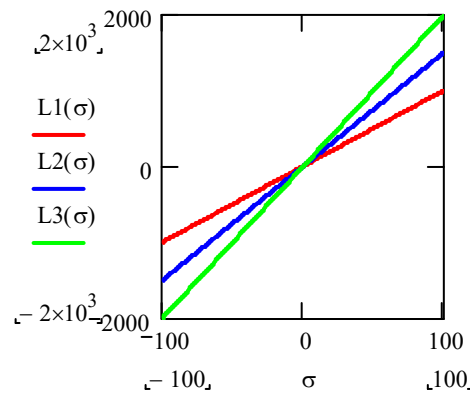


Figure 2: Harmfulness functions

The next up is the illustration for the subjective individuals' preferences functions represented in the Figure 3.

Also, the normalizing condition (3) check is realized with the “one” value made visible in the Figure 3.

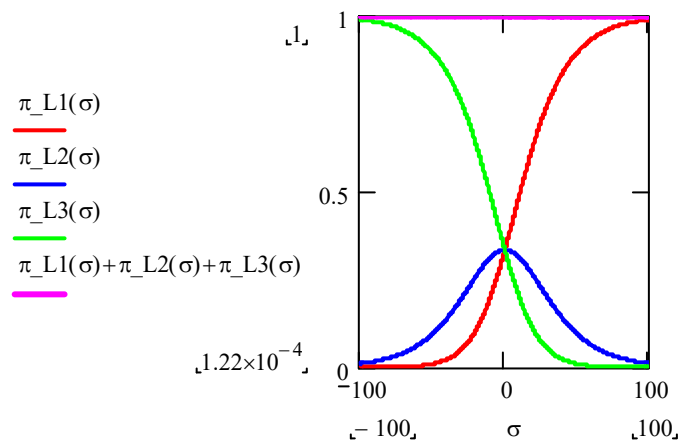


Figure 3: Subjective preferences functions

In order to get the phase portrait equivalent to the one shown in the Figure 1, the diagrams in the Figure 4 are plotted. The normalizing conditions (3) check is presented there as well.

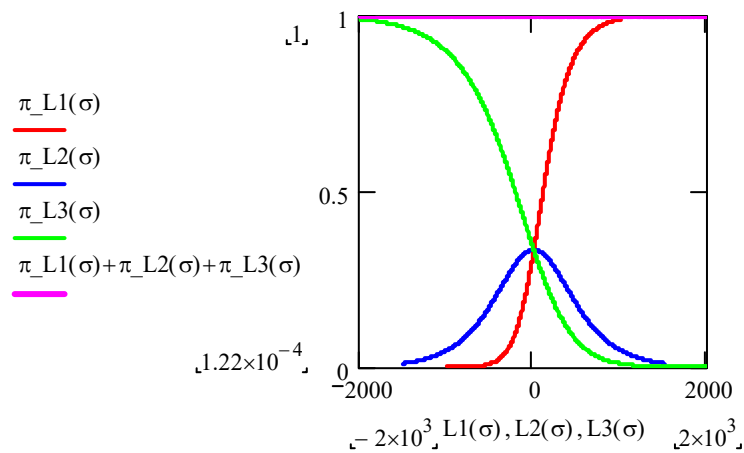


Figure 4: Subjective preferences functions with respect to losses functions

Thus, the diagram of $\pi^-(\sigma_i)$ shown in the Figure 1 deals with just a very particular case.

The subjective entropy of H_π , computed with the help of the expression (2), of the subjective individuals' preferences functions $\pi^-(\sigma_i)$, computed with the use of the first equation of the expressions of (6), is illustrated in the Figure 5.

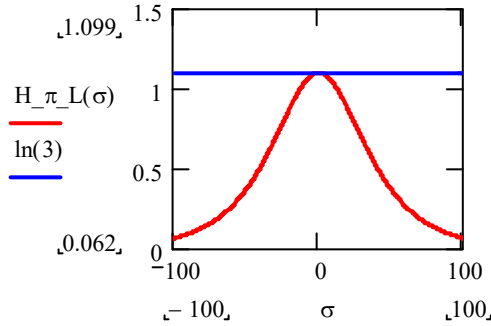


Figure 5: Subjective preferences functions with respect to losses functions

The maximal subjective entropy value, which is

$$\ln(3), \quad (8)$$

is also shown in the Figure 5.

Here, one more important theoretical question arises, whether the traditional view entropy H_π , having the expression of (2), is able to distinguish what the preferences distributional uncertainty the entropy H_π determines.

In order to clarify this issue, the relative combined pseudo-entropy function is proposed to be used in addition to the entropy H_π of (2).

2.3. Modeling based upon pseudo-entropy function

As it is seen from the Figure 5, the traditional view subjective entropy of H_π , computed with the help of the expression (2), is incapable to clarify the entropy value for the specific distribution of the subjective individuals' preferences functions $\pi^-(\sigma_i)$.

Therefore, it is proposed to make use of the measure of the certainty/uncertainty in the view of the hybrid model of the combined pseudo-entropy function of the subjective functions [17]:

$$\bar{H}_{\max - \frac{\Delta\pi}{|\Delta\pi|}} = \frac{H_{\max} - H_\pi}{H_{\max}} \frac{\Delta\pi}{|\Delta\pi|} = \frac{H_{\max} + \sum_{i=1}^N \pi(\sigma_i) \ln \pi(\sigma_i)}{H_{\max}} \frac{\left[\sum_{j=1}^M \pi(\sigma_j^+) - \sum_{k=1}^L \pi(\sigma_k^-) \right]}{\left| \sum_{j=1}^M \pi(\sigma_j^+) - \sum_{k=1}^L \pi(\sigma_k^-) \right|}, \quad (9)$$

where $H_{\max}$ – maximal subjective entropy value, in discrete alternatives problem settings this value constitutes:

$$H_{\max} = \ln N; \quad (10)$$

$\Delta\pi$ – factor/index of the preferences functions prevailing/dominance:

$$\Delta\pi = \sum_{j=1}^M \pi(\sigma_j^+) - \sum_{k=1}^L \pi(\sigma_k^-), \quad (11)$$

where σ_j^+ – positive and σ_k^- – negative alternatives correspondingly; M – the number of the positive alternatives; L – the number of the negative alternatives in respect:

$$M + L = N. \quad (12)$$

Thus, computer modeling allows, designating some subjective individuals' preferences functions as "positive" or "negative", making it visible which alternative preference dominates in the certainty/uncertainty degree. Moreover, it is important that the value of (9) is relative which is also more convenient.

2.4. Simulation

For the case considered above, it is possible to distinguish alternatives subjective preferences functions one by one. For the first alternative the preferences prevailing factor (11) gives:

$$\Delta\pi = \pi_1 - (\pi_2 + \pi_3). \quad (13)$$

Computation results for (6) – (12), with (13) for (11), are shown in the Figure 6.

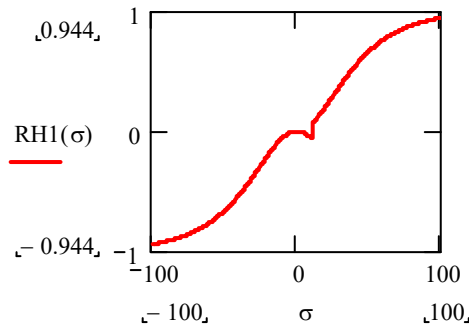


Figure 6: Relative pseudo-entropy function for the first alternative function dominance

The curve in the diagram in the Figure 6 is plotted by the calculations with formula (9) where (13) stands for (11). Similar procedures for the other two alternatives:

$$\Delta\pi = \pi_2 - (\pi_1 + \pi_3) \quad \text{and} \quad \Delta\pi = \pi_3 - (\pi_1 + \pi_2). \quad (14)$$

bring the results shown in the Figures 7 and 8.

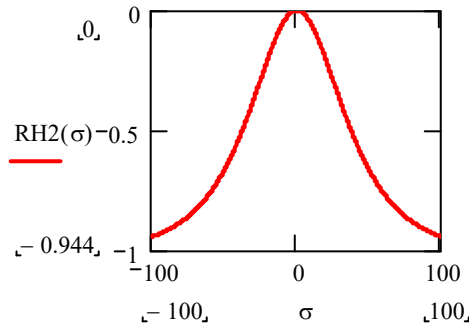


Figure 7: Relative pseudo-entropy function for the second alternative function dominance

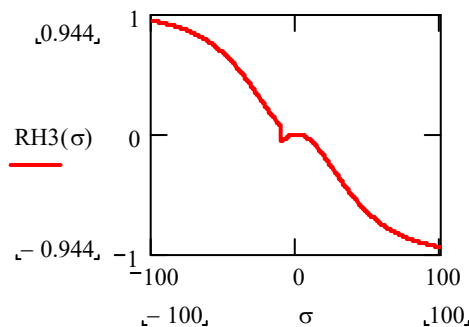


Figure 8: Relative pseudo-entropy function for the third alternative function dominance

All the three pseudo-entropy functions are shown in the Figure 9.

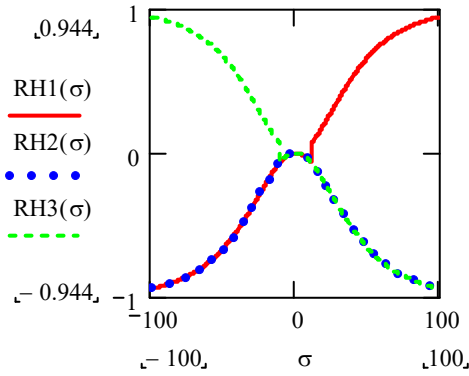


Figure 9: Relative pseudo-entropy functions for the three alternative functions dominance

For the visibility and comparison analysis the entropies and preferences functions are represented in the Figure 10.

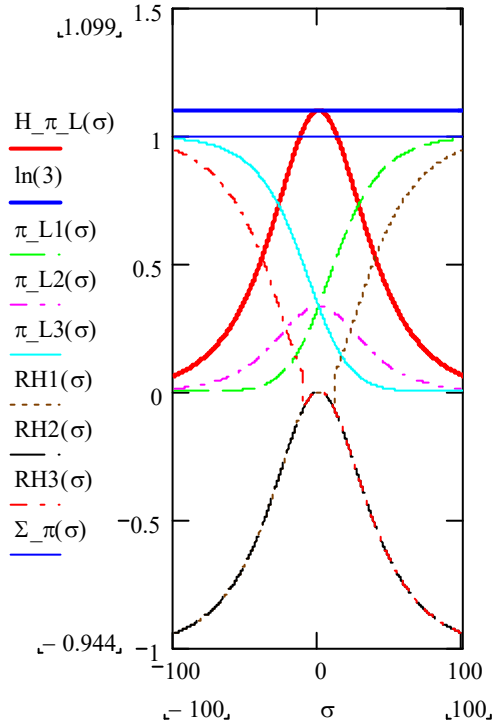


Figure 10: Relative pseudo-entropy functions for the three alternative functions dominance

2.5. Modeling a more general case

Now, it is possible to consider some more generalized models when some of the functions of losses are nonlinear.

The data that are different from the above case are as follows:

$$L_2(\sigma) = -0.00225\sigma^3, \quad L_3(\sigma) = -0.2\sigma^2. \quad (15)$$

The results of the computer modeling calculation simulations in the style of (1) – (14) with the corresponding functions of (15) are represented in the Figures 11 – 19.

The diagrams for functions of losses (harmfulness) are shown in the Figure 11.

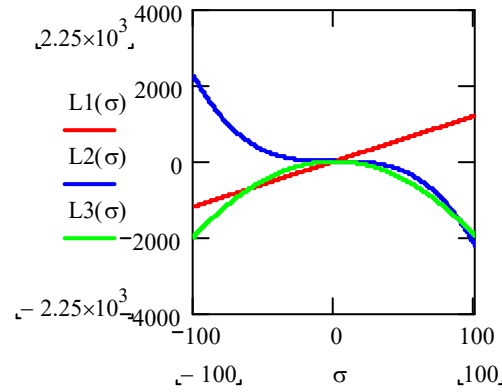


Figure 11: Functions of losses for the three alternatives

The subjective preferences functions are portrayed in the Figure 12.

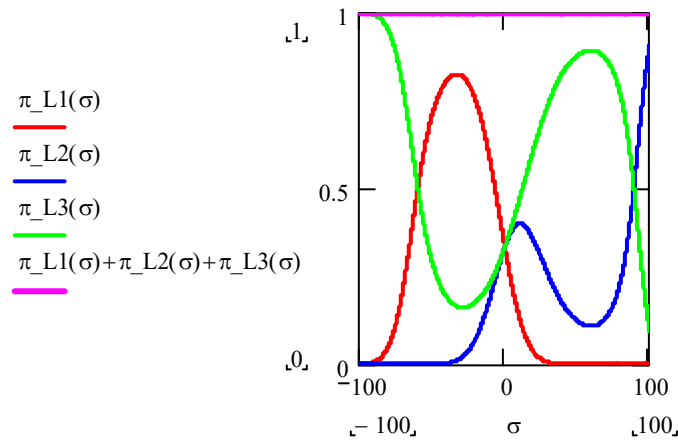


Figure 12: Functions of subjective preferences for the three alternatives

The normalizing conditions are also presented in the Figure 12.

Phase diagrams of the subjective individuals' preferences functions with regards to the corresponding harmfulness functions are illustrated in the Figure 13.

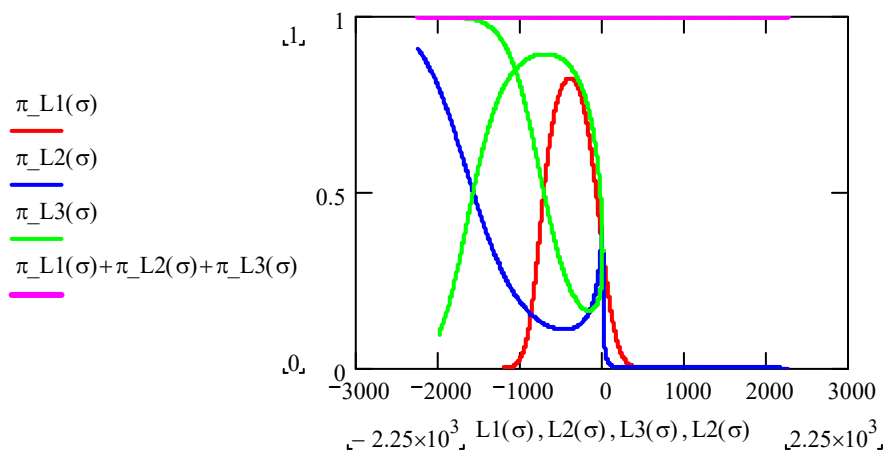


Figure 13: Phase diagrams of subjective preferences functions with respect to the losses functions for the three alternatives

The phase portrays as well as the check line for the normalizing conditions are, in fact, the corresponding projections of the curves within the three coordinates reference system (see and

compare the Figures 11 – 13). The traditional view subjective entropy of the individuals' preferences functions is shown in the Figure 14.

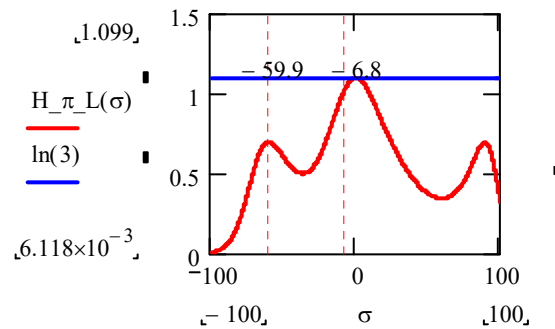


Figure 14: Traditional view subjective entropy of the individuals' preferences functions

The relative pseudo-entropy function in the case of the first alternative preference function domination is shown in the Figure 15.

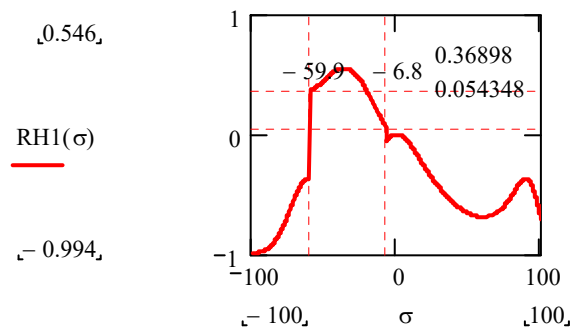


Figure 15: Relative pseudo-entropy function for the first alternative subjective individuals' preferences function prevailing

As for the second alternative preference function prevailing, the relative pseudo-entropy function is shown in the Figure 16.

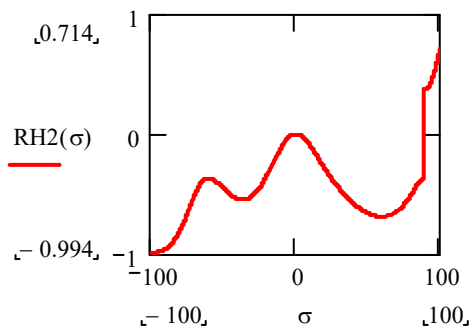


Figure 16: Relative pseudo-entropy function for the second alternative subjective individuals' preferences function dominance

The third alternative preference function domination impact upon the relative pseudo-entropy function is represented in the Figure 17.

The relative pseudo-entropy functions for all three cases of the three alternative preference functions domination are shown in the Figure 18.

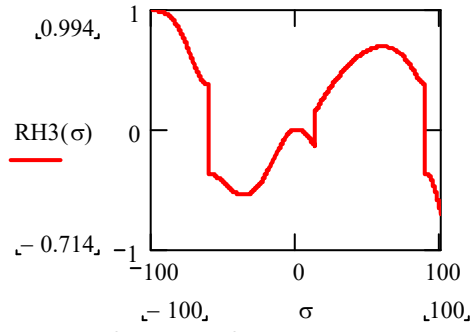


Figure 17: Relative pseudo-entropy function for the third alternative subjective individuals' preferences function prevalence

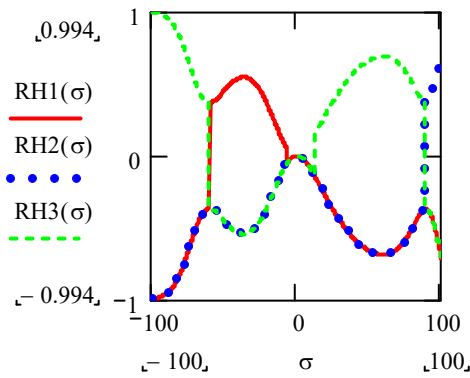


Figure 18: Relative pseudo-entropy functions for the three alternatives subjective individuals' preferences functions prevalence

The diagrams showing all curves of the case study are plotted the Figure 19.

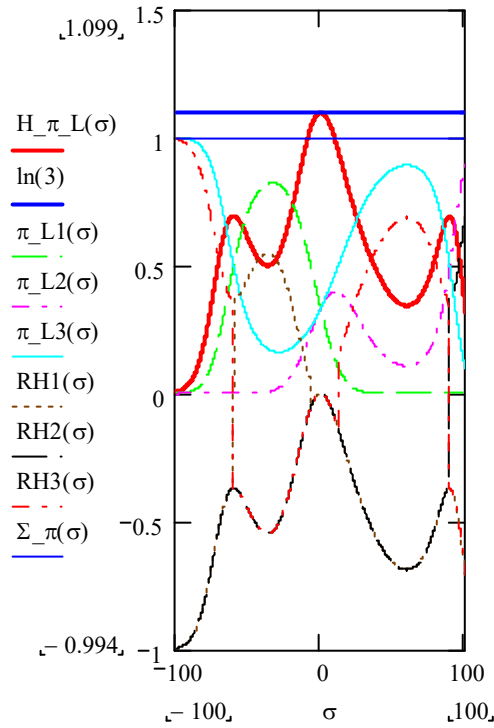


Figure 19: All curves of the case study

3. Discussion

Computer modeling of intelligent systems should be based upon the laws and regularities of the natural intellect functioning. One of such regularities is the available for an individual's alternatives choice subjective preferences optimal distribution in accordance with the postulated in subjective analysis theory of preferences principle which is called the subjective entropy maximum principle.

Active air transportation systems management elements have to include the corresponding subjective individual's preferences functions of the responsible decision making persons distributed with respect to the negative qualities of the achievable for the person's goals alternatives modeled with the objective intelligent air transportation management system functional (1). Subjective entropy (2) and normalizing constraint (3) are essential components of the objective functionals (1) and (4) for obtaining solutions (6) on conditions of (5).

However, the generally accepted character of the subjective preferences functions (6) dependence upon the functions of losses ("harmfulness") (illustrated in the Figure 1, [12]), is now broadened to the understandings of the modeled cases (7) – (15) (see the Figures 2 – 19). The point is that the subjective individuals' preferences functions (6) presumably measure the relative either harmfulness (losses) or the utilities ("usefulness") of the attainable alternatives due to the subjective preferences entropy conditional optimization principle.

That is why the shape of the phase diagrams for preferences to losses functions (illustrated in the Figure 1, [12]) is revealed to be correct only for the greatest of the linear case losses functions with respect to the determining parameter (see and compare the Figures 1 – 4 and Figures 11 – 13 correspondingly).

The proposed relative pseudo-entropy function (9) happened to be helpful in determining the relative certainty/uncertainty degree concerning prevailing/dominating preference (11). For instance, the hybrid relative combined pseudo-entropy function (9) (see the Figure 15) in the diapason of the determining parameter of $[-59.9...-6.8]$ shows the systems certainty increase with the following decrease because of the first alternative subjective preference function domination. Although the traditional view entropy (2) is incapable to distinguish such phenomenon (see and compare the Figures 15 and 14 in regards). For other diapasons the certainty/uncertainty varies in negative values, which means that the first alternative subjective preference function has no prevalence (13) (also see and compare the Figures 15 and 14 in correspondences). The same effectiveness of the proposed hybrid relative combined pseudo-entropy function (9) comparatively to the traditional view entropy (2) is visible in the Figures 14 and 16 – 19; as well as in the Figures 5 – 10. The relative value of the pseudo-entropy function (9), having "positive certainty": "+1", "negative certainty": "-1", and "uncertainty": "0" values, is more convenient rather than the bare value of the traditional view entropy (2). The same is observed for the utilities ("usefulness") subjective preferences functions entropy conditional optimization described with the formulas of (4) and (6).

4. Conclusion

Computer modeling of the intelligent air transportation management system functioning in the conditions of the operational alternatives subjective preferences uncertainty helps reveal the important phenomenon unknown before. That is the shape of the phase diagrams for the preferences to losses functions. The proposed hybrid combined relative pseudo-entropy function happened to be helpful in determining the relative certainty/uncertainty degree concerning prevailing/dominating subjective preferences functions.

Further studies should investigate more special cases of the intelligent systems multi-alternativeness.

5. References

- [1] M. J. Kroes, W. A. Watkins, F. Delp, R. Sterkenburg, Aircraft Maintenance and Repair, 7th. ed., McGraw-Hill, Education, New York, NY, 2013.

- [2] T. W. Wild, M. J. Kroes, Aircraft Powerplants, 8th. ed., McGraw-Hill, Education, New York, NY, 2014.
- [3] B. S. Dhillon, Maintainability, Maintenance, and Reliability for Engineers, Taylor & Francis Group, New York, NY, 2006.
- [4] D. J. Smith, Reliability, Maintainability and Risk. Practical Methods for Engineers, Elsevier, London, 2005.
- [5] R. D. Luce, D. H. Krantz, Conditional expected utility, *Econometrica* 39 (1971) 253–271.
- [6] R. D. Luce, Individual Choice Behavior: A theoretical analysis, Dover Publications, Mineola, NY, 2014.
- [7] E. T. Jaynes, Information theory and statistical mechanics, *Physical review* 106 (4) (1957) 620–630. doi:10.1103/PhysRev.106.620.
- [8] E. T. Jaynes, Information theory and statistical mechanics. II, *Physical review* 108 (2) (1957) 171–190. doi:10.1103/PhysRev.108.171.
- [9] E. T. Jaynes, On the rationale of maximum-entropy methods, *Proceedings of the IEEE* 70 (1982) 939–952. <https://ieeexplore.ieee.org/document/1456693>. doi:10.1109/PROC.1982.12425
- [10] F. C. Ma, P. H. Lv, M. Ye, Study on global science and social science entropy research trend, in: *Proceedings of the IEEE International Conference on Advanced Computational Intelligence, ICACI*, Nanjing, Jiangsu, China, 2012, pp. 238–242. <https://ieeexplore.ieee.org/document/6463159>. doi:10.1109/ICACI.2012.6463159
- [11] E. Silberberg, W. Suen, *The Structure of Economics. A Mathematical Analysis*, McGraw-Hill Higher Education, New York, NY, 2001.
- [12] V. Kasianov, Subjective Entropy of Preferences. Subjective Analysis, Institute of Aviation Scientific Publications, Warsaw, Poland, 2013.
- [13] O. Solomentsev, M. Zaliskyi, T. Herasymenko, O. Kozhokhina, Yu. Petrova, Data processing in case of radio equipment reliability parameters monitoring, in: *Proceedings of the International Conference on Advances in Wireless and Optical Communications, RTUWO*, Riga, Latvia, 2018, pp. 219–222. <https://ieeexplore.ieee.org/abstract/document/8587882>. doi:10.1109/RTUWO.2018.8587882.
- [14] V. Marchuk, M. Kindrachuk, Ya. Krysak, O. Tisov, O. Dukhota, Y. Gradiskiy, The mathematical model of motion trajectory of wear particle between textured surfaces, *Tribology in Industry (Tribol. Ind.)* 43 (2) (2021) 241–246. <https://doi.org/10.24874/ti.1001.11.20.03>.
- [15] D. Shevchuk, O. Yakushenko, L. Pomytkina, D. Medynskyi, Y. Shevchenko, Neural network model for predicting the performance of a transport task, *Lecture Notes in Civil Engineering*. 130 LNCE (2021) 271–278. doi:10.1007/978-981-33-6208-6_27.
- [16] S. Subbotin, The neuro-fuzzy network synthesis and simplification on precedents in problems of diagnosis and pattern recognition, *Opt. Mem. Neural Networks* 22 (2013) 97–103. <https://doi.org/10.3103/S1060992X13020082>.
- [17] A. V. Goncharenko, Multi-optional hybridization for UAV maintenance purposes, in: *Proceedings of the IEEE International Conference on Actual Problems of UAV Developments, APUAVD*, Kyiv, Ukraine, 2019, pp. 48–51. <https://ieeexplore.ieee.org/abstract/document/8943902>. doi:10.1109/APUAVD47061.2019.8943902.
- [18] A. V. Goncharenko, Active systems communicational control assessment in multi-alternative navigational situations, in: *Proceedings of the IEEE International Conference on Methods and Systems of Navigation and Motion Control, MSNMC*, Kyiv, Ukraine, 2018, pp. 254–257. <https://ieeexplore.ieee.org/abstract/document/8576285>. doi:10.1109/MSNMC.2018.8576285.
- [19] A. Goncharenko, A multi-optional hybrid functions entropy as a tool for transportation means repair optimal periodicity determination, *Aviation* 22 (2) (2018) 60–66. <https://journals.vgtu.lt/index.php/Aviation/article/view/5930>. <https://doi.org/10.3846/aviation.2018.5930>.
- [20] A. V. Goncharenko, Airworthiness support measures analogy to the prospective roundabouts alternatives: theoretical aspects, *Journal of Advanced Transportation* Article ID 9370597 2018 (2018) 1–7. <https://www.hindawi.com/journals/jat/2018/9370597/>. <https://doi.org/10.1155/2018/9370597>.

Optimization of File Distribution in Cloud-Based Data Storages

Ivan Muzyka¹, Dennis Kuznetsov¹, Yurii Kumchenko¹ and Anton Senko¹

¹ Kryvyi Rih National University, 11 Vitaliy Matusevych Street, Kryvyi Rih, 50027, Ukraine

Abstract

The article presents one of the approaches to optimizing the distribution of files across servers. Distributed information system operating in a global computer network with a mixed topology is considered. The authors prove that there is a certain correlation between traffic speed and network delay. This made it possible to use ping as a parameter that indicates the overall speed of the information system. The article proposes a mathematical model that uses two optimization criteria: the total cost of server maintenance and the weighted average ping. The cost criterion takes into account the total load on the server, the amount of additional storage, and the ability of the software to be parallelized. The authors showed the possibility of using genetic algorithms to find a suboptimal distribution. This approach works acceptably for small amounts of files and servers. However, for large volumes of data, the proposed heuristic algorithm gives an adequate result in a reasonable number of steps. The proposed model can be used for load balancing on servers that store huge amounts of audio and video files or provide online streaming services.

Keywords

File distribution, cloud storage, network latency, heuristic optimization, load balancing.

1. Introduction

Information technology has undergone huge changes over the past 5 years. The global challenges posed by pandemic COVID-19 have led to a significant increase of data in computer networks. Remote work and studying have led to increasing in video and audio content. Platforms such as Google Meet, Zoom, Microsoft Teams, YouTube as well as Viber, Telegram, WhatsApp messengers have become everyday tools not only for IT professionals but also for ordinary people. Cisco's Visual Networking Index (VNI) report demonstrated a significant increase in global IP traffic during 2018-2023, including video, online education, and entertainment [1]. Zoom has accumulated more than 3.3 trillion annual meeting minutes. 50 billion minutes of webinars are hosted on Zoom every year. Nowadays, the rapid development of cloud storage is driving up costs. Globalization and increase in traffic volumes require upgrading of network equipment. Every year, Internet Service Providers (ISPs) try to grow their bandwidth capacity. Despite reducing the cost of high-speed SSD drives to \$30-60/TB, the problems of cloud storage optimization remain relevant [2]. Software developers and database architects should take into account the problems that arise when there is no balancing between individual servers. Microservice architecture helps to divide a complex application into separate components. Thus, finding the optimal location of files in distributed information systems can significantly affect the cost of their support.

2. Related work

Optimizing distributed databases and content in cloud-based storages involves various strategies aimed at improving performance, scalability, reliability, and cost-effectiveness. Professor at University of Waterloo M. T. Özsu defined a distributed database as a collection of multiple, logically interrelated databases located at the nodes of distributed system [3]. A distributed

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ musicvano@knu.edu.ua (I. Muzyka); kuznetsov.dennis.1706@knu.edu.ua (D. Kuznetsov); kumchenko@knu.edu.ua (Y. Kumchenko); antonsenko@knu.edu.ua (A. Senko)

ORCID iD 0000-0002-9202-2973 (I. Muzyka); 0000-0002-2021-5207 (D. Kuznetsov); 0000-0001-9940-4854 (Y. Kumchenko); 0000-0002-4104-8372 (A. Senko)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

database management system (DBMS) is defined as the software system that permits the management of the distributed database and makes the distribution transparent to the users. It is important to note that a distributed DBMS is not just a collection of files that can be stored individually on servers. However, today, as a rule, unique identifiers of resources such as videos, sound files, image files, etc. are stored in the database. Unique keys are used to identify files. For example, Universally Unique Identifiers (UUID) are best suited for generating unique keys in distributed systems. At the same time, UUIDs provide an extremely low probability of collisions.

Often, to optimize data storage in distributed systems, researchers can use the following approaches [4] such as sharding and partitioning, load balancing, dynamic resource allocation etc. Dividing the dataset into smaller partitions (shards) distributed across nodes enhances parallel processing and reduces contention, leading to improved performance.

Some authors in their scientific works [5, 6] justify the feasibility of using heuristic or genetic algorithms to query optimization, etc. Professor G. G. Tselchik of Lviv University described in detail a series of mathematical models that can be used to optimize file distribution in information systems [7]. The main optimization criteria include the following:

- minimizing the time for processing queries in a distributed database;
- minimizing the traffic transmitted in the computer network and generated by the corresponding database queries;
- minimization of the cost of the distributed system operation, taking into account the hardware requirements for the nodes.

Other criteria may include the minimizing of electricity cost [8]. Overall, this research offers a valuable approach for Internet data centers (IDC) operators to optimize their energy costs while maintaining QoS in dynamic electricity markets.

There are a lot of modern researches concerning problems of effective data placement in cloud storages [9-11]. The article offers a comprehensive overview of data deduplication techniques and their applications in cloud storage, focusing on challenges and future directions. Also, authors tackle the problem of choosing the optimal location for storing data across geographically distributed cloud storage systems, considering cost and access latency. It's worth to note that there are methods based on using tiered cloud storage with dynamic data migration based on access patterns to optimize costs. However, insufficient attention has been paid to algorithms for redistributing content in a distributed system depending on changes in the intensity of user requests.

Thus, the scientific problems of researching distributed systems are quite relevant today and require further research, in particular with methods of parallelizing computing on the basis of high-performance computers.

3. Problem statement

Suppose we have a distributed video content storage system like YouTube, Facebook or TikTok. To ensure the operation of such an information system, many servers are needed around the world. Since data centers are located on different continents and serve requests from different countries, it is necessary to propose a model for optimal file distribution. Some videos may have different popularity in different regions. The number of views over a certain period of time will determine the intensity of requests.

However, the question arises whether placing files on a particular server can really affect the user experience. Obviously, a single server cannot serve hundreds of thousands of users streaming millions of videos. Some estimates suggest Google, which owns YouTube, might have over 1 million servers in its data centers worldwide [12]. However, this figure likely includes servers for all Google services, not just YouTube. Experts agree YouTube's infrastructure is highly scalable and continuously adapts to handle the ever-growing demand for video content. It likely utilizes a distributed network of data centers spread across various locations globally.

Let's look at the example of the cloud provider DigitalOcean to see how the physical distance from the user to the data center affects the network latency. DigitalOcean has datacenters across 9 regions [13]. Developers strongly suggest to pick a data center location geographically close to the end users who most frequently access your applications [14]. Figure 1 shows the locations of data centers around the world. The ping command demonstrates a clear correlation. The closer

the data center is located, the lower the delay. It is assumed that the communication line is of high quality.

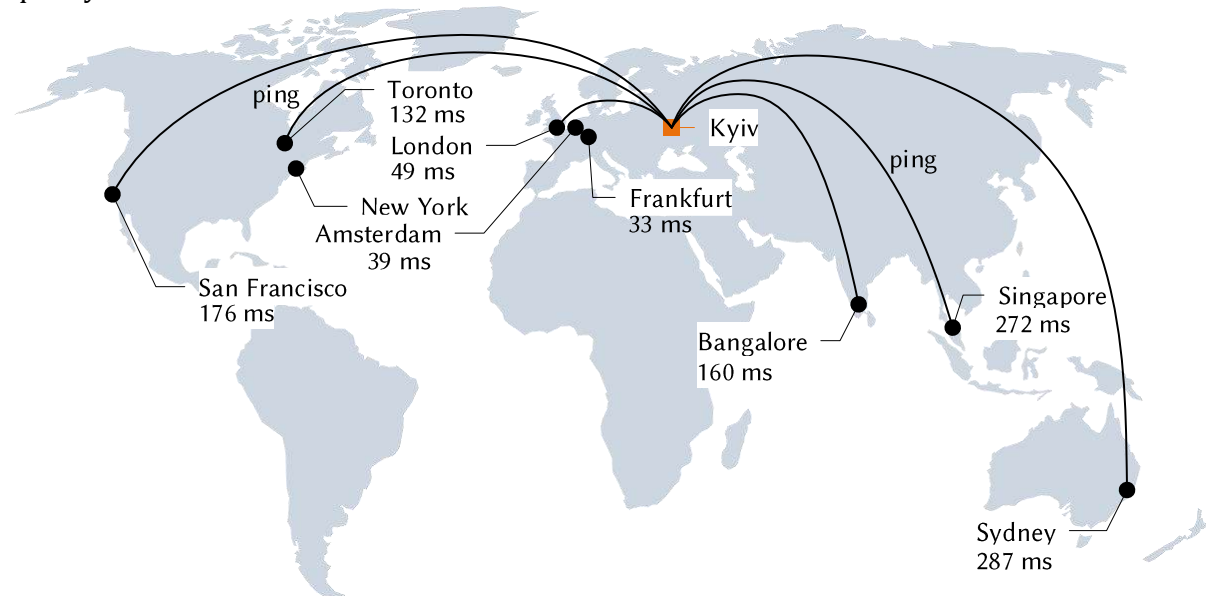


Figure 1: DigitalOcean data centers with network ping latency (tested from Ukraine)

Download speed and ping latency are related but not directly dependent on each other. They measure different aspects of network performance. Download speed refers to the rate at which data is transferred from the internet to your device. Download speed depends on various factors such as your internet service plan, network congestion, and the capabilities of your internet service provider. Ping, or latency, is the time it takes for a data packet to travel from your device to a server and back. Lower ping values indicate faster response times. Ping is influenced by the physical distance between your device and the server, as well as the efficiency of the network infrastructure. Let's try to consider this dependence in more detail.

According to the information collected by independent service Meter.net [15] there is some correlation between network latency and download speed. We used different data servers of Vultr cloud provider for testing (Table 1).

Table 1
Testing of data servers (Vultr.com)

Data center location	Country	Ping, ms	Download, Mb/s	Upload, Mb/s
Chicago	USA	82.44	29.15	14.61
Delhi	India	29.58	66.82	65.29
Frankfurt	Germany	20.83	58.99	74.93
Hawaii	USA	223	4.85	0.3
Johannesburg	South Africa	72	17	12.6
Melbourne	Australia	50.22	30.35	9.69
Mexico	Mexico	62.25	21.31	10.2
Osaka	Japan	24.4	103.3	252.6
Paris	France	16.75	103.8	154.5
Seoul	South Korea	16.3	175.1	417.9
Sydney	Australia	35.63	22.93	21.95

The graph in Figure 2 shows that as the ping delay increases, the data download speed decreases.

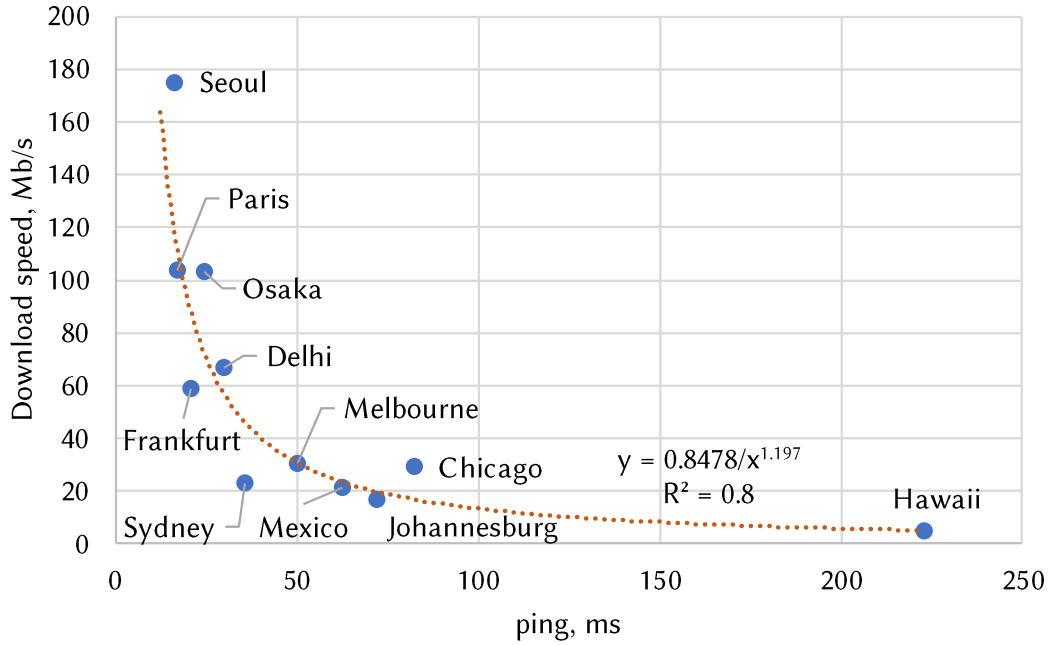


Figure 2: Dependency of download speed on ping network latency

Dependency of average download speed on ping latency can be described such empirical equation

$$v = \frac{0.8478}{\tau^{1.197}}, \quad (1)$$

where v – download speed, Mb/s, τ – ping latency, s. The coefficient of determination is $R^2=0.8$.

Thus, when creating a mathematical model to find the optimal file distribution, it is reasonable to use network latency as one of the parameters that directly affects the quality and performance of the system.

4. Proposed method

Let's have a look at a typical structure of a segment of a Wide Area Network (WAN), as shown in Figure 3. WANs support various applications such as data sharing, cloud access, file transfers, video conferencing, and more. Personal computers (PC1, PC2, ..., PC10, PC11, ..., PC30, PC31 etc.) are connected to the global network through switches (Sw1, Sw2, Sw4) and routers (R1, R4) of Internet service providers. In the vast and interconnected world of the global network, routers play the critical role of traffic directors, ensuring data reaches its intended destination efficiently and seamlessly. The servers (S1, S2, S3) are located in data centers that provide cloud services to users. Large facilities hosting major cloud providers or enterprise systems can have hundreds of thousands of servers, while smaller ones might have just a few hundred.

Many cloud providers (AWS, Google, Microsoft Azure) use different types of load balancers (LB1): hardware, software and cloud load balancers. Hardware load balancers (HLBs) are dedicated physical appliances designed to efficiently distribute traffic across multiple servers. Their robust hardware and specialized software offer high performance, security, and scalability, making them suitable for mission-critical applications and high-traffic websites. But load balancers make decisions based on CPU, RAM or network usage and don't guarantee the best user experience for sophisticated scenarios.

Databases (Db1, Db2, Db3) located on servers (S1, S2, S2) form a single distributed storage for the information system. Databases must support replication, transactions, multi-threaded operation, and handle high loads. Millions of files (F1, F2, ..., F20, ..., F120 etc.) can be stored in cloud storage on servers (S1, S2, S3). Users make requests to these files with different intensity. In the diagram, files from different servers are marked with different colors for better clarity.

It should be noted that the global computer network is not permanent and its characteristics are constantly changing throughout the day. Routing protocols use various metrics to ensure the speed and quality of information transmission. This means that network latency can vary widely over the period of a month. The ping value sometimes increases by 2-4 times when network problems occur.

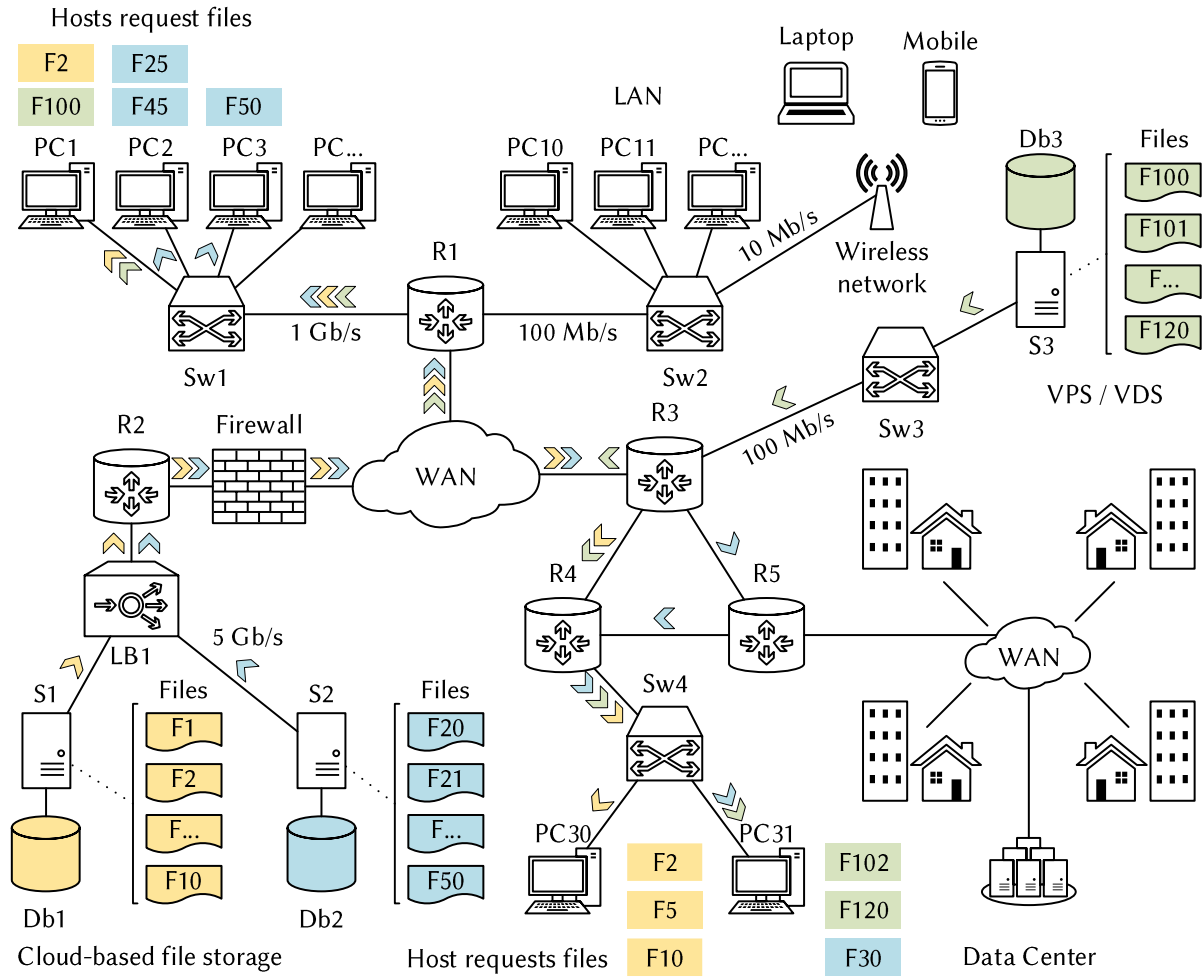


Figure 3: Typical structure of computer network with distributed file storage

Let's introduce the following notation:

n – the number of available working servers;

m – the total number of files (video, audio, images) to be stored by the distributed system;

p – the total number of users (personal computers / hosts);

S_i – the server with index i , $i = 1, 2, \dots, n$;

V_i – the size in GB of the server's data storage S_i , which is required to store files;

F_j – the file with index j , $j = 1, 2, \dots, m$;

L_j – size in GB of file with index j ;

H_k – the host with index k , $k = 1, 2, \dots, p$;

τ_{ki} – network latency (ping) from the host H_k to the server S_i ;

λ_{kj} – intensity of requests per unit of time from the host H_k to the file F_j ;

x_{ji} – Boolean value that indicates that the file F_j is stored on the server S_i ;

$$x_{ji} = \begin{cases} 1, & \text{if file } F_j \text{ is stored on the server } S_i \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

Q_i – cost of server maintenance with index i .

Obviously, the optimal file distribution is the one that minimizes the cost of all working servers and maximizes the speed of request processing

$$Q = \sum_{i=1}^n Q_i \rightarrow \min, \quad (3)$$

However, determining the cost of running one server is not as easy as it might seem at first glance. Most cloud providers build separate web applications that operate as a specialized calculator for estimating the cost depending on the hardware parameters of the server. Based on the information provided by the official website of Vultr company [16], we can conclude that the cost of a virtual server depends linearly on the number of processors (Figure 4). This approach allows us to conveniently scale the server performance when the number of requests changes.

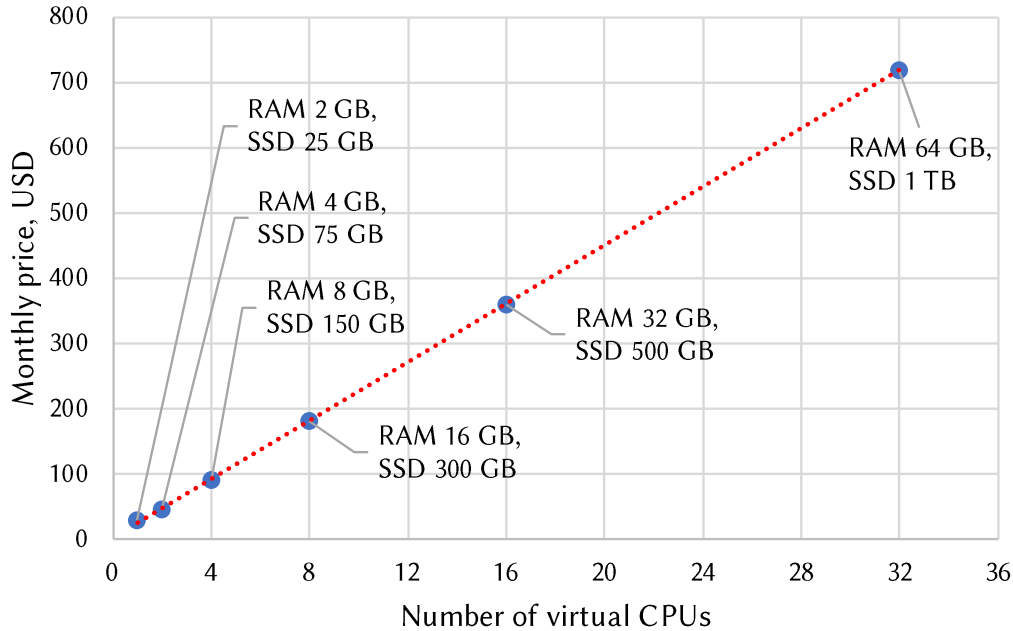


Figure 4: Dependency of VPS monthly cost on the server configuration

According to Amdahl's law, the speedup factor of the algorithm s , when the number of processors N_{CPU} for parallel query processing increases, is limited by the percentage of the program code α that is executed only in the sequential mode [17]

$$s = \frac{1}{\alpha + \frac{1 - \alpha}{N_{CPU}}} \quad (4)$$

Increasing the number of virtual processors in a server is more efficient the higher percentage of code that can be fully parallelized (Figure 5).

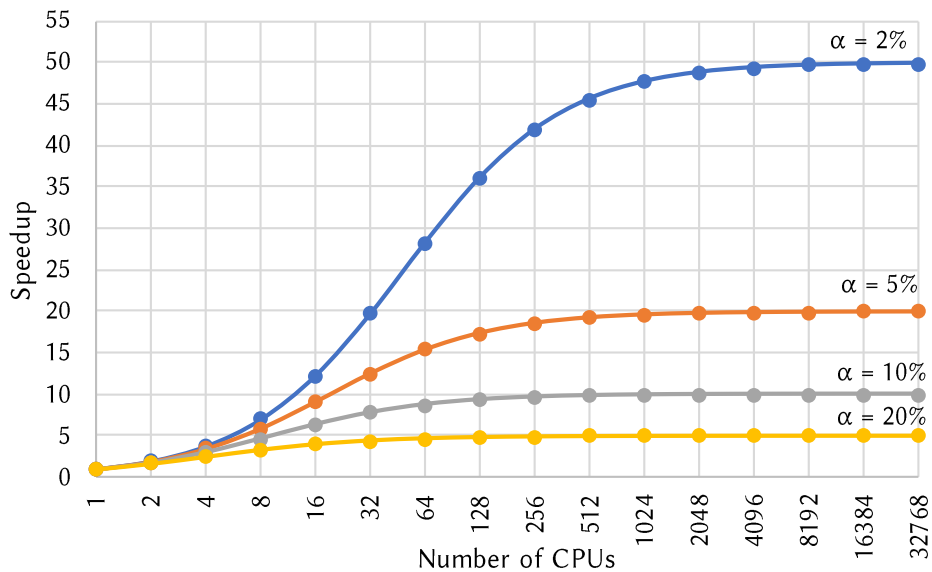


Figure 5: Dependency of program speedup on the number of virtual processors (cores)

If we assume that the speedup from using a more powerful server is the ratio of a certain desired request intensity λ_i to the base one λ_0 , then expression (4) can be written as follows

$$\frac{\lambda_i}{\lambda_0} \cong \frac{1}{\alpha + \frac{1-\alpha}{N_{CPUi}}} \Rightarrow N_{CPUi} \cong \frac{1-\alpha}{\frac{\lambda_0}{\lambda_i} - \alpha}. \quad (5)$$

The running cost of a server consists of many components, but for simplicity's sake, it is advisable to consider the most important components: the actual cost of a cloud server with a certain performance and additional data storage that allows you to store a large amount of files.

In addition to the standard server that processes the requests, additional storage is also required, which typically costs about \$100/TB. Therefore, it is possible to estimate the cost using the formula

$$Q_i \cong q_{CPU} \cdot N_{CPUi} + q_{SSD} \cdot V_i, \quad (6)$$

where $q_{CPU} = \$22/\text{core}$ – a coefficient that depends on cloud pricing and may be defined from a plot (Figure 4); $q_{SSD} = \$0.1/\text{GB}$ – typically cost of additional SSD blocks.

The total intensity of requests from all hosts to a particular server S_i can be calculated as following

$$\lambda_i = \sum_{k=1}^p \sum_{j=1}^m \lambda_{kj} x_{ji}. \quad (7)$$

With a given distribution of files x_{ji} and their sizes L_j , multiplying the vector L by the matrix x , we can get the total volume of certain server storage

$$V_i = \sum_{j=1}^m L_j x_{ji}. \quad (8)$$

Thus, the optimization criterion in its simplest form may look like this

$$Q \cong \sum_{i=1}^n \left(q_{CPU} \frac{(1-\alpha) \sum_{k=1}^p \sum_{j=1}^m \lambda_{kj} x_{ji}}{\lambda_0 - \alpha \sum_{k=1}^p \sum_{j=1}^m \lambda_{kj} x_{ji}} + q_{ssd} \sum_{j=1}^m L_j x_{ji} \right) \rightarrow \min, \quad (9)$$

The formula (9) shows that if the total intensity of requests to the server increases, an error and service denial may occur

$$\sum_{k=1}^p \sum_{j=1}^m \lambda_{kj} x_{ji} \geq \frac{1}{\alpha} \lambda_0. \quad (10)$$

Since each file F_j should be stored on one of the working servers only S_i , therefore

$$\sum_{i=1}^n x_{ji} = 1, j = 1..m. \quad (11)$$

Consequently, the problem of optimal file distribution on servers in a computer network with a mixed topology is to determine the following variables x_{ji} , where

$$x_{ji} = \{0 \cup 1\}, j = 1..m, i = 1..n. \quad (12)$$

However, the research conducted by the authors shows that it is not enough to have only one criterion for the operation of an information system that serves millions of users. It is also necessary to take into account the quality of service (QoS), speed of service, reliability indicators, etc.

Understanding QoS is crucial for designing and managing robust and efficient distributed information systems. By considering various aspects like performance, reliability, security, and dynamic environments, developers and administrators can implement effective QoS mechanisms and ensure optimal service delivery for users and applications. As Equation 1 shows, there is a certain correlation between the geographical distance from the client to the server, the speed of data transmission on the global network, and network latency. Therefore, the ping value can be used to estimate the data transfer rate. In the user's opinion the lower value of the average network delay, the higher quality of the information system. For example, when we talk about web browsing higher ping can lead to slower page loading times, causing delays and frustration. Even small increases in ping can be noticeable, especially on websites with a lot of content or interactive elements. High ping during the watching media can cause buffering and stuttering in

video or audio streaming, ruining the smooth playback experience. This is especially frustrating for high-resolution content or live streams.

Based on the above, it is necessary to introduce an additional parameter that will determine the weighted average delay in the system. This indicator should be minimal. Obviously, this parameter has a greater weight for large files and for servers with high request rates

$$T = \frac{\sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^p \tau_{ki} \lambda_{kj} L_j x_{ji}}{\sum_{j=1}^m \sum_{k=1}^p \lambda_{kj} L_j} \rightarrow \min. \quad (13)$$

The formulated challenge is similar to the linear programming problem described in paper [7]. However, it is still difficult to offer a clear algorithm for finding the optimal solution. Criterion (9) becomes minimal when the files are distributed evenly across all available servers, as this results in the lowest maintenance cost. Moving all the files to one super server is technically not possible, because there are channel bandwidth and memory capacity constraints. Besides, this solution is less reliable in terms of data transfer issues. Criterion (13) can be minimized by applying a simple principle: large files with a high intensity of requests should be placed on the server with minimal network latency (ping).

5. Modeling and results

A C# program was developed to simulate the distribution process. Using of .NET 8.0 and parallel LINQ technology made it possible to implement a stochastic method of searching for optimal values according to criteria (9) and (13). Authors have used genetic algorithm [18]. The initial distribution was random. The simulation was performed on a computer with AMD Ryzen processor (6 cores) and 8 GB of RAM.

Figure 6 shows the results of stochastic simulation. 20 files were distributed across 3 servers. The file sizes were in the range of 1.5-10 GB, and the ping from hosts to servers was in the range of 25-220 ms. The intensity of requests to the files did not exceed 40 requests per second. Each point on the graph represents a certain distribution of files and is described by two parameters: the total cost of maintaining all servers and the average network latency. Only 1000 points are shown on the graph for simplicity of presentation. However, if parallelization is used, the above computing system is capable of processing up to 1 million combinations per second. Given that the total number of all possible file arrangements on the servers is $n^m = 3^{20} \approx 3.48 \cdot 10^9$, this problem can be solved even by a full brute-force search and will take about 1 hour.

As shown in the graph, there are a large number of suboptimal file allocations that have approximately the same cost (within 5-10%), but the average latency of the system can vary within a significant range of 105-145 ms (almost 40%). This fact proves the need to apply the criterion (13) when making a decision.

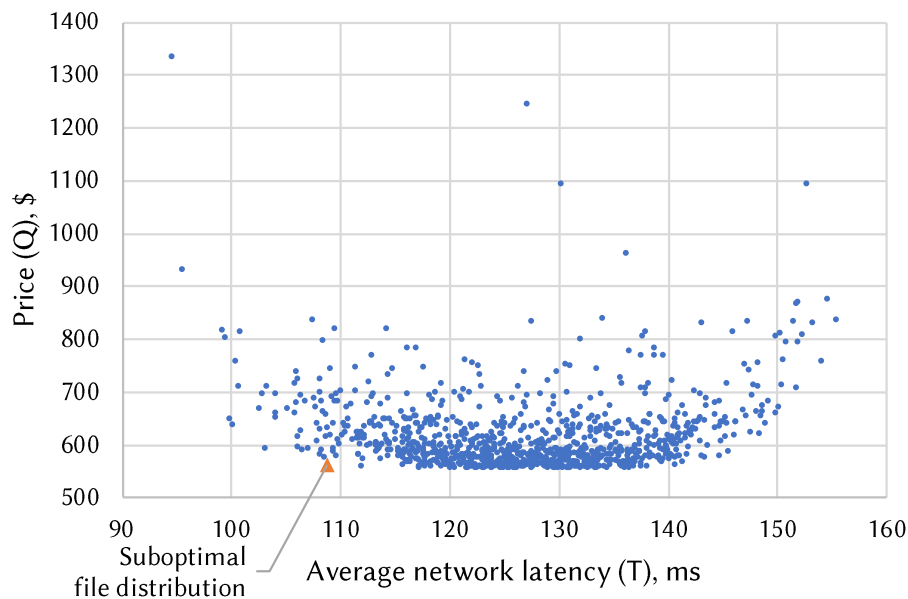


Figure 6: The set of random file distributions (1000 attempts)

It should be noted that a genetic algorithm can effectively find a suboptimal solution if it is applied in two stages. First, the minimum cost is sought, and then the algorithm starts looking for the minimum latency in the network. The cost can vary within 10% in this case. However, if the number of servers is increased to 10 and the number of files exceeds 10-20 thousand, the genetic algorithm may have poor convergence. The number of all possible combinations is too large to process in a reasonable time even 1% of all variations. Genetic algorithms are stochastic, meaning they rely on randomness. While they often find good solutions, there is no guarantee they will reach the absolute best one. They require evaluating many potential solutions across multiple generations, making them computationally expensive for complex problems. This can be a significant issue if resources are limited or real-time solutions are necessary. Therefore, it is worth considering an alternative approach to optimization.

At the initial stage, all files are placed on those servers that have the lowest ping. Let the weighted average network latency when placing a file F_j on the server S_i be determined by the formula

$$\bar{\tau}_{ji} = \frac{\sum_{k=1}^p \tau_{ki} \lambda_{kj} L_j}{\sum_{k=1}^p \lambda_{kj} L_j}. \quad (14)$$

Then we can calculate a matrix $\bar{\tau}$ from which it is easy to find the initial distribution X_0 . The ones ("1") should be in the positions where the latency value is minimal in each row of the matrix $\bar{\tau}$.

$$\bar{\tau} = \begin{bmatrix} 140 & 100 & 125 & 50 \\ 80 & 95 & 130 & 90 \\ 55 & 200 & 115 & 70 \\ \dots & \dots & \dots & \dots \\ 70 & 125 & 65 & 90 \end{bmatrix} \xRightarrow{\min} X_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (15)$$

For the initial allocation, we need to determine the initial cost, which is usually not optimal. Given the assumption that all servers in different parts of the world scale equally, a balanced load should be achieved. When the load is even, the cost is minimized.

In each line, we analyze the possibility of moving the file to another server. With such a rearrangement, it is necessary to keep the minimum possible increase in the value of $\Delta\tau$. It is worth moving the file that leads to a minimal increase in network latency, but the cost is reduced. This process of redistribution should be continued as long as the total cost is reduced. In general, this can be described by the following flowchart (Figure 8).

Figure 7 shows the result of finding the optimum. The input parameters remain the same as in the previous example. As you can see from the graph, the heuristic algorithm took no more than 20 steps to find an acceptable solution. The difference between the results does not exceed 5%.

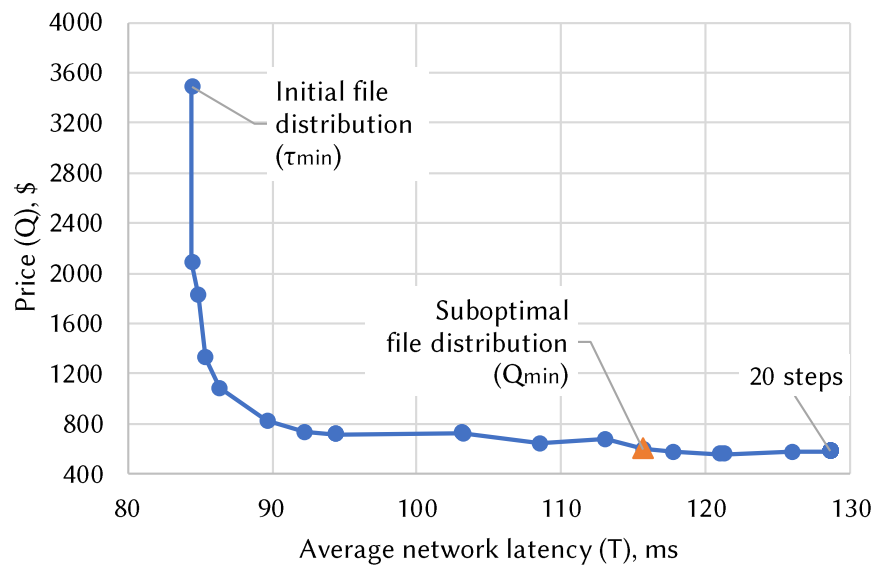


Figure 7: How the heuristic optimization algorithm works

It is important to note that by adjusting the parameters of the algorithm, we can get a compromise solution ($\tau \approx 90$ ms, $Q \approx 800$ \$). This option may be necessary to ensure a certain level of service quality.

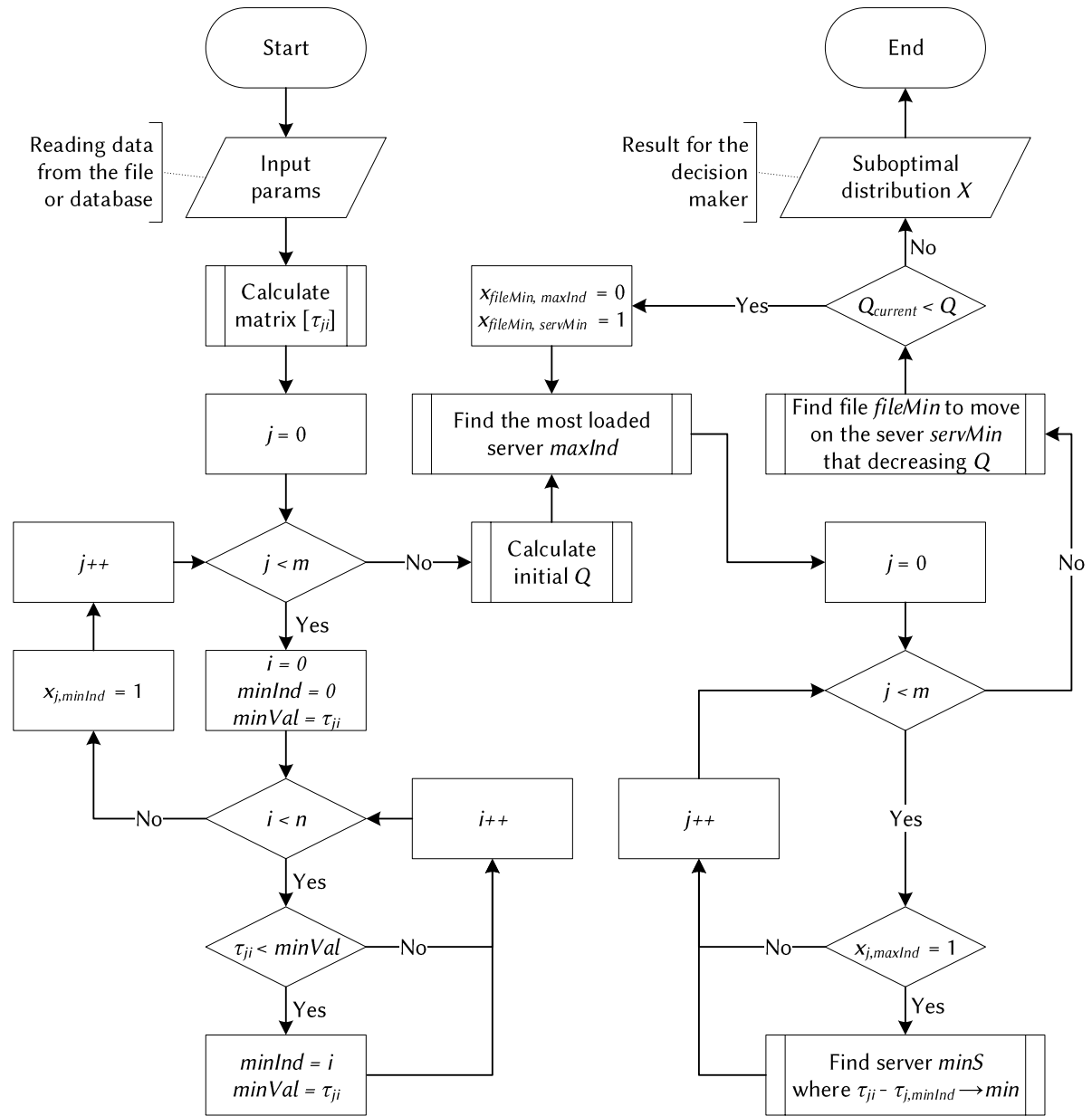


Figure 8: Flowchart of a heuristic algorithm for finding suboptimal file distribution

6. Conclusions

Thus, optimizing the placement of files in information systems is important. If there are several criteria, it is not always possible to combine them into one, because they have different units of measurement. The proposed heuristic algorithm has an advantage over the genetic optimization algorithm. It has a much higher speed, although it does not guarantee an absolute minimum. As shown in the graphs above, the error of the result does not exceed 5%.

Conducted research shows a significant variation in the weighted average network delay of more than 40%. At the same time, the cost can vary between 5-10%. This means that by moving files, you can improve the responsiveness of graphic user interface because high ping values significantly degrade the user experience.

An important aspect of optimization is reducing the load on the global computer network, since traffic passes through fewer hops and servers are loaded evenly.

Acknowledgements

The team of authors conducts research in various areas of information technology. The authors are sincerely grateful to the head of the department of Computer Systems and Networks, professor Andrey Kupin, the first vice-rector of Kryvyi Rih National University, associate professor Vladyslav Chubarov, and other colleagues. Valuable advice, organizational assistance, and support for research helped to obtain scientific results and publish this article. The authors are open for further fruitful cooperation.

References

- [1] Cisco.com, Cisco Annual Internet Report (2018–2023) White Paper, 2024. URL: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.
- [2] A. Piltch, SSD Prices Have Dropped 25% Since March, Now Average \$0.06 per GB, 2024. URL: <https://www.tomshardware.com/news/ssd-prices-sink-june-2023>.
- [3] M. T. Özsu, P. Valduriez, Principles of Distributed Database Systems, 4th ed., Springer, 2020.
- [4] A. Petrov, Database Internals: A Deep Dive into How Distributed Data Systems Work, 1st ed., O'Reilly Media, 2019.
- [5] J. Konečný, B. McMahan, D. Ramage, Federated Optimization: Distributed Optimization Beyond the Datacenter, 2015. ArXiv. doi:10.48550/arXiv.1511.03575.
- [6] E. Sevinç, A. Coşar, An Evolutionary Genetic Algorithm for Optimization of Distributed Database Queries. The Computer Journal, 54(5), 2011, pp. 717-725. <https://doi.org/10.1093/comjnl/bxp130>.
- [7] G. G. Tsehelyk, Distributed Database Systems, Lviv, Svit, 1990. 167 p.
- [8] Lei Rao, Xue Liu, Le Xie, Wenyu Liu, Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment, San Diego, CA, USA, 2010. doi: 10.1109/infcom.2010.5461933.
- [9] X. Ma, W. Yang, Y. Zhu and Z. Bai, A Secure and Efficient Data Deduplication Scheme with Dynamic Ownership Management in Cloud Computing, IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, TX, USA, 2022, pp. 194-201, doi:10.1109/IPCCC55026.2022.9894331.
- [10] P. Austria, C. H. Park, A. Hoffman and Y. Kim, Performance and Cost Analysis of Sia, a Blockchain-Based Storage Platform, 2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD), Zhuhai, China, 2021, pp. 98-103, doi:10.1109/BCD51206.2021.9581866.
- [11] K. Lee, J. Kim, J. Kwak and Y. Kim, Dynamic Multi-Resource Optimization for Storage Acceleration in Cloud Storage Systems, in IEEE Transactions on Services Computing, vol. 16, no. 2, pp. 1079-1092, 1 March-April 2023, doi:10.1109/TSC.2022.3173333.
- [12] Data Center Knowledge, How Many Servers Does Google Have? 2017. URL: <https://www.datacenterknowledge.com/data-center-faqs/google-data-center-faq>.
- [13] Digitalocean.com, Regional Availability Matrix, DigitalOcean Documentation, 2024. URL: <https://docs.digitalocean.com/products/platform/availability-matrix>.
- [14] K. K. Devleker, How to Choose a Data Center Location for Your Business, 2023. URL: <https://www.digitalocean.com/blog/choosing-a-data-center-location>.
- [15] Meter.net, VULTR.net – statistics, 2024. URL: <https://www.meter.net/test-server/102-vultr>.
- [16] Vultr.com, High Performance, High Frequency, Bare Metal, Affordable Cloud Computing, 2024. URL: <https://www.vultr.com/pricing/#optimized-cloud-compute>.
- [17] M. A. Noaman Al-hayanni, F. Xia, A. Rafiev, A. Romanovsky, R. Shafik, A. Yakovlev, Amdahl's law in the context of heterogeneous many-core systems – a survey, IET Comput. Digit. Tech., 2020, Vol. 14: pp. 133-148. <https://doi.org/10.1049/iet-cdt.2018.5220>.
- [18] A. Kupin, I. Muzyka, D. Kuznetsov, Y. Kumchenko, Stochastic Optimization Method in Computer Decision Support System, International Conference on Theory and Applications of Fuzzy Systems and Soft Computing. Advances in Intelligent Systems and Computing, 2018, Vol. 754. pp. 349-358.

Multi-Agent Reinforcement Learning Methods with Dynamic Parameters for Logistic Tasks

Eugene Fedorov¹, Olga Nechyporenko¹, Yaroslav Korpan¹ and Tetiana Neskoroedieva²

¹Cherkasy State Technological University, Shevchenko blvd., 460, Cherkasy, 18006, Ukraine

²Uman National University of Horticulture, Uman, Instytutska str., 1, Uman, 20305, Ukraine

Abstract

Part of Industry 4.0 is building computer systems by combining artificial intelligence with robotics. Such computer systems play an important role in the planning of cargo transportation in supply chain management. One of the approaches to building such computer systems is the use of multi-agent systems. The aim of the work is to create a methodology for constructing proactive agents based on reinforcement learning to solve the problem of optimal planning of cargo transportation. To solve the problem of insufficient efficiency of computer agents, the existing methods of statistical and machine learning were investigated. To date, the most efficient approaches to creating proactive agents are reinforcement learning approaches. The formalization of the functioning of proactive agents is performed. As a part of creating a model for the functioning of proactive agents based on reinforcement learning, a procedure for generating a quasi-optimal action plan is proposed that models the planning function of a proactive agent, which speeds up the decision-making process. Multi-agent reinforcement learning methods are proposed, which are close to random search at the initial iterations, and close to directed search at the final iterations. This is ensured by the use of dynamic parameters and allows the increase in the learning rate by approximately 10 times while maintaining the mean squared error of the method.

Keywords

supply chain management, multi-agent system, proactive agent, reinforcement learning, dynamic programming, Monte Carlo, Temporal-Difference Learning

1. Introduction

The fourth industrial revolution or Industry 4.0 has brought about rapid changes in technology, manufacturing and social processes in the 21st century due to increasing interconnection and intelligent automation. Part of this phase of industrial change is the integration of artificial intelligence with robotics, which blurs the boundaries between the physical, digital and biological worlds and is based on parallel and distributed computing [1].

Such computer systems play an important role in the planning of cargo transportation in supply chain management (CSM) [2] and audit [3-4]. One of the approaches to building such computer systems is the use of multi-agent systems.

Despite a large number of studies on the problem of improving the efficiency of supply chains and reducing logistics costs, some questions remain open. The complexity of supply chains is constantly increasing due to globalization and beyond. If earlier goods were purchased in centralized hypermarkets, now online trading is developing with its unique SCM stages.

The aim of the work is to create a methodology for constructing proactive agents based on reinforcement learning to solve the problem of optimal planning of cargo transportation. To achieve the goal, the following tasks were set and solved:

- formalization of the functioning of proactive agents;
- propose models for the functioning of proactive agents with a utility function based on reinforcement learning;
- propose a multi-agent reinforcement learning method with time difference and dynamic parameters;

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ fedorovee75@ukr.net (E. Fedorov); olne@ukr.net (O. Nechyporenko); y.korpan@chdtu.edu.ua (Y. Korpan); tytnesk1@gmail.com (T. Neskoroedieva)

ORCID 0000-0003-3841-7373 (E. Fedorov); 0000-0002-3954-3796 (O. Nechyporenko); 0000-0002-1455-5977 (Y. Korpan); 0000-0003-2474-7697 (T. Neskoroedieva)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

- propose a multi-agent reinforcement learning method based on Monte Carlo and dynamic parameters;
- propose a multi-agent reinforcement learning method based on adaptive dynamic programming and dynamic parameters.

2. Formulation of the research problem

The problem of increasing the efficiency of optimal cargo transportation planning comes down to the problem of finding such a set of plans $\{\pi_\mu\}$, that delivers a minimum of the mean square error (the difference between the cost of the resulting plan and the cost of the optimal plan),

$$F = \frac{1}{P} \sum_{\mu=1}^P (\mathcal{F}(\pi_\mu) - \mathcal{F}(\pi_\mu^*))^2 \rightarrow \min_{\{\pi_\mu\}},$$
 where P – power of multiple plans, π_μ – μ^{th} received plan, π_μ^* – μ^{th} optimal plan, $\mathcal{F}(\cdot)$ – cost function of the plan (for example, the length of the route in the case of the traveling salesman problem).

3. Literature review

Currently, the main types of computer agents of multi-agent systems are reactive and proactive agents [5-6].

Typically, a simple reactive agent has a set of behaviours (production rules), a database (stores its current state), and a knowledge base (stores its behaviours). A simple reactive agent makes a decision based on production rules. Each production rule consists of an antecedent (one or more perceptions) and a consequent (action).

Advantages of simple reactive agents [5-6]:

1. Simplicity of software implementation.
2. Ease of organization of multi-agent interaction.
3. High decision-making speed.
4. High probability of making the right decision.

Disadvantages of simple reactive agents [5-6]:

1. Simple reactive agents require a lot of information about their current state to determine an acceptable action.
2. Simple reactive agents do not take into account information about the current state of other agents.
3. Weak adaptability of simple reactive agents.
4. The behaviour of simple reactive agents is not based on a formal mathematical apparatus.
5. In the case of a large base of production rules, it is difficult to create a simple reactive agent.
6. Lack of logical inference leads to low autonomy.

Typically, a reactive agent with an internal state has a database (stores its current state), a knowledge base (stores knowledge about the world changes both independently and dependent on the agent's actions) and an inference engine. A reactive agent with an internal state makes a decision through logical inference.

Advantages of reactive agents with an internal state [5-6]:

1. The behaviour of reactive agents with an internal state is based on a formal mathematical apparatus (first-order predicate logic).
2. The presence of a logical inference leads to high autonomy.

Disadvantages of reactive agents with an internal state [5-6]:

1. Insufficient decision-making speed.
2. The complexity of organizing multi-agent interaction.
3. The complexity of the implementation of the perception function that maps signals from receptors into formulas in the language of first-order predicate logic.
4. The complexity of the formal description in the language of the first-order predicate logic of the dynamic environment.

Typically, a proactive agent has a database (stores information about its internal state, as well as the selected goal), a knowledge base (stores knowledge about the world changes both independently and dependent on the actions of the agent) and an inference engine. The proactive agent makes a

decision about choosing a goal from a set of possible goals and how to achieve it by forming an action plan based on logical inference. A proactive agent may also be based on a utility function.

The advantages and disadvantages of proactive agents and reactive agents with an internal state are practically the same [5-6].

Thus, the current problem is the low efficiency of the considered software agents.

At present, instead of expert systems with logical inference used in decision-making agents, reinforcement learning is actively used [7-8]. The main areas of single-agent reinforcement learning are:

- dynamic programming [9-10];
- adaptive dynamic programming [11-12];
- Monte Carlo [13-14];
- temporal-difference learning [15-16];
- policy-based methods [17-18];
- actor-critic methods [19-20].

Today, multi-agent methods are actively developed [21-22].

The advantages of reinforcement learning over inference are:

- no labeled data sets are required, this is especially relevant for large amounts of data [23-24];
- there is no imitation of a teacher, but a new solution can be proposed that people have not even thought about [25-26];
- the quality criterion / utility function is used [27-28].

Disadvantages of reinforcement learning based on dynamic programming [9-10]:

- a priori knowledge about the probabilities of transitions between states is required;
- action is not selected (for a fixed policy).

Disadvantages of reinforcement learning based on adaptive dynamic programming [11-12]:

- action is not selected (for a fixed policy);
- cannot directly optimize the policy;
- a large number of interactions between the agent and the environment;
- converges to the global optimum only in the case of a finite number of actions and states;
- susceptible to retraining.

Disadvantages of Monte Carlo based reinforcement learning [13-14]:

- action is not selected (for a fixed policy);
- cannot directly optimize the policy;
- a large number of long trajectories is required;
- updating the value of the cost function only after receiving the entire trajectory;
- does not always converge to the global optimum;
- susceptible to retraining.

Disadvantages of reinforcement learning based on temporal-difference learning [15-16]:

- the policy is fixed, so no action is selected (if TD-learning);
- cannot directly optimize the policy;
- a large number of interactions between the agent and the environment;
- converges to the global optimum only in the case of a finite number of actions and states;
- susceptible to undertraining (if one-step TD learning).

Disadvantages of policy-based reinforcement learning [17-18]:

- requires a large number of long trajectories;
- does not always converge to the global optimum;
- subject to retraining.

Disadvantages of actor-critic reinforcement learning [19-20]:

- a large number of long trajectories (if MC learning) or a large number of interactions between the agent and the environment (if TD learning);
- does not always converge to the global optimum (if MC learning) or converges to the global optimum only in the case of a finite number of actions and states (if TD learning);
- subject to retraining (if MC training);
- susceptible to undertraining (if one-step TD training).

4. Formalization of models of proactive agents functioning

For such agents, the internal state is called belief, the possible goal is called desire, the best goal is called intention.

Formalization of the functioning of a proactive agent.

Perception function (1)

$$see: E \rightarrow Per \quad (1)$$

maps the current state of the environment E into a new perception Per .

The state change function $next$ is called the belief change function brf (2)

$$brf: Bel \times Per \rightarrow Bel \quad (2)$$

and maps belief (internal state) Bel (belief) and perception Per into belief (internal state) Bel .

Changing the intention (the best goal) is the sequential execution of the function for selecting the set of wishes (possible goals) $options$ and the filtering function $filter$, which ensures the choice of the intention (the best goal) from the set of desires (possible goals).

Function to generate possible variants $options$ (3)

$$options: Bel \times Int \rightarrow Des \quad (3)$$

maps belief (internal state) Bel and intention (best goal) Int into a set of desires (possible goals) Des .

Filter function $filter$ (4)

$$filter: Bel \times Des \times Int \rightarrow Int \quad (4)$$

maps belief (internal state) Bel , a subset of desires (possible goals) Des and intention (best goal) Int to intention (best goal) Int .

Plan π is a sequence of actions

$$\pi = \{\alpha_1, \dots, \alpha_n\},$$

where each α_i is an element of the set Ac .

$Plan = \{\pi_0, \pi_1, \dots\}$ – set of all plans.

Instead of an action selection function $actic$ a new planning function plan is used (5)

$$plan: Bel \times Int \times Ac \rightarrow Plan \quad (5)$$

which maps a belief (internal state) Bel , an intention (best goal) Int , and a subset of actions Ac into $Plan$.

5. Modelling the functioning of proactive agents with a utility function through reinforcement learning

Let a utility function u assign a utility to a state and be represented as (6)

$$u(s(n)) = \max_{a \in A(s(n))} Q(s(n), a), \quad (6)$$

where $Q(s(n), a)$ – state-action cost function (profit in case of state $s(n)$ and action a),

$A(s(n))$ – set of actions available in state $s(n)$.

Let there be a memory of reproducing experiments (7)

$$M = \{ \langle s, a, R(s, a, s') \rangle \}, \quad (7)$$

where $R(s, a, s')$ – reward (reward for the transition from state s to state s' as a result of action a).

Then, for a proactive agent with a utility function, the procedure for generating an action plan π for the transition from the internal state (belief) s_0 to the target state (intention) s^* models the planning function plan and is presented in the following form.

1. Initialization

$$s(0) = s_0, \text{ iteration number } n = 1.$$

2. Choice of action (8) and observation of the internal state (9)

$$y(n) = \arg \max_{a \in A(s(n-1))} Q(s(n-1), a), \quad (8)$$

$$\exists \langle s, a, R(s, a, s') \rangle \in M : s(n-1) = s \wedge y(n) = a \rightarrow s(n) = s'. \quad (9)$$

3. Termination condition

If $n < N$, then $n = n + 1$, go to step 2, otherwise $\pi = (y(1), \dots, y(n))$.

The paper proposes reinforcement learning methods based on temporal-difference learning, based on Monte Carlo and based adaptive dynamic programming.

6. Multi-agent reinforcement learning with temporal-difference and dynamic parameters

The method consists of the following steps.

1. Initialization.

1.1. The maximum number of iterations N , the number of agents K , the maximum length of the states' sequence T , the discrete set of states S , the discrete set of actions $A(s)$, $s \in S$, the reward $R(s, a)$, $a \in A(s)$, $s \in S$, the parameters $\rho_1^{\min}, \rho_1^{\max}, \rho_2^{\min}, \rho_2^{\max}$ (control the learning rate), $0 < \rho_1^{\min} < \rho_1^{\max} < 1$, $0 < \rho_2^{\min} < \rho_2^{\max} < 1$, the parameters $\varepsilon^{\min}, \varepsilon^{\max}$ (control the ε -greedy policy), $0 < \varepsilon^{\min} < \varepsilon^{\max} < 1$, the parameters $\gamma^{\min}, \gamma^{\max}$ (control discounting), $0 < \gamma^{\min} < \gamma^{\max} < 1$ are set.

1.2. Reward tables are initialized for the k^{th} agent

$$Q_k = [Q_k(s, a)],$$

$$Q_k(s, a) = 0, a \in A(s), s \in S, k \in \overline{1, K}.$$

1.3. The reward table is initialized for a swarm of agents

$$Q_{\text{swarm}} = [Q_{\text{swarm}}(s, a)],$$

$$Q_{\text{swarm}}(s, a) = 0, a \in A(s), s \in S.$$

2. Iteration number $n=1$.

3. The parameters are calculated (10)-(13)

$$\rho_1(n) = \rho_1^{\max} - (\rho_1^{\max} - \rho_1^{\min}) \frac{n-1}{N-1}, \quad (10)$$

$$\rho_2(n) = \rho_2^{\max} - (\rho_2^{\max} - \rho_2^{\min}) \frac{n-1}{N-1}, \quad (11)$$

$$\varepsilon(n) = \varepsilon^{\max} - (\varepsilon^{\max} - \varepsilon^{\min}) \frac{n-1}{N-1}, \quad (12)$$

$$\gamma(n) = \gamma^{\min} + (\gamma^{\max} - \gamma^{\min}) \frac{n-1}{N-1}. \quad (13)$$

4. The number of the moment in time is set $t=1$.

5. The initial state s_{kt} , $k \in \overline{1, K}$ is observed for each k^{th} agent.

6. For each k^{th} agent, action a_{kt} is chosen using the ε -greedy policy π . If $U(0,1) < \varepsilon(n)$, then choose action a_{kt} randomly from the set of allowed actions $A(s_{kt})$, otherwise choose action a_{kt} in the form (14)

$$a_{kt} = \underset{b \in A(s_{kt})}{\operatorname{argmax}} Q(s_{kt}, b), \quad b \in A(s_{kt}), \quad (14)$$

i.e.

$$a_{kt} = \pi(s_{kt}), \quad k \in \overline{1, K}.$$

7. For each k^{th} agent, a reward $R(s_{kt}, a_{kt})$, $k \in \overline{1, K}$ is calculated.

8. For each k^{th} agent a new state $s'_{kt} = a_{kt}$, $k \in \overline{1, K}$ is observed.

9. For each k^{th} agent, the value of the combinations of the state-action cost functions of the swarm and the k^{th} agent is calculated, i.e.

$$Q_{\text{swarm}}(s_{kt}, a_{kt}) \text{ and } \tilde{Q}_k(s_{kt}, a_{kt}),$$

in the form (15)

$$\tilde{Q}_k(s_{kt}, a_{kt}) = (1 - \rho_2(n)) Q_{\text{swarm}}(s_{kt}, a_{kt}) + \rho_2(n) Q_k(s_{kt}, a_{kt}), \quad k \in \overline{1, K}. \quad (15)$$

10. For each k^{th} agent, the value of the cost function of the state-action $Q_k(s_{kt}, a_{kt})$ is calculated as (16)

$$Q_k(s_{kt}, a_{kt}) = \begin{cases} (1 - \rho_1(n)) \tilde{Q}_k(s_{kt}, a_{kt}) + \\ + \rho_1(n) \left(R(s_{kt}, a_{kt}) + \gamma(n) \max_{b \in A(s_{kt})} \tilde{Q}_k(s'_{kt}, b) \right), & t < T, \\ (1 - \rho_1(n)) \tilde{Q}_k(s_{kt}, a_{kt}) + \rho_1(n) R(s_{kt}, a_{kt}), & t = T \end{cases}, k \in \overline{1, K}. \quad (16)$$

11. Calculate the value of the cost function of the state-action of the swarm of agents $Q_{\text{swarm}}(s_{kt}, a_{kt})$ for each k^{th} agent in the form (17)

$$Q_{\text{swarm}}(s_{kt}, a_{kt}) = \begin{cases} \max_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}), & \left| \max_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| > \left| \min_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right|, \\ \min_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}), & \left| \max_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| \leq \left| \min_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right|, \end{cases} k \in \overline{1, K}. \quad (17)$$

12. For each k^{th} agent, the current state $s_{kt} = s'_{kt}$, $k \in \overline{1, K}$ is set.

13. If the current time is not the last, i.e. $t < T$, then increase the iteration number, i.e. $t = t + 1$, go to step 6.

14. If the current iteration is not the last one, i.e. $n < N$, then increase the iteration number, i.e. $n = n + 1$, go to step 3.

Note. Upon completion of the method, plan $\pi_k = (Y_{k1}, \dots, Y_{kT})$ is formed for each k^{th} agent (18)

$$Y_{kt} = \underset{a \in A(s_{kt})}{\operatorname{argmax}} Q_k(s_{kt}, a), s \in S, k \in \overline{1, K}. \quad (18)$$

The plan of the agent that satisfies the quality criterion better than others is selected.

7. Multi-agent Monte Carlo reinforcement learning with dynamic parameters

This method is presented in the following form.

1. Initialization.

1.1. The maximum number of iterations N , the number of agents K , the discrete set of states S , the discrete set of actions $A(s)$, $s \in S$, the reward $R(s, a)$, $a \in A(s)$, $s \in S$, the parameters $\rho_1^{\min}, \rho_1^{\max}, \rho_2^{\min}, \rho_2^{\max}$ (control the learning rate), $0 < \rho_1^{\min} < \rho_1^{\max} < 1$, $0 < \rho_2^{\min} < \rho_2^{\max} < 1$, the parameters $\varepsilon^{\min}, \varepsilon^{\max}$ (control the ε -greedy policy), $0 < \varepsilon^{\min} < \varepsilon^{\max} < 1$, the parameters $\gamma^{\min}, \gamma^{\max}$ (control discounting), $0 < \gamma^{\min} < \gamma^{\max} < 1$ are set.

1.2. Reward tables are initialized for the k^{th} agent

$$Q_k = [Q_k(s, a)]; \\ Q_k(s, a) = 0, a \in A(s), s \in S, k \in \overline{1, K}.$$

1.3. The reward table is initialized for a swarm of agents

$$Q_{\text{swarm}} = [Q_{\text{swarm}}(s, a)]; \\ Q_{\text{swarm}}(s, a) = 0, a \in A(s), s \in S.$$

1.4. Tables of the number of transitions for the k^{th} agent are initialized

$$D_k = [D_k(s, a)]; \\ D_k(s, a) = 0, a \in A(s), s \in S, k \in \overline{1, K}.$$

2. Iteration number $n = 1$.

3. The parameters are calculated (19)-(22)

$$\rho_1(n) = \rho_1^{\max} - (\rho_1^{\max} - \rho_1^{\min}) \frac{n-1}{N-1}, \quad (19)$$

$$\rho_2(n) = \rho_2^{\max} - (\rho_2^{\max} - \rho_2^{\min}) \frac{n-1}{N-1}, \quad (20)$$

$$\varepsilon(n) = \varepsilon^{\max} - (\varepsilon^{\max} - \varepsilon^{\min}) \frac{n-1}{N-1}, \quad (21)$$

$$\gamma(n) = \gamma^{\min} + (\gamma^{\max} - \gamma^{\min}) \frac{n-1}{N-1}. \quad (22)$$

4. Trajectory $\tau_k = (s_{k0}, a_{k0}, r_{k0}, \dots, s_{kT}, a_{kT}, r_{kT})$ is generated for each k^{th} agent, and $a_{kt} = \pi(s_{kt})$, $r_{kt} = R(s_{kt}, a_{kt})$, as a result of action a_{kt} a new state $s_{k,t+1}$ and reward r_{kt} , are observed, state s_{k0} can change at each iteration, the policy of choosing action π is ε -greedy, $k \in \overline{1, K}$.

5. Number of the moment in time $t = T$.

6. Calculate for each k^{th} agent the profit in the form of a discounted amount of reward from time t to time T (23)

$$R_{kt}(\tau_k) = \sum_{t'=t}^T \gamma^{t-t'}(n) r_{kt'}, \quad k \in \overline{1, K}. \quad (23)$$

7. For each k^{th} agent, the value of the combinations of the state-action cost functions of the swarm and the k^{th} agent is calculated, i.e.

$$Q_{\text{swarm}}(s_{kt}, a_{kt}) \text{ and } \tilde{Q}_k(s_{kt}, a_{kt}),$$

in the form (24)

$$\tilde{Q}_k(s_{kt}, a_{kt}) = (1 - \rho_2(n)) Q_{\text{swarm}}(s_{kt}, a_{kt}) + \rho_2(n) Q_k(s_{kt}, a_{kt}), \quad k \in \overline{1, K}. \quad (24)$$

8. For each k^{th} agent, the transition counter $D_k(s_{kt}, a_{kt})$ is increased, i.e.

$$D_k(s_{kt}, a_{kt}) = D_k(s_{kt}, a_{kt}) + 1, \quad k \in \overline{1, K}.$$

9. For each k^{th} agent, the value of the cost function of the state-action $Q_k(s_{kt}, a_{kt})$ is calculated as (25)

$$Q_k(s_{kt}, a_{kt}) = \left(1 - \frac{\rho_1(n)}{D_k(s_{kt}, a_{kt})} \right) \tilde{Q}_k(s_{kt}, a_{kt}) + \frac{\rho_1(n)}{D_k(s_{kt}, a_{kt})} R_{kt}(\tau_k), \quad k \in \overline{1, K}. \quad (25)$$

10. Calculate the value of the cost function of the state-action of the swarm of agents $Q_{\text{swarm}}(s_{kt}, a_{kt})$ for each k^{th} agent in the form (26)

$$Q_{\text{swarm}}(s_{kt}, a_{kt}) = \begin{cases} \max_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}), & \left| \max_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| > \left| \min_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| \\ \min_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}), & \left| \max_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| \leq \left| \min_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| \end{cases}, \quad k \in \overline{1, K}. \quad (26)$$

11. If $t > 0$, then $t = t - 1$, go to step 6.

12. If the current iteration is not the last one, i.e. $n < N$, then increase the iteration number, i.e. $n = n + 1$, go to step 3, otherwise stop.

Note. Upon completion of the method, plan $\pi_k = (y_{k1}, \dots, y_{kt}, \dots, y_{kT})$ is formed for each k^{th} agent (27)

$$y_{kt} = \underset{a \in A(s_{kt})}{\operatorname{argmax}} Q_k(s_{kt}, a), \quad s \in S, \quad k \in \overline{1, K}. \quad (27)$$

The plan of the agent that satisfies the quality criterion better than others is selected.

8. Multi-agent reinforcement learning method based on adaptive dynamic programming and dynamic parameters

The method consists of the following steps.

1. Initialization.

1.1. The maximum number of iterations N , the number of agents K , the maximum length of the states' sequence T , the discrete set of states S , the discrete set of actions $A(s)$, $s \in S$, the parameters $\rho_1^{\min}, \rho_1^{\max}, \rho_2^{\min}, \rho_2^{\max}$ (control the learning rate), $0 < \rho_1^{\min} < \rho_1^{\max} < 1$, $0 < \rho_2^{\min} < \rho_2^{\max} < 1$, the parameters $\varepsilon^{\min}, \varepsilon^{\max}$ (control the ε -greedy policy), $0 < \varepsilon^{\min} < \varepsilon^{\max} < 1$, the parameters $\gamma^{\min}, \gamma^{\max}$ (control discounting), $0 < \gamma^{\min} < \gamma^{\max} < 1$ are set.

1.2. Reward tables are initialized for the k^{th} agent

$$Q_k(s, a), \bar{Q}_k(s, a) = 0, a \in A(s), s \in S, k \in \overline{1, K}.$$

1.3. The reward table is initialized for a swarm of agents

$$Q_{\text{swarm}}(s, a), \bar{Q}_{\text{swarm}}(s, a) = 0, a \in A(s), s \in S.$$

1.4. The tables of the number of transitions for the k^{th} agent are initialized

$$D_k(s, a), \bar{D}_k(s, a) = 0, a \in A(s), s \in S, k \in \overline{1, K}.$$

1.5. The state observation quantity tables for the k^{th} agent are initialized

$$D_k(s), \bar{D}_k(s) = 0, s \in S, k \in \overline{1, K}.$$

2. Iteration number $n=1$.

3. The parameters are calculated (28)-(31)

$$\rho_1(n) = \rho_1^{\max} - (\rho_1^{\max} - \rho_1^{\min}) \frac{n-1}{N-1}, \quad (28)$$

$$\rho_2(n) = \rho_2^{\max} - (\rho_2^{\max} - \rho_2^{\min}) \frac{n-1}{N-1}, \quad (29)$$

$$\varepsilon(n) = \varepsilon^{\max} - (\varepsilon^{\max} - \varepsilon^{\min}) \frac{n-1}{N-1}, \quad (30)$$

$$\gamma(n) = \gamma^{\min} + (\gamma^{\max} - \gamma^{\min}) \frac{n-1}{N-1}. \quad (31)$$

4. The number of the moment in time is set $t=1$.

5. The initial state s_{kt} , $k \in \overline{1, K}$ is observed for each k^{th} agent.

6. For each k^{th} agent, action a_{kt} is chosen using the ε -greedy policy π . If $U(0,1) < \varepsilon(n)$, then choose action a_{kt} randomly from the set of allowed actions $A(s_{kt})$, otherwise choose action a_{kt} in the form (32)

$$a_{kt} = \underset{b \in A(s_{kt})}{\operatorname{argmax}} Q(s_{kt}, b), \quad b \in A(s_{kt}), \quad (32)$$

i.e.

$$a_{kt} = \pi(s_{kt}), \quad k \in \overline{1, K}.$$

7. For each k^{th} agent, a reward $R(s_{kt}, a_{kt})$, $k \in \overline{1, K}$ is calculated.

8. For each k^{th} agent a new state $s'_{kt} = a_{kt}$, $k \in \overline{1, K}$ is observed.

9. The transition function is calculated (33)

$$P_k(s'_{kt} | s_{kt}, a_{kt}) = \frac{D_k(s_{kt}, a_{kt})}{D_k(s_{kt})}, \quad (33)$$

and in advance

$$D_k(s_{kt}) = D_k(s_{kt}) + 1, \quad D_k(s_{kt}, a_{kt}) = D_k(s_{kt}, a_{kt}) + 1.$$

10. For each k^{th} agent, the value of the combinations of the state-action cost functions of the swarm and the k^{th} agent is calculated, i.e.

$$Q_{\text{swarm}}(s_{kt}, a_{kt}) \text{ and } \tilde{Q}_k(s_{kt}, a_{kt}),$$

in the form (34)

$$\tilde{Q}_k(s_{kt}, a_{kt}) = (1 - \rho_2(n))Q_{\text{swarm}}(s_{kt}, a_{kt}) + \rho_2(n)Q_k(s_{kt}, a_{kt}), \quad k \in \overline{1, K}. \quad (34)$$

11. For each k^{th} agent, the value of the cost function of the state-action $Q_k(s,a)$ is calculated as (35)

$$Q_k(s_{kt}, a_{kt}) = \begin{cases} (1 - \rho_1(n)) \tilde{Q}_k(s_{kt}, a_{kt}) + \\ + \rho_1(n) P_k(s'_{kt} | s_{kt}, a_{kt}) \left(R(s_{kt}, a_{kt}) + \gamma(n) \max_{b \in A(s_{kt})} \tilde{Q}_k(s'_{kt}, b) \right), & t < T, \\ (1 - \rho_1(n)) \tilde{Q}_k(s_{kt}, a_{kt}) + \rho_1(n) P_k(s'_{kt} | s_{kt}, a_{kt}) R(s_{kt}, a_{kt}), & t = T \end{cases}, \quad (35)$$

$k \in \overline{1, K}$

12. Calculate the value of the cost function of the state-action of the swarm of agents $Q_{\text{swarm}}(s_{kt}, a_{kt})$ for each k^{th} agent in the form (36)

$$Q_{\text{swarm}}(s_{kt}, a_{kt}) = \begin{cases} \max_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}), & \left| \max_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| > \left| \min_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| \\ \min_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}), & \left| \max_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| \leq \left| \min_{z \in \overline{1, K}} Q_z(s_{kt}, a_{kt}) \right| \end{cases}, \quad k \in \overline{1, K}. \quad (36)$$

13. For each k^{th} agent, the current state $s_{kt} = s'_{kt}$, $k \in \overline{1, K}$ is set.

14. If the current time is not the last, i.e. $t < T$, then increase the iteration number, i.e. $t = t + 1$, go to step 6.

15. If the current iteration is not the last one, i.e. $n < N$, then increase the iteration number, i.e. $n = n + 1$, go to step 3.

Note. Upon completion of the method, plan $\pi_k = (Y_{k1}, \dots, Y_{kT})$ is formed for each k^{th} agent (37)

$$Y_{kt} = \operatorname{argmax}_{a \in A(s_{kt})} Q_k(s_{kt}, a), \quad s \in S, \quad k \in \overline{1, K}. \quad (37)$$

The plan of the agent that satisfies the quality criterion better than others is selected.

9. Experiments and results

The numerical study of the proposed methods was carried out using the Python package.

For multi-agent reinforcement learning methods, the value of parameters $\rho_1^{\min} = 0.1, \rho_1^{\max} = 0.9$, $\rho_2^{\min} = 0.1, \rho_2^{\max} = 0.9$ (control the learning rate), parameters $\varepsilon^{\min} = 0.1, \varepsilon^{\max} = 0.9$ (control the ε -greedy policy), parameters $\gamma^{\min} = 0.1, \gamma^{\max} = 0.9$ (control discounting), the number of agents is $K = 20$.

The dependence of parameter $\gamma(n)$ is defined as

$$\gamma(n) = \gamma^{\min} + (\gamma^{\max} - \gamma^{\min}) \frac{n-1}{N-1}.$$

The dependence of parameter $\gamma(n)$ on the iteration number n is linear and shows that its share increases with the iteration number.

The dependence of parameter $\rho_1(n), \rho_2(n)$ and $\varepsilon(n)$ is defined as

$$\begin{aligned} \rho_1(n) &= \rho_1^{\max} - (\rho_1^{\max} - \rho_1^{\min}) \frac{n-1}{N-1}, \\ \rho_2(n) &= \rho_2^{\max} - (\rho_2^{\max} - \rho_2^{\min}) \frac{n-1}{N-1}, \\ \varepsilon(n) &= \varepsilon^{\max} - (\varepsilon^{\max} - \varepsilon^{\min}) \frac{n-1}{N-1}. \end{aligned}$$

The dependence of parameter $\rho_1(n), \rho_2(n)$ and $\varepsilon(n)$ on the iteration number n is linear; it shows that their share decreases with increasing iteration number.

The results of comparing the proposed temporal-difference reinforcement learning method with dynamic parameters and the traditional Q-learning method based on the mean squared error criterion

and the number of iterations for solving the travelling salesman problem (Berlin52 standard dataset), which is used for planning cargo transportation are presented in Table 1.

Table 1

Comparison of the proposed optimization method with the traditional Q-learning method

Mean squared error of the method		Number of iterations	
proposed	existing	proposed	existing
0.05	0.05	310	2030

The results of comparing the proposed Monte Carlo based reinforcement learning method with dynamic parameters and with the traditional every-visit method based on the mean squared error criterion and the number of iterations for solving the travelling salesman problem (Berlin52 standard dataset), which is used for planning cargo transportation are presented in Table 2.

Table 2

Comparison of the proposed optimization method with the traditional every-visit method

Mean squared error of the method		Number of iterations	
proposed	existing	proposed	existing
0.05	0.05	420	4050

The results of a comparison of the proposed reinforcement learning method based on adaptive dynamic programming with dynamic parameters and the traditional passive adaptive dynamic programming method based on the mean square error criterion and the number of iterations for solving the traveling salesman problem (Berlin52 standard dataset), which is used for freight planning, are presented in Table 3.

Table 3

Comparison of the proposed optimization method with the traditional passive adaptive dynamic programming method

Mean squared error of the method		Number of iterations	
proposed	existing	proposed	existing
0.05	0.05	110	1040

10. Discussion

Advantages of the proposed methods:

1. Modification of reinforcement learning methods due to dynamic parameters allows for an increase in the learning rate while maintaining the mean squared error of the method (Tables 1-3).
2. The use of a multi-agent approach makes distributed computing possible and increases the learning speed while maintaining the root-mean-square error of the method (Tables 1-3).
3. Reinforcement learning methods with dynamic parameters use the ϵ -greedy approach, which is close to random search at initial iterations, and close to directed search at final iterations. This is ensured by the use of dynamic parameters and allows for an increase in the learning rate while maintaining the mean squared error of the method (Tables 1-3).

11. Conclusions

To solve the problem of insufficient efficiency of computer agents, the existing methods of statistical and machine learning were investigated. These studies have shown that, to date, the most effective approaches to creating proactive agents are reinforcement learning approaches. The formalization of the functioning of proactive agents has been conducted.

As part of creating a model for the functioning of proactive agents based on reinforcement learning, a procedure for generating a quasi-optimal action plan is proposed that models the planning function of a proactive agent, which speeds up the decision-making process. Reinforcement learning methods are proposed, which at the initial iterations are close to random search, and at the final iterations are close to the directed search. This is ensured by the use of dynamic parameters and multi-agent approach and allows for an increase in the learning rate while maintaining the mean squared error of the method. The proposed multi-agent methods will be used for freight planning in supply chain management and auditing, and were investigated on a standard data set.

12. References

- [1] G. G. Shvachych, O. V. Ivaschenko, V. V. Busygin, Ye. Ye. Fedorov, Parallel computational algorithms in thermal processes in metallurgy and mining, *Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu*, 4 (2018) 129–137. doi: 10.29202/nvngu/2018-4/19.
- [2] O. Grygor, E. Fedorov, O. Nechyporenko, M. Grygorian, Neural network forecasting method for inventory management in the supply chain, in: *CEUR Workshop Proceedings*, 2022, volume 3137, pp. 14-27.
- [3] T. Neskorođieva, E. Fedorov, Method for automatic analysis of compliance of expenses data and the enterprise income by neural network model of forecast, in: *CEUR Workshop Proceedings*, 2020, volume 2631, pp. 145–158
- [4] T. Neskorođieva, E. Fedorov, I. Izonin, Forecast method for audit data analysis by modified liquid state machine, in: *CEUR Workshop Proceedings*, 2020, volume 2623, pp. 25-35.
- [5] G. Jezic, J. Chen-Burger, M. Kusek, R. Sperka, R. J. Howlett, L. C. Jain (Eds.), *Agents and multi-agent systems: technologies and applications*, volume 186 of *Smart innovation, systems and technologies*, 2020. doi: 10.1007/978-981-15-5764-4.
- [6] S. Russell, P. Norvig, *Artificial Intelligence: a Modern Approach*, Englewood Cliffs, NJ: Prentice Hall PTR, 2020.
- [7] A. L. C. Ottoni, E. G. Nepomuceno, M. S. de Oliveira, D. C. R. de Oliveira, Reinforcement learning for the traveling salesman problem with refueling, *Complex & Intelligent Systems*, 8 (2021) 2001-2015. doi: 10.1007/s40747-021-00444-4.
- [8] A. Oroojlooy, D. Hajinezhad, A review of cooperative multi-agent deep reinforcement learning, *Applied Intelligence*, 53 (2023) 13677–13722. doi:10.1007/s10489-022-04105-y.
- [9] D. Wang, X. Li, P. Xin, A. Liu, J. Qiao, Supplementary heuristic dynamic programming for wastewater treatment process control, *Expert Systems with Applications*, 247 (2024) 123280. doi: 10.1016/j.eswa.2024.123280.
- [10] U. Satic, P. Jacko, C. Kirkbride, A simulation-based approximate dynamic programming approach to dynamic and stochastic resource-constrained multi-project scheduling problem, *European Journal of Operational Research*, 315 (2024) 454–469. doi: 10.1016/j.ejor.2023.10.046.
- [11] H. Shen, Z. Li, J. Wang, J. Cao, Nonzero-sum games using actor-critic neural networks: A dynamic event-triggered adaptive dynamic programming, *Information Sciences*, 662 (2024) 120236. doi: 10.1016/j.ins.2024.120236.
- [12] K. Xie, Y. Zheng, Y. Jiang, W. Lan, X. Yu, Optimal dynamic output feedback control of unknown linear continuous-time systems by adaptive dynamic programming, *Automatica*, 163 (2024) 111601. doi: 10.1016/j.automatica.2024.111601.
- [13] J. Pascal, Artificial neural networks to solve dynamic programming problems: A bias-corrected Monte Carlo operator, *Journal of Economic Dynamics and Control*, 162 (2024) 104853. doi: 10.1016/j.jedc.2024.104853.
- [14] S. V. Albrecht, F. Christianos, L. Schäfer, *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*, MIT Press, Cambridge, MA, USA, 2023.
- [15] X. Chen, G. Yang, Sh. Yang, H. Wang, Sh. Dong, Ya. Gao, Online attentive kernel-based temporal difference learning, *Knowledge-Based Systems*, 278 (2023) 110902. doi:10.1016/j.knosys.2023.110902.
- [16] M. S. Stanković, M. Beko, S. S. Stanković, Distributed consensus-based multi-agent temporal-difference learning, *Automatica*, 151 (2023) 110922. doi:10.1016/j.automatica.2023.110922.

- [17] A. S. Stebenkov, N. O. Nikitin, Automated Generation of Ensemble Pipelines using Policy-Based Reinforcement Learning method, *Procedia Computer Science*, 229 (2023) 70-79. doi:10.1016/j.procs.2023.12.009.
- [18] F. Huang, X. Deng, Y. He, W. Jiang, A novel policy based on action confidence limit to improve exploration efficiency in reinforcement learning, *Information Sciences*, 640 (2023) 119011. doi:10.1016/j.ins.2023.119011.
- [19] J. Zhang, Sh. Han, X. Xiong, Sh. Zhu, Sh. Lu, Explorer-Actor-Critic: Better actors for deep reinforcement learning, *Information Sciences*, 662 (2024) 120255. doi:10.1016/j.ins.2024.120255.
- [20] Zh. Zhang, X. Liang, C. Chen, D. Liu, Ch. Yu, W. Li, Defense penetration strategy for unmanned surface vehicle based on modified soft actor-critic, *Ocean Engineering*, 304 (2024) 117840. doi: 10.1016/j.oceaneng.2024.117840.
- [21] T. Li, K. Zhu, N. C. Luong, D. Niyato, Q. Wu, Y. Zhang, B. Chen, Applications of multi-agent reinforcement learning in future Internet: A comprehensive survey, *IEEE Communications Surveys & Tutorials*, 24 (2) (2022) 1240–1279. doi:10.1109/COMST.2022.3160697.
- [22] L. M. Schmidt, J. Brosig, A. Plinge, B. M. Eskofier, C. Mutschler, An introduction to multi-agent reinforcement learning and review of its application to autonomous mobility, in: *IEEE 25th International Conference on Intelligent Transportation Systems*, 2022, pp. 1342–1349.
- [23] P. Yadav, A. Mishra, S. Kim, A comprehensive survey on multi-agent reinforcement learning for connected and automated vehicles, *Sensors*, 23 (10) (2023) 4710. doi:10.3390/s23104710.
- [24] J. Orr, A. Dutta, Multi-agent deep reinforcement learning for multi-robot applications: A survey, *Sensors*, 23 (7) (2023) 3625. doi.org/10.3390/s23073625.
- [25] L. Canese, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Re, S. Spanò, Multi-agent reinforcement learning: A review of challenges and applications, *Applied Sciences*, 11 (11) (2021) 4948. doi: 10.3390/app11114948.
- [26] Z. Xu, H. van Hasselt, M. Hessel, J. Oh, S. Singh, D. Silver, Meta-gradient reinforcement learning with an objective discovered, *arXiv:2007.08433*, 2020. doi:10.48550/arXiv.2007.08433.
- [27] H. Wang, E. Miah, M. White, M. C. Machado, Z. Abbas, R. Kumaraswamy, V. Liu, A. White, Investigating the properties of neural network representations in reinforcement learning, *Artificial Intelligence*, 330 (2024) 1-24. doi: 10.1016/j.artint.2024.104100.
- [28] F. Robertazzi, M. Vissani, G. Schillaci, E. Falotico, Brain-inspired meta-reinforcement learning cognitive control in conflictual inhibition decision-making task for artificial agents, *Neural Networks*, 154 (2022) 283–302. doi: 10.1016/j.neunet.2022.06.020.

A Comparison of the Effectiveness Architectures LSTM1024 and 2DCNN for Continuous Sign Language Recognition Process

Nurzada Amangeldy¹, Iurii Krak^{2,3}, Bekbolat Kurmetbek¹ and Nazerke Gazizova¹

¹ L.N. Gumilyov Eurasian National University, 2, Satpayev str., Astana, 010000, Republic of Kazakhstan

² Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska str. Kyiv, 01601, Ukraine

³ Glushkov Institute of Cybernetics, 40 Glushkov ave., Kyiv, 03187, Ukraine

Abstract

This study advances the field of real-time sign language recognition by assessing the efficacy of deep learning architectures, specifically those based on recurrent neural networks (RNNs) and convolutional neural networks (CNNs), for the intricate task of continuous sign language interpretation. Our research highlights the RNN-based model's superior capability in processing temporal gesture sequences, indicating its suitability for predicting dynamic sign language with high accuracy. In contrast, the CNN-based model demonstrates remarkable proficiency in identifying spatial patterns within gestures, proving its utility in visually complex situations. Employing advanced data processing techniques, such as real-time pose estimation and sophisticated array manipulation for gesture classification, this work signifies a leap forward in developing AI-powered assistive technologies. The study leverages a diverse dataset to ensure the models' robust performance, underscoring deep learning's role in promoting communication inclusiveness. This scientific exploration into the capacities of deep learning methodologies underscores our commitment to leveraging cutting-edge technology for fostering more inclusive communication solutions. This research is focusing on creating automated interpretation systems for Kazakh sign language, aiming at broader inclusivity and technological innovation in communication aids.

Keywords

Deep learning; sign language recognition; inclusive communication; human pose detection

1. Introduction

In recent years, interest in sign language recognition systems has grown significantly due to their potential to reduce communication barriers for people with hearing and speech impairments. Continuous sign language recognition is a complex task due to the high degree of variability in movements and the need for precise identification and analysis of multiple parameters, including hand, body, lip movements, and facial expressions.

In this study, we compare two architectures we previously proposed, based on Long Short-Term Memory LSTM1024 [1] and 2DCNN [2], designed to enhance the Continuous Sign Language Recognition (CSLR) process. Our approach begins with the initiation of the video capture process, during which either a specially installed camera or a standard video file can be utilized.

Using the MediaPipe Holistic model, our system detects and tracks key hand points in each video frame, creating a sequence necessary for further analysis. These sequences are then processed using either the LSTM1024 or 2DCNN models to predict specific gestures based on hand movements and overall body posture.

One of the key aspects of our approach is its ability to dynamically respond to input data: - if the probability of the predicted gesture exceeds the threshold we have established, that gesture is automatically integrated into the sentence being formed.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ nurzadaamangeldy@gmail.com (N. Amangeldy); yuri.krak@gmail.com (Iu. Krak); bigboss990930@gmail.com (B. Kurmetbek); g.nazerke2755@gmail.com (N. Gazizova)

🆔 0000-0002-4669-9254 (N. Amangeldy); 0000-0002-8043-0785 (Iu. Krak); 0009-0001-7510-2445 (B. Kurmetbek); 0009-0007-7278-4202 (N. Gazizova)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

All of this occurs in real-time, with feedback for the user via a demonstration window that displays the current sentence, the most recently recognized gesture, and the corresponding probability of its accuracy of recognition.

Through the comparative tests conducted, our system demonstrates significant progress in the field of CSLR, promising more efficient and inclusive communication for individuals using sign language. This research underscores not only a technological breakthrough but also the social importance of enhancing communication means for individuals with hearing impairments.

2. Relevant works review

Deep neural networks (DNNs) have been effectively used for sign language recognition, facilitating communication between individuals who use sign language and those who may not understand it. Sign language recognition involves translating sign language gestures or signs into text or spoken language.

CNNs, widely utilized for image-based sign language recognition, are proficient in extracting spatial information from gesture images or video frames, with architectures like VGG, ResNet, and Inception being particularly effective [1,3,4]. A novel approach is demonstrated in a study [5] that employs a sequential CSLR mechanism specifically for Japanese sign language. This method involves a two-step process: firstly, tense segments are identified automatically via a Random Forest algorithm, and secondly, a CNN performs word categorization within these segments. The system analyzes detailed 2D angular trajectories from the presenter's body, hands, and face, using data captured through the OpenPose framework. Tested under various conditions and with different word class labels, this mechanism has achieved notable accuracy rates, reaching 0.92 for segmentation and 0.91 for classification.

Visual Geometry Group (VGG) is known for its simplicity and effectiveness. VGG networks typically feature a uniform architecture with multiple convolutional layers followed by max-pooling layers. These networks have been adapted for sign language recognition by modifying the output layer to accommodate the specific number of sign language classes [6]. ResNet (Residual Networks) is notable for its deep network architecture with skip connections or residual blocks. These skip connections help combat the vanishing gradient problem, allowing the training of very deep networks. Sign language recognition models based on ResNet often leverage this architecture's ability to capture fine-grained spatial information in sign gestures [7].

One innovative approach is the SignBERT system [8], which tackles these challenges by utilizing high-quality video clips and intelligently selecting key frames. It employs a (3+2+1)D ResNet for visual feature extraction and a pre-trained BERT model for language modeling, incorporating partially obscured videos from isolated sign language datasets. The advanced SignBERT version includes hand images as additional input and uses an iterative learning strategy to fully leverage the system's potential on available datasets. The results demonstrate a significant enhancement in continuous sign language recognition (CSLR).

Inception (GoogLeNet): The Inception architecture, also known as GoogLeNet, introduced the inception modules that use multiple filter sizes. This architecture is particularly suitable for sign language recognition as it enables the model to capture spatial information at various scales. By adapting Inception, features from sign gesture images with different spatial resolutions can be efficiently extracted [9].

RNNs, including Long Short-Term Memory (LSTM)s [2] and Gated recurrent units (GRU)s [10], are pivotal for sign language recognition in videos, capturing temporal dynamics inherent in gestures. The Bi-ST-LSTM-A system [11] enhances this by eliminating sequence segmentation, utilizing dual-stream CNNs for comprehensive motion analysis, and merging data using ST-LSTM. Attention-focused Bi-LSTMs and bidirectional networks improve accuracy by emphasizing key frames and considering contextual information. These advancements underscore the importance of sophisticated temporal-spatial analysis and real-time data processing in translating sign language video sequences into text. Hybrid Models: the combination of CNNs and RNNs is prevalent in sign language recognition [12]. CNNs are employed for feature extraction from video frames, while RNNs are used to model the temporal dependencies between sequential frames. An alternative approach [13] presents a sequential CSLR system.

Another strategy [14] employs a cross-modal learning approach that utilizes textual information to discern sign language states and their tense boundaries from minimally annotated video sequences. This method uses a multimodal transformer to simulate intra-class dependencies, improving the network's

recognition accuracy. This technique has proven to surpass contemporary methods on various CSLR datasets, including RWTH-Phoenix-Weather-2014 [15], RWTH-Phoenix-Weather-2014T [16], and CSL [17].

3D Convolutional Neural Networks (3D CNNs): These networks, explicitly designed to capture both spatial and temporal information simultaneously, are particularly advantageous for video-based sign language recognition. 3D CNNs take into account the progression of sign gestures over time, rendering them highly suitable for this application [18]. Another innovative approach is the sign language recognition generative adversarial network (SLRGAN) [19], which utilizes a generative adversarial network framework. This system comprises a generator, which discerns sign language annotations by extracting spatial and temporal cues from video sequences, and a discriminator, which evaluates the quality of the generator's outputs by analyzing textual information related to sentences and glosses.

In conclusion, Deep Neural Networks (DNNs) have significantly propelled the field of sign language recognition forward, helping to bridge the communication divide between sign language users and those who do not understand it. This technological advancement plays a critical role in promoting inclusivity and accessibility in communication. The adaptation and implementation of various DNN architectures highlight the complexity and diversity inherent in interpreting sign language, underscoring that it is not merely a series of static gestures but a dynamic language rich in spatial and temporal components.

After a comprehensive review of various methodologies and deep learning architectures employed in sign language recognition, it becomes evident that the dynamic and complex nature of sign language poses unique challenges that demand innovative solutions. The exploration of different neural network models, including CNNs, LSTMs, and hybrid models, underscores the diversity of approaches in addressing spatial and temporal aspects of sign language. These models, each with their strengths in processing spatial information (like images) and temporal data (like sequences), have shown considerable promise in improving the accuracy and reliability of sign language recognition systems.

Based on this fundamental understanding, the problem statement in this work is the use and detailed comparative analysis of two neural networks - LSTM and CNN with known architectures for gesture recognition from a continuous video stream. The selection of these specific networks is driven by their proven effectiveness in our previous works for processing sequential data and image data, respectively, both of which are integral components of sign language communication.

3. Methodology

3.1. Data acquisition

Our approach commences with the real-time collection of video data via an integrated or external camera connected to the system. We employ high-resolution streaming video to enhance detection accuracy, targeting a frame rate of 60 frames per second for gestural words and 30 frames for the finger-spelling alphabet to ensure smoothness and minimize latency.

3.2. Video processing and pose detection using MediaPipe

Each frame from the real-time video feed undergoes processing using MediaPipe, a cutting-edge computer vision library, for human pose detection. MediaPipe offers pre-trained machine learning models capable of identifying various body landmarks including hands, facial features, and overall body posture (Figure 1).

We leverage this technology not only to pinpoint and visualize these key points on every processed video frame but also to capture comprehensive motion data with the aim of generating dynamic gestures in the future.

This extensive data acquisition is foundational for our system's ability to understand and interpret a more nuanced and natural form of sign language, which includes a combination of static poses and fluid, continuous motions.

3.3. Key points extraction

After the key points are identified on the frames, we extract the coordinates of these points, especially those related to hands and gestures (e.g., wrist position, finger placement, etc.). These data

are structured in a format suitable for input into a machine learning model and used as features for gesture recognition.

The `extract_keypoints` function in the script processes data from MediaPipe's pose detection, methodically extracting and structuring key points from various human body components into a unified NumPy array. It selectively retrieves x , y , z , and visibility values for general body pose landmarks, and, if specified, facial landmarks, alongside the coordinates for landmarks on both hands. These values are then flattened and concatenated. In scenarios where certain landmarks are undetected, the function intelligently fills in these gaps with zeros to maintain data consistency, crucial for subsequent machine learning operations. This comprehensive approach ensures a rich, uniform dataset, poised for advanced analyses and interpretations, particularly in gesture recognition applications.

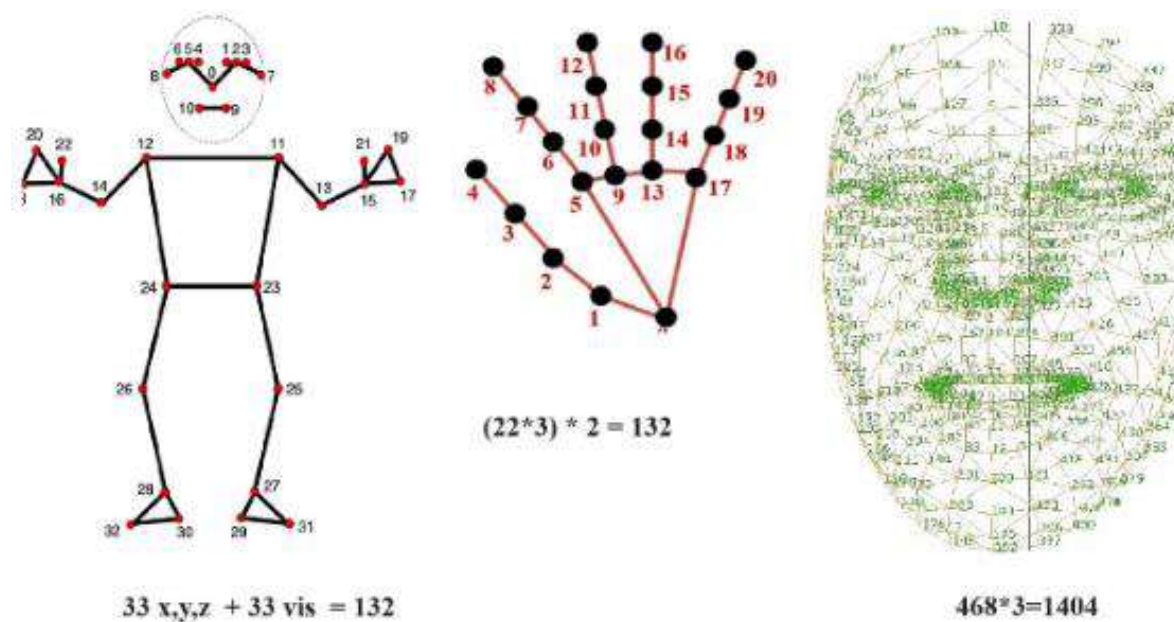


Figure 1: Various body landmarks

3.4. Gesture language recognition

In the enhanced explanation [1], the Convolutional Neural Networks (CNNs) designed for dynamic gesture recognition intricately balance the convolutional layers, kernel sizes, and pooling strategies to adeptly extract relevant features. This balance is pivotal in preventing the model from overfitting, a common challenge when the model complexity exceeds the necessity for accurate feature representation. The architecture incorporates layers specifically arranged to process and classify sign language gestures, employing a sequential approach that includes convolutional layers for feature extraction and dense layers for classification. The introduction of dropout techniques within the densely connected layers further mitigates overfitting by randomly ignoring a subset of neurons during training, enhancing generalization to unseen data.

The model's performance is critically analyzed using the categorical cross-entropy loss function, a standard in multiclass classification tasks, which quantifies the difference between the predicted probabilities and the actual distribution of the classes. This metric is instrumental in fine-tuning the model's parameters to minimize prediction error. The training process, supported by a substantial dataset comprising 30,960,000 data points across 40 dynamic gestures, demonstrates the model's ability to learn and generalize from the complex, high-dimensional input space. The dataset includes a comprehensive range of gestures, each represented by multiple frames and parameters, offering a rich basis for the model to learn the nuanced patterns of sign language.

Upon evaluation, the model exhibits remarkable accuracy in both validation and testing phases, confirming its robustness and the effectiveness of the chosen architecture and training regimen. This precision underscores the potential of 2DCNNs in bridging communication gaps for the deaf and hard-of-hearing community, providing a powerful tool for real-time sign language interpretation.

To delve deeper into the methodology [2], the model's LSTM layer, tailored with 1024 units, is pivotal for processing sequences inherent in sign language gestures, capturing temporal dependencies essential for dynamic gesture recognition. The dataset, meticulously curated from 1495 gesture samples, translates into 89,700 frames, providing a comprehensive foundation for model training. The inclusion of a Dropout layer, set at a 0.5 rate, strategically combats overfitting by nullifying half of the neuron activations, thereby forcing the model to learn more robust features.

The architecture concludes with a softmax-activated Dense layer, optimized for class distribution across gesture categories. This system, fine-tuned with the Adam optimizer and the cross-entropy loss function, undergoes rigorous evaluation to ensure accuracy and generalizability across diverse datasets, demonstrated by high performance metrics in training, testing, and validation phases. The methodology's thoroughness is further exemplified by the adoption of a Leave-One-Out cross-validation strategy, enhancing the model's reliability and applicability in real-world sign language interpretation scenarios.

The culmination of this research underscores the integration of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks as a robust methodology for dynamic gesture recognition in sign language. By balancing architectural elements to prevent overfitting and employing dropout and cross-entropy optimization, the models demonstrate exceptional ability to generalize from extensive training data. This approach not only achieves high accuracy in recognizing a wide range of gestures but also highlights the potential for real-time communication assistance for the deaf and hard-of-hearing community.

The meticulous dataset preparation, alongside rigorous validation strategies like Leave-One-Out cross-validation, further validates the models' effectiveness and applicability in practical scenarios, marking a significant advancement in assistive communication technologies.

3.5. Visualization and user interaction

The system is engineered for seamless real-time operation, ensuring the continuous intake of video input and delivering instant feedback to the user. It is meticulously structured around the following foundational steps, which form the core of its operational algorithm:

- Data acquisition: collect video data using an integrated or external camera, ensuring high-resolution and appropriate frame rates for capturing gestural words and finger-spelling alphabet.
- Video processing and pose detection: process each video frame with MediaPipe for human pose detection, identifying various body landmarks including hands, facial features, and body posture.
- Key points extraction: extract coordinates of key points related to hands and gestures from the video frames, structuring them in a format suitable for machine learning models.
- Gesture language recognition: use the extracted features to classify gestures employing LSTM1024 (for sequence data) and 2DCNN (for spatial structures in images) models, trained on a comprehensive dataset.
- Visualization and user interaction: display recognized gestures on the user interface alongside the video feed for real-time feedback, allowing users to interact with the system via a graphical interface.

This algorithm emphasizes real-time processing, dynamic response to input data, and inclusivity for individuals using sign language, leveraging advanced neural network architectures for accurate gesture recognition. Recognized gestures are displayed on the user interface alongside the video feed, providing an intuitive interaction for the user (Figure 2).

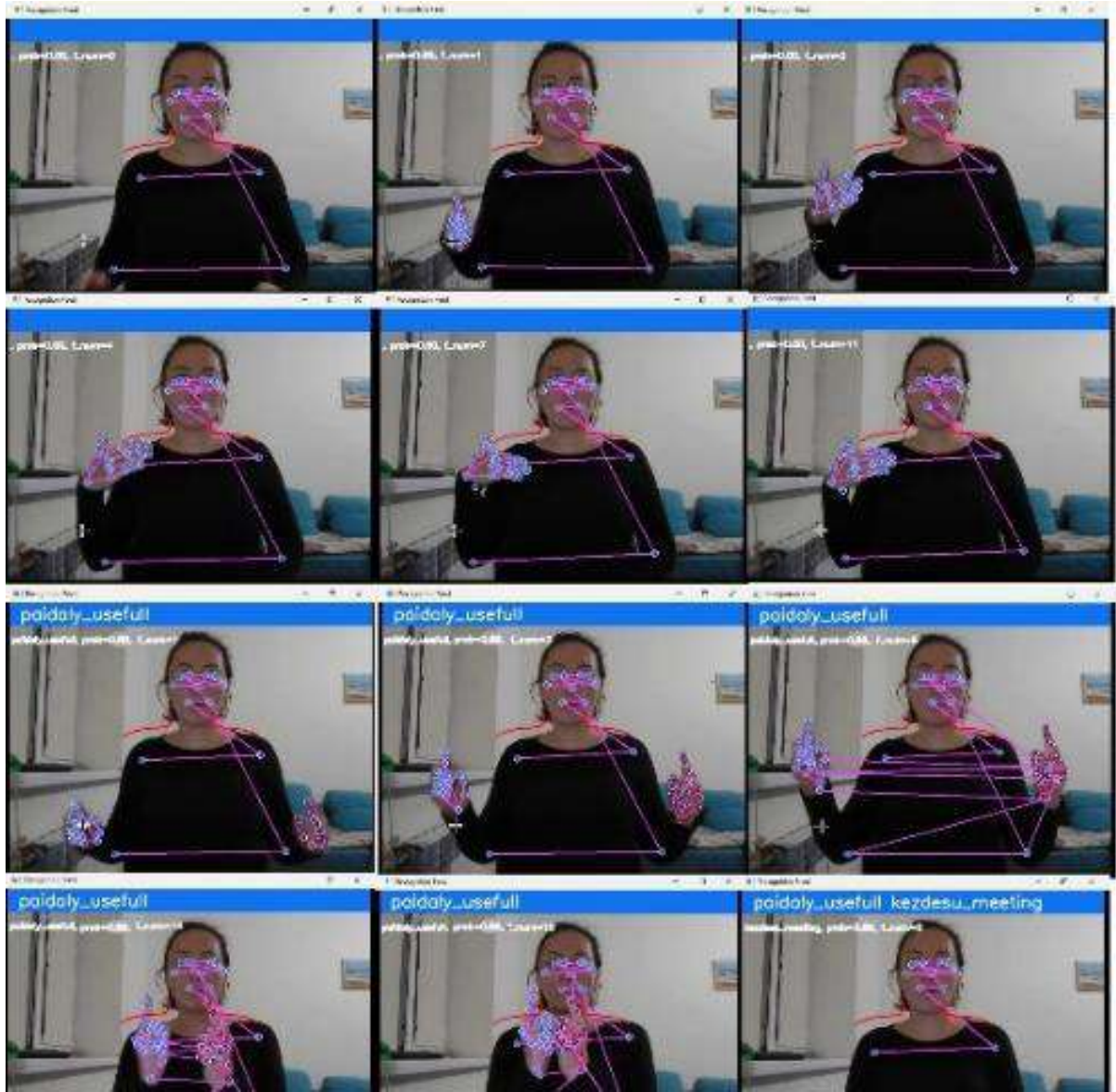


Figure 2: Current sentence consisting of words in the root form

Users can initiate or halt the gesture recognition process at their discretion and interact with the system through a simple graphical interface. The system's design prioritizes real-time operation, ensuring continuous video input and immediate feedback, as detailed in the algorithm below. This approach facilitates dynamic, instantaneous response to sign language gestures, offering an inclusive, accessible communication tool for the deaf and hard-of-hearing community.

Through a sophisticated process involving data acquisition, video processing, key points extraction, gesture language recognition, and interactive visualization, the system utilizes advanced neural network architectures to achieve precise gesture recognition.

This methodology underscores the importance of real-time processing and user interaction, showcasing the technology's capability to translate sign language gestures into text or speech instantaneously, providing an intuitive, user-friendly interface for effective communication.

4. Comparing methods for continuous gesture recognition: LSTM vs CNN

The performance variation between LSTM1024 and 2DCNN across the two datasets [1,2] suggests that the models respond differently to different data distributions. This is a common scenario in machine learning, where a model that performs well on one dataset may not necessarily perform as well on

another due to inherent differences in data characteristics (such as context, phrasing, language style, etc.).

It should be noted that the experiments were conducted on the NVIDIA GeForce RTX 3050 Laptop GPU, where training took from 5 to 10 minutes, and namely, the extraction of characteristics from the video stream and the preparation of data for the input of the neural network takes about 40 minutes.

To compare the performance of two different machine learning models, LSTM1024 and 2DCNN, on sign language words recognition tasks it is based on data from Table I and Table II.

Table 1

Kazakh sign language (KSL) dataset

No.	Sign word	LSTM1024			2DCNN		
		<i>P</i>	<i>R</i>	<i>F1-sc</i>	<i>P</i>	<i>R</i>	<i>F1-sc</i>
0.	Bring	1.00	1.00	1.00	1.00	1.00	1.00
1.	Mother	1.00	1.00	1.00	0.75	1.00	0.86
2.	Grand mother	0.75	1.00	0.86	1.00	1.00	1.00
3.	Affect	1.00	0.80	0.89	1.00	1.00	1.00
4.	Children	0.75	1.00	0.86	1.00	0.67	0.80
5.	Belt	0.83	1.00	0.91	1.00	0.60	0.75
6.	Room	1.00	1.00	1.00	1.00	0.80	0.89
7.	Medications	1.00	1.00	1.00	0.75	1.00	0.86
8.	Good	1.00	1.00	1.00	1.00	1.00	1.00
9.	Light	1.00	0.80	0.89	1.00	1.00	1.00
10.	Nephew	0.62	1.00	0.77	0.67	0.40	0.50
11.	Meeting	1.00	1.00	1.00	0.71	1.00	0.83
12.	Rejoice	0.83	1.00	0.91	1.00	0.80	0.89
13.	Blanket	1.00	1.00	1.00	1.00	1.00	1.00
14.	Neighbor	1.00	1.00	1.00	1.00	1.00	1.00
15.	Sit down	0.75	0.60	0.67	0.60	0.60	0.60
16.	Usefull	1.00	0.80	0.89	0.62	1.00	0.77
17.	Teapot	0.80	0.80	0.80	0.80	0.80	0.80
18.	Call	1.00	0.33	0.50	0.75	1.00	0.86
19.	Fast	1.00	0.80	0.89	0.75	0.60	0.67
accuracy		0.89			0.85		
macro avg		0.92	0.90	0.90	0.87	0.86	0.85
weighted avg		0.92	0.90	0.90	0.87	0.85	0.85

Each row represents a different sign word, and the models' performance is measured in terms of:

Precision P ($P = TP / (TP + FP)$),

Recall R ($R = TP / (TP + FN)$),

F1-score ($F1-sc = 2 * P * R / (P + R)$).

Here TP – precision ration of true positive; FP – precision ration of faulse positive; FN – precision ration of faulse negative.

Both models exhibit strong performance across most sign words, with high scores in precision, recall, and F1-score. However, there are variances on a per-sign-word basis.

In the Table I, LSTM1024 has higher average precision, recall, and F1-score than 2DCNN, indicating it may be better at recognizing those particular sign words.

In the Table II, both models are mostly comparable, though 2DCNN has a slightly higher overall accuracy.

Certain words like "Nephew" in the first table and "Continuation" in the second table show significant differences in performance between the two models, indicating that one model may have learned the features for these signs better than the other. The "Call" sign in the first table is a clear weak point for the LSTM1024 model, with a recall of 0.33, indicating it frequently misses this sign.

The overall val_accuracy is high for both models, but remember, accuracy might not be a sufficient measure when the classes are imbalanced.

For both models, the performance metrics for both training and testing are 0.99. This demonstrates exceptional accuracy and indicates that the models have adapted superbly to the provided data. Within the experiment outlined in articles [3] and [9], we encountered overfitting.

However, by employing an early stopping strategy, we achieved the results reflected in the Table I.

When applied to continuous recognition tasks, the LSTM model exhibited superior performance, achieving the highest values in average probabilities compared to other models. This underscores the model's robustness and advanced capability in handling sequential data, enhancing its reliability for real-time, continuous sign language recognition [2].

Table 2

Continuous Kazakh Sign Language (KSL) Dataset

No.	Sign word	LSTM1024			2DCNN		
		<i>P</i>	<i>R</i>	<i>F1-sc</i>	<i>P</i>	<i>R</i>	<i>F1-sc</i>
1	Pity	1.00	1.00	1.00	1.00	1.00	1.00
2	Bring	1.00	1.00	1.00	1.00	0.80	0.89
3	Drag	0.62	1.00	0.77	0.62	1.00	0.77
4	Influence	1.00	1.00	1.00	1.00	1.00	1.00
5	Skirt	1.00	0.60	0.75	1.00	0.80	0.89
6	Continuation	1.00	0.40	0.57	1.00	0.80	0.89
7	Nephew	0.83	1.00	0.91	1.00	1.00	1.00
8	Forgive	1.00	0.80	0.89	0.80	0.80	0.80
9	Husband	1.00	0.80	0.89	0.75	0.60	0.67
10	Loading	0.83	1.00	0.91	1.00	1.00	1.00
11	Bring	0.50	1.00	0.67	1.00	1.00	1.00
12	Mother	1.00	1.00	1.00	0.75	1.00	0.86
13	ter	0.75	1.00	0.86	0.75	1.00	0.86
14	Disease	1.00	1.00	1.00	1.00	1.00	1.00
15	Children	1.00	1.00	1.00	1.00	1.00	1.00
16	Medicine	1.00	1.00	1.00	1.00	0.67	0.80
17	Meat	1.00	1.00	1.00	1.00	1.00	1.00
18	Neighbor	1.00	1.00	1.00	1.00	0.67	0.80
19	The invitation	1.00	0.33	0.50	1.00	1.00	1.00
20	Dating	1.00	1.00	1.00	0.75	1.00	0.86
accuracy		0.89			0.90		
macro avg		0.93	0.90	0.89	0.92	0.91	0.90
weighted avg		0.93	0.89	0.89	0.92	0.90	0.90

1. "Көрші шәйнекті алып келді." ("The neighbor brought a kettle.") - accuracy 0.88
2. "Анам шаттанды." ("My mother was happy.") - accuracy 0.91
3. "Балалар жарық бөлмеде отыр." ("The children are sitting in a bright room.") - accuracy 0.89
4. "Апам күйеу баласын қуана шақырды." ("My mother happily invited her son-in-law.") - accuracy 0.85
5. "Тәтеме дәрі тез әсер етті." ("The medicine affected my aunt quickly.") - accuracy 0.89
6. "Апам жақсы белдемшені алып келді." ("My mother brought a nice skirt.") - accuracy 0.88
7. "Анам балалардың көрпесін әкелді." ("My mother brought the children's blankets.") - accuracy 0.82
8. "Апам жиенін кездесуге апарды." ("My mother took her nephew to the meeting.") - accuracy 0.90

5. Conclusion

In summary, the performance of LSTM1024 and 2DCNN models varies with specific sign words, indicating that the nature of these models influences their effectiveness. LSTM networks excel in recognizing dynamic gestures due to their ability to learn order dependence in sequence prediction problems, a crucial aspect of continuous sign language recognition.

Conversely, 2D CNNs, primarily utilized for image data, show heightened effectiveness with static signs or in datasets rich in spatial features.

Despite these differences, the LSTM model stands out for its exceptional efficiency and precision in continuous sign language recognition, evident from its superior average probability metrics. Its proficiency in handling temporal dependencies makes it ideal for the nuanced task of interpreting the fluid nature of sign language.

This theoretical superiority is corroborated by practical tests, where the LSTM model consistently showed high prediction accuracy across various sentence examples.

Conducted research on ordinary computers allows to achieve fairly high accuracy and speed on test data without the use of special graphic tools. Disadvantages of using such methods on ordinary computers include a rather long time of training and data preparation for the neural network.

Future research will be aimed at expanding the set of Kazakh sign language sentences and increasing the accuracy of the proposed methods.

Research will also be conducted for other sign languages.

Acknowledgements

This work is part of the AP19175452 project funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan, aimed at developing automated interpretation models for Kazakh sign language.

References

- [1] N. Amangeldy, M. Milosz, S. Kudubayeva, A. Kassymova, G. Kalakova, L. Zhetkenbay. A Real-Time Dynamic Gesture Variability Recognition Method Based on Convolutional Neural Networks. *Appl. Sci.* (2023) 13, 10799. <https://doi.org/10.3390/app131910799>.
- [2] N. Amangeldy, A. Ukenova, G. Bekmanova, B. Razakhova, M. Milosz, S. Kudubayeva. Continuous Sign Language Recognition and Its Translation into Intonation-Colored Speech. *Sensors* (2023) 23, 6383. <https://doi.org/10.3390/s23146383>.
- [3] A. S. M. Miah, M. A. M. Hasan, J. Shin, Y. Okuyama, Y. Tomioka. Multistage Spatial Attention-Based Neural Network for Hand Gesture Recognition. *Computers*, (2023) 12(1). <https://doi.org/10.3390/computers12010013>.
- [4] D. Satybaldina, G. Kalymova. Deep learning based static hand gesture recognition. *Indonesian Journal of Electrical Engineering and Computer Science*, (2021) 21(1). <https://doi.org/10.11591/ijeecs.v21.i1.pp398-405>.
- [5] H. Brock, I. Farag, K. Nakadai. Recognition of non-manual content in continuous Japanese sign language. *Sensors*, (2020). 20(19). <https://doi.org/10.3390/s20195621>.
- [6] B. Aksoy, O. K. M. Salman, Ö. Ekrem. Detection of Turkish Sign Language Using Deep Learning and Image Processing Methods. *Applied Artificial Intelligence*, (2021) 35(12). <https://doi.org/10.1080/08839514.2021.1982184>.
- [7] D.R. Kothadiya, C.M. Bhatt, A. Rehman, F.S. Alamri, T. Saba. SignExplainer: An Explainable AI-Enabled Framework for Sign Language Recognition with Ensemble Learning. *IEEE Access*, (2023) 11. <https://doi.org/10.1109/ACCESS.2023.3274851>.
- [8] Z. Zhou, V. W. L. Tam, E. Y. Lam. SignBERT: A BERT-Based Deep Learning Framework for Continuous Sign Language Recognition. *IEEE Access*, (2021) 9. <https://doi.org/10.1109/ACCESS.2021.3132668>.
- [9] K. Amrutha, P. Prabu, R. C. Poonia. "LiST: A Lightweight Framework for Continuous Indian Sign Language Translation". *Information*, (2023) 14, 2: 79. <https://doi.org/10.3390/info14020079>.

- [10] M. S. Abdallah, G. H. Samaan, A. R. Wadie, F. Makhmudov, Y. I. Cho. Light-Weight Deep Learning Techniques with Advanced Processing for Real-Time Hand Gesture Recognition. *Sensors*, (2023) 23(1). <https://doi.org/10.3390/s23010002>.
- [11] Q. Xiao, X. Chang, X. Zhang, X. Liu. Multi-Information Spatial-Temporal LSTM Fusion Continuous Sign Language Neural Machine Translation. *IEEE Access*, (2020) 8. <https://doi.org/10.1109/ACCESS.2020.3039539>.
- [12] T. Witchuda, A. Wiranata, S. Maeda, C. Premachandra. Reservoir Computing Model for Human Hand Locomotion Signal Classification. *IEEE Access*, (2023) 11. <https://doi.org/10.1109/ACCESS.2023.3247631>.
- [13] Y. Jiang, L. Song, J. Zhang, Y. Song, M. Yan. Multi-Category Gesture Recognition Modeling Based on sEMG and IMU Signals. *Sensors*, (2022) 22(15). <https://doi.org/10.3390/s22155855>.
- [14] I. Papastratis, K. Dimitropoulos, D. Konstantinidis, P. Daras. Continuous Sign Language Recognition through Cross-Modal Alignment of Video and Text Embeddings in a Joint-Latent Space. *IEEE Access*, (2020) 8. <https://doi.org/10.1109/ACCESS.2020.2993650>.
- [15] O. Koller, J. Forster, H. Ney. "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers", *Comput. Vis. Image Understand.*, vol. 141, pp. 108-125, Dec. 2015. DOI: 10.1016/j.cviu.2015.09.013.
- [16] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney and R. Bowden, "Neural sign language translation", *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 7784-7793, March 2018. DOI: 10.1109/CVPR.2018.00812.
- [17] J. Huang, W. Zhou, Q. Zhang, H. Li and W. Li, "Video-based sign language recognition without temporal segmentation", *Proc. 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*: New Orleans, Louisiana, USA, February 2018. pp. 1-8. ArXiv:1801.10111v1 [cs.CV]30 Jan2018.
- [18] S. Kondratiuk, I. Krak, V. Kuznetsov, A. Kulias. Using the Temporal Data and Three-dimensional Convolutions for Sign Language Alphabet Recognition. *CEUR Workshop Proceedings* 3137, CEUR-WS.org 2022. – P.78-87. DOI: 10.32782/cmis/3137-7.
- [19] I. Papastratis, K. Dimitropoulos, P. Daras. Continuous Sign Language Recognition through a Context-aware Generative Adversarial Network. *Sensors*, (2021) 21(7) 2437. <https://doi.org/10.3390/s21072437>.

Decision Support System Regarding the Possibility of Financing Cross-Border Cooperation Projects

Volodymyr Polishchuk^{1,2}, Martin Kelemen Jr.², Inna Polishchuk¹ and Miroslav Kelemen²

¹ Uzhhorod National University, Narodna Square, 3, Uzhhorod, 88000, Ukraine

² Technical University of Kosice, Rampova 7, Kosice, 04121, Slovak republic

Abstract

Research was conducted on the current task of developing a decision-making support system regarding the possibility of financing cross-border cooperation projects under the conditions of guaranteeing the national security of partner countries during project implementation, as well as designing innovative software as a means of technical support for managers of cross-border project competitions. In this study, an information model for forecasting the level of process control during project implementation was developed for the first time. Based on theoretical-multiple generalization, the factors affecting the implementation of the project are classified, namely: factors of internal and external influences; risk-oriented influencing factors; the effects of human factors; and factors guaranteeing the national security of the partner countries during project implementation. Also, for the first time, a fuzzy model for evaluating cross-border cooperation projects was developed regarding the possibility of their financing, considering the future level of management of processes during implementation. The model has been verified and tested on real and test data. An approbation example of the calculation is given. The conducted research will be a useful decision-making support tool for managers of cross-border project tenders under the conditions of guaranteeing the national security of partner countries during project implementation.

Keywords

Cross-border cooperation projects, National security, Fuzzy set, Expert evaluation, Decision-making

1. Introduction

Cross-border cooperation projects promote mutual understanding and cooperation between bordering countries, thereby contributing to stability. Cross-border cooperation creates new opportunities for economic development, by promoting trade, investment, and infrastructure development in border regions. Such projects often involve the joint construction and modernization of transport, energy, and other types of infrastructure that help improve access to services and markets. In addition, these projects help countries to solve common problems together, such as environmental issues, border security, migration, and other border challenges.

Nevertheless, cross-border cooperation projects require significant investments and joint efforts from partner countries. However, decision-making regarding the financing of such projects is associated with risks regarding the national security of the participating countries. Taking these risks into account in the decision-making process becomes critically important for ensuring the success of cross-border cooperation projects and preventing possible negative consequences for the national security of countries. Therefore, the development of a decision support system that takes these aspects into account is extremely relevant and will contribute to effective management in the implementation of cross-border projects.

Cross-border cooperation is carried out by the procedures of the European Commission. The documents related to this activity cover many aspects of evaluation of all sizes. The effectiveness of the implemented programs is limited in terms of potential analytical gaps [1].

The purpose of the presented research is to develop a decision-making support system regarding the possibility of financing cross-border cooperation projects under the conditions of guaranteeing the

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ volodymyr.polishchuk@uzhnu.edu.ua (V. Polishchuk); martin.kelemen@tuke.sk (M. Kelemen Jr.);

inna.polishchuk@uzhnu.edu.ua (I. Polishchuk); miroslav.kelemen@tuke.sk (M. Kelemen)

ORCID 0000-0003-4586-1333 (V. Polishchuk); 0000-0003-1015-1112 (M. Kelemen Jr.); 0009-0002-6395-4744 (I. Polishchuk); 0000-0001-7459-927X (M. Kelemen)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

national security of partner countries during project implementation, as well as designing innovative software as a means of technical support for managers of cross-border project competitions.

The above argues and confirms the relevance of the conducted research on the development of a decision-making support system regarding the possibility of financing cross-border cooperation projects. The relevance of this study proves the need to develop information models for evaluating cross-border cooperation projects and mathematical models for knowledge processing, using the theory of fuzzy sets and approaches to intellectual analysis of knowledge. In addition, the relevance of this study is reinforced by the European data strategy [2], which provides for the achievement of certain goals by 2030. One of the main goals of this strategy is to create an advanced model society in the European Union that uses data to solve problems in both business and the public sector.

2. Overview of domestic and foreign research studies

Cross-border cooperation reflects one of the forms of support for the development of regions and arises because of the strategic activity of various participants, which involves trust, mutual understanding, and the desire to cooperate. Durand and Decoville [3] consider cross-border integration in Europe as a complex and multidimensional process that has different effects on regions. Their various models show that there is no single best strategy for cross-border cooperation within the EU. Scott [4] argues for the socio-political significance of borders in Europe and beyond, arguing that the border perspective reflects the boundaries between communities and groups through ideologies, political institutions, attitudes, and agencies. Cross-border cooperation initiatives, institutional cooperation, and political support are also important [5]. Crescenzi and Iammarino [6] emphasize the need to review the current processes of regional development and argue that regional development is accompanied by different concepts, empirical evidence, and political approaches that influence decision-making processes.

Obstacles to the development of cross-border cooperation exist in several dimensions. Internal barriers affect the social goals of cooperation the most, and they can be overcome through the policies of municipalities and entrepreneurs [7]. External barriers are characteristic of peripheral regions, which are distant from national and regional decision-making centers, and lead to many negative consequences. Cross-border cooperation can be an effective tool for overcoming them. For the possibility of evaluating such projects, modern information and analytical systems are needed, which embody decision-making support systems for evaluating projects and supporting the controllability of processes during their implementation.

Methods based on the theory of fuzzy mathematics can provide analytical support for quality decision-making processes at any level of management [8]. For the successful development of regions, it is necessary to develop innovative tools and knowledge systems that will support their sustainability, especially in periods of intensive support from EU funds to ensure information security and in the interests of the national security of the partner countries implementing the project.

New decision support systems are technologies that focus on data and knowledge analysis, including the UN Sustainable Development Goals. Today, such technologies are developing rapidly, and especially great attention is paid to mathematical models of knowledge representation. The construction of such models is based on objective information about the object, but it can also be based on incomplete information, since in the process of creating the model, data obtained from experts are used, which reflect the significant features of the object under study and are formulated in natural language. Therefore, to display knowledge, it is worth using the theory of fuzzy sets [9-10] and modern approaches to the use of intellectual analysis of knowledge in decision support systems [11].

As for the application of the theory of fuzzy mathematics for the tasks of evaluating project cross-border activities, the following can be distinguished. For example, source [12] describes the task of evaluating and selecting projects using the hybrid technique of multi-criteria evaluation in a fuzzy environment, based on financial factors. As a result, projects are ranked from most important to least important. In [13], a study aimed at building an integrated framework for assessing the sustainability of public-private partnership projects using the extended VIKOR method in a fuzzy image environment is considered. In work [14], a fuzzy multi-criteria evaluation model of heterogeneous cross-border cooperation projects was developed. The model is aimed at ensuring the sustainable development of neighboring regions, considering the goals of the announced competition.

All these studies are designed in such a way that the evaluation process takes place now and cannot predict what will happen during the implementation of the project. Thus, to date, there is no comprehensive study that makes it possible to evaluate cross-border cooperation projects, while

predicting the level of process control during project implementation and using the approaches of intellectual analysis of knowledge.

3. Decision support system regarding the possibility of financing cross-border cooperation projects

3.1. Formal problem statement and input data

Consider a project (or a set of projects) of cross-border cooperation P . Such projects are presented as project applications for competitive selection, with the aim of their financing and implementation. The project application is evaluated by the expert (experts) E of the competitive selection. All projects of cross-border cooperation P are divided into three groups G , considering the strategic goal of the programs and specific thematic goals: G_1 – micro-projects; G_2 – regular projects; G_3 – large infrastructure projects [14]. The projects will be evaluated using the information model for evaluating cross-border cooperation projects regarding the possibility of their financing – K_{FO} [14]. Also, project applications will be evaluated using the information model for predicting the level of process control during project implementation – K_{PR} . Processed input data according to information models are calculated using a fuzzy model for evaluating cross-border cooperation projects regarding the possibility of their financing, considering the future level of process control during project implementation – M_{TP} .

Formally, the decision support system regarding the possibility of financing cross-border cooperation projects is proposed to be presented in the form of an operator:

$$X(P, E, G, K_{FO}, K_{PR}, M_{TP}) \rightarrow Y(y_{TP}, TP). \quad (1)$$

Where X is an operator that, based on the input data $P, E, G, K_{FO}, K_{PR}, M_{TP}$ matches the set of output values Y . Two values are obtained at the output: y_{TP} – generalized assessment of the level of the possibility of financing cross-border cooperation projects; TP is the linguistic level of the possibility of financing cross-border cooperation projects, considering the future level of manageability of their implementation processes.

The following management subjects are defined for the presented research. Experts are persons who have functional responsibilities for evaluating cross-border cooperation projects based on project applications submitted within the framework of the competition. System analysts are individuals who configure all project evaluation processes in the decision support system. Decision-making persons (DM) are persons who make further management decisions regarding the selection of cross-border cooperation projects for their financing.

Next, a structural diagram of the decision-making support system regarding the possibility of financing cross-border cooperation projects is presented (Fig. 1).

Fig. 1. shows the structural diagram of the decision support system. The cross-border cooperation project P is submitted for evaluation by expert E . After that, the evaluation takes place using the information model K_{FO} , considering which group G the project belongs to. The expert also evaluates the project using an information model for predicting the level of process control during project implementation – K_{PR} . The obtained data according to information models form the research knowledge base. After that, the obtained data are processed with the help of a three-stage fuzzy model for evaluating cross-border cooperation projects regarding the possibility of their financing, considering the future level of process control during project implementation – M_{TP} . On the one hand, an assessment of the project regarding the possibility of its financing ($m(P)$), is obtained, and on the other hand, the quantitative levels $\varphi_1, \varphi_2, \varphi_3, \varphi_4$, which determine the controllability of the processes during the implementation of the project according to the studied factors.

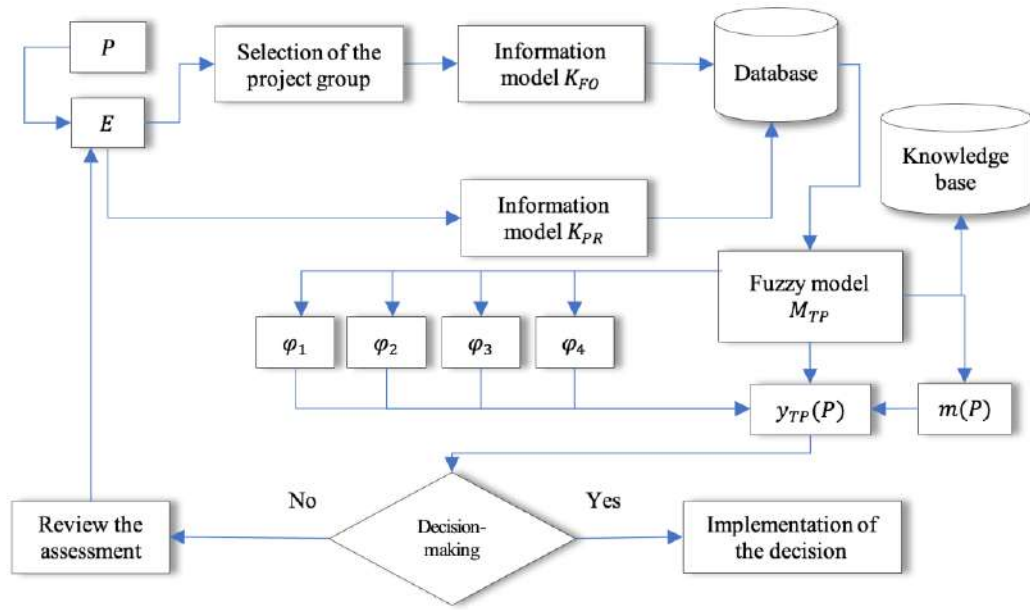


Figure 1: Structural diagram of the decision support system

After that, a generalized quantitative assessment of the possibility of financing the cross-border cooperation project ($y_{TP}(P)$) is obtained. The acquired knowledge forms the research knowledge base. Next, the DM analyzes the received initial evaluations and makes a decision or reviews the evaluation together with experts.

3.2. Information model for evaluating cross-border cooperation projects regarding the possibility of their financing

The process of evaluating cross-border projects is a complex task, due to their implementation by applicants and partners from different countries, which have different goals and address different problems. By analyzing various programs, projects, and challenges of cross-border cooperation, it is possible to generalize the evaluation criteria. Usually, the strategic goal of such programs should be achieved by financing and implementing projects that are determined by the specific thematic goals of the program. The criteria for evaluating projects divided into three groups were summarized: micro-projects, regular projects, and large infrastructure projects.

All criteria for evaluating projects are set out in the form of questions to which the applicant must provide a descriptive answer in his project application. The expert evaluates the project application and determines the appropriate score from the interval [1; 10] to assess the quality of the description. Where 10 points is considered the highest level of quality of the description, this or that criterion. The criteria for evaluating cross-border cooperation projects regarding the possibility of their financing for the K_{FO} information model is described in detail in the work [14] based on official sources [15-16]. Thus, without reducing generality, for this study, we will use the same evaluation indicators presented by the authors in [14].

3.3. Information model for predicting the level of process control during project implementation

To improve the quality of project application evaluation, it is necessary to forecast the possibility of project implementation by partners, considering the conditions affecting this complex process. It is proposed to look at this task from the point of view of determining the level of controllability of the processes of a complex system. Let us know the complex functioning system of the implementation of some cross-border cooperation project P . We know the set of factors F that affect the controllability of the processes of such a complex system. The main goal of the system is the implementation of the project within the established terms. For this applied problem, a theoretical-multiple generalization of the factors that most strongly influence the level of process control during project implementation was

carried out. It is proposed, but not limited, to classify the factors into four main groups, the influence of which is the most significant on the implementation of the project, for example f_1 – factors of internal and external influences; f_2 – risk-oriented influencing factors; f_3 – effects of human factors; f_4 – factors guaranteeing the national security of the partner countries during project implementation. This classification is quite general, but nevertheless, it fully reveals the methodology of the task of assessing controllability in complex systems, and each group of factors complements each other. Of course, many factors are open, when considering an applied problem, assigning factors to one of the groups depends on the competence of system analysts.

Next, the proposed classes of influence factors are considered in more detail.

f_1 – factors of internal and external influences. To predict the level of controllability of processes during the implementation of a cross-border cooperation project, it is first of all necessary to assess this possibility, taking into account internal and external influences.

Internal (external) influences on the implementation of the cross-border cooperation project are events or phenomena in the internal (external) environment that pose a threat to the security of achieving the goal, as they go beyond the permissible estimates of the indicators of the object under study. Therefore, the group of internal and external influences can include goals, tasks, structures, technologies, and people, as well as the external management environment of direct influence and indirect influence.

For example, the following open set of evaluation criteria is proposed for the group of factors of internal and external influences f_1 :

F_1^1 – Partners have a realistic and balanced budget that corresponds to their actual participation.

F_2^1 – The financial plan is realistic and effective.

F_3^1 – Project objectives correspond to the needs analysis and are covered by the main objective of the program or challenge.

F_4^1 – All activities in the project are aimed at achieving its tasks and main goal.

F_5^1 – The project has a real need, determined by the clear needs of the public.

F_6^1 – The level of influence on the implementation of the draft legislation and the political situation.

F_7^1 – The level of influence on the implementation of the project of socio-cultural factors, such as cultural characteristics, language barriers, and other social factors.

f_2 – risk-oriented influencing factors. The level of safety of human activities improves every day due to the improvement of technologies, intelligent analysis of data and knowledge, artificial intelligence management systems, etc. At the same time, the standards for the protection of information security in the management of international cooperation projects have become higher. Such standards are based on the principles of guaranteeing the national security of partner countries in the implementation of projects. However, we face many risks that could potentially jeopardize the success of the cross-border cooperation project if they are not adequately managed. One of the key components of measuring the risk of project implementation is the generation of scenarios for the development of the main risk factors.

To date, general principles and signs of risk classification have not been formed, there are practically no developments on the generalization and formalization of risk classification that can be applied to cross-border cooperation projects. For example, the following open set of evaluation criteria for a risk-oriented group of influence factors is proposed f_2 :

F_1^2 – The project contains a comprehensive and detailed analysis of risks, as well as a plan for their elimination.

F_2^2 – Risks of change in the political environment of the partner countries, conflicts, and instability, the possibility of a change of government, which may affect the implementation of the project.

F_3^2 – Risks related to the possibility of changes in the world economy, tax, or customs policy and will have a significant impact on the implementation of the project.

F_4^2 – Risks of public dissatisfaction during project implementation.

F_5^2 – Risks of negative impact of the project on the environment and/or environmental pollution.

F_6^2 – Risks of conflicts regarding rights and obligations between project partners.

f_3 – effects of human factors. In the process of professional training, professional activity and acquired experience, the individual qualities of management subjects are transformed into a complex system of important professional qualities that ensure the successful formation of professional skills and their reliability, especially in extreme operating conditions. Therefore, there is an interdependence between the personal properties of management subjects, professional activity (professional training),

the level of development of professional qualities and the ability of a person to integrative realization in the process of professional activity. The complex integrative complex of the formation of important professional qualities is not additive according to the individual qualities of the subject of management. The mastery of rigid algorithms of professional activity and the optimal level of development of important professional qualities in the subject of management determines the adequacy of decision-making in extreme conditions.

The success of the implementation of cross-border cooperation projects depends on the professionalism of the partners. Therefore, the systemic vision of formative influences on decision-making by the management subject contains a number of information subsystems, namely: the level of knowledge, skills, abilities, educational and qualification characteristics, psychophysiological, individual-psychological, social-psychological. Thus, the following are proposed as criteria for the influence of human factors on the level of manageability of project implementation processes:

F_1^3 – Insufficient qualifications and experience of project managers.

F_2^3 – The possibility of inefficient project management and insufficient control over the execution of works.

F_3^3 – Insufficient qualification of executors and subjects who will directly participate in project implementation.

F_4^3 – The possibility of conflicts between project participants.

F_5^3 – Ability to interact ineffectively with partners, including communication, collaboration, and conflict management.

f_4 – factors guaranteeing the national security of the partner countries during project implementation. Guaranteeing the national security of partner countries during the implementation of a cross-border cooperation project is a set of measures and strategies aimed at ensuring the protection and stability of each of the participating countries in the process of implementing joint projects. This includes various aspects of security such as political, economic, social, cyber security, border security, and others. The main goal is to prevent any threats that may arise because of the implementation of the project and preserve the interests and security of each of the participating countries. This may include measures to monitor and respond to potential threats, joint security arrangements, information sharing, joint training, and the development of joint security strategies. For example, the following open set of evaluation criteria for factors guaranteeing the national security of partner countries is proposed f_4 :

F_1^4 – Resistance to geopolitical and international risks is the project's ability to withstand geopolitical tensions and international challenges that may arise during its implementation.

F_2^4 – Effective use of resources is an analysis of the effectiveness of the use of financial, technical, and human resources of the project to maximize effectiveness and minimize risks.

F_3^4 – Protection against cyber threats and cyber-attacks is an assessment of cyber security measures and the protection of information systems of the project against potential cyber threats and cyber-attacks.

F_4^4 – Availability of mechanisms that ensure transparency and openness in the project implementation process, including access to information for interested parties and the public.

F_5^4 – Risk and conflict management is an assessment of risk and conflict management strategies to prevent and minimize possible threats to national security.

F_6^4 – Raising Education and Awareness is an assessment of education and awareness of national security issues among participating partner countries.

All criteria of influencing factors are expertly evaluated using one of the terms, the next proposed term-set of linguistic variables $L = \{L; BA; A; AA; H\}$. Linguistic variables characterize the predicted level of the indicator during project implementation, where: L is "low"; BA is "below average"; A is "average"; AA is "above average"; H is "high". Also, experts put a quantitative value from the interval $[1; 100]$, which characterizes the expert's confidence in his assessment – $\mu(F_e^g)$, $g = \overline{1,4}$, $e = \overline{1, m_g}$, m_g – the number of criteria in group g .

The set of criteria does not cover all aspects of cross-border cooperation, which can take place within the framework of various programs and initiatives. Therefore, the set of criteria is flexible, the formal presentation of a fuzzy evaluation model does not depend on their number, and depending on the specific program and its purpose, the DM may add other indicators for evaluation.

3.4. Fuzzy model for evaluating cross-border cooperation projects regarding the possibility of their financing, considering the future level of process control during project implementation

Next, a three-stage mathematical fuzzy model for evaluating cross-border cooperation projects is presented.

The first stage is finding an aggregated assessment of cross-border cooperation projects regarding the possibility of their financing.

Without reducing the generality, in the future we will consider one project of cross-border cooperation P , which belongs to one of the groups G . Such a project underwent an expert evaluation according to the information model – K_{FO} , and received points for each evaluation criterion: $O_1, O_2, \dots, O_m$; $O_i \in [1; 10]$, $i = \overline{1, m}$, де m -where m is the number of criteria for the corresponding evaluation group G .

So, as for each question, experts set a score from the interval $[1; 10]$ regarding the level of quality of the project application, then first you need to normalize the evaluations according to the criteria. Normalization is proposed using intellectual analysis of knowledge and membership functions. The choice of membership functions rests with the system analyst. Here it is suggested to use membership functions of the "large value" type. This type is given by S-like membership functions, for example, a quadratic S-spline is given by the following analytical formula:

$$\mu(O_i) = \begin{cases} 0, & O_i \leq 1; \\ 2 \frac{(O_i-1)^2}{81}, & 1 < O_i \leq 5; \\ 1 - 2 \frac{(10-O_i)^2}{81}, & 6 < O_i < 10, \\ 1, & O_i \geq 10. \end{cases}, i = \overline{1, m}. \quad (2)$$

Thus, a transition was made from quantitative point estimates to normalized and compared ones, $\mu(O_i) \in [0; 1]$.

For each criterion $\{K_1, K_2, \dots, K_m\}$, the DM sets the weighting coefficients $\{v_1, v_2, \dots, v_m\}$ from some interval $[1; 10]$. Next, normalized weighting factors for each criterion are determined:

$$w_i = \frac{v_i}{\sum_{i=1}^m v_i}, i = \overline{1, m}; w_i \in [0; 1]. \quad (3)$$

It is noted that if there is no need to enter weighting coefficients, the weights of the criteria are considered balanced.

In the next step, the evaluation of cross-border cooperation projects is aggregated regarding the possibility of their financing using a convolutional approach. For this, the DM can choose one of the convolutions, considering its considerations [14]:

$$m_1(P) = \frac{1}{\sum_{i=1}^m \frac{w_i}{\mu(O_i)}}. \quad (4)$$

$$m_2(P) = \prod_{i=1}^m (\mu(O_i))^{w_i}. \quad (5)$$

$$m_3(P) = \sum_{i=1}^m w_i \cdot \mu(O_i). \quad (6)$$

$$m_4(P) = \sqrt{\sum_{i=1}^m w_i \cdot (\mu(O_i))^2}. \quad (7)$$

$i = \overline{1, m}$. Thus, aggregated assessments of cross-border cooperation projects regarding the possibility of their financing are obtained, which completes the first stage of solving the task.

In addition to the convolutional approach, other more traditional methods can also be used, such as the method of weighted summation of ranking; the method of analysis of hierarchies; the method of direct comparison, and others. It is noted that convolution methods have advantages over traditional methods in cases where complex alternatives need to be evaluated using multiple criteria and trade-offs

between different objectives must be considered. This can be especially important in large or complex projects where large volumes of data need to be processed and complex interdependencies between different criteria need to be dealt with. Convolution methods allow considering trade-offs between different criteria, which can be important when making decisions in complex situations. They allow you to find the most optimal solutions that consider certain compromises and prevent extreme options.

The second stage is the finding of aggregated predicted estimates of factors influencing the level of manageability of project implementation processes.

Each criterion of influence factors is evaluated by an expert using one of the L terms and the expert's confidence in issuing the term estimate – $\mu(F_e^g)$. Therefore, the input data of the cross-border cooperation project P is a set of estimates $\lambda_e^g = \{L_e^g; \mu(F_e^g)\}$, $g = \overline{1,4}$, $e = \overline{1, m_g}$.

Fuzzification of fuzzy linguistic reasoning of experts is carried out, which will make it possible to compare the obtained data regarding the impact on the predicted level of controllability of processes during project implementation. For each linguistic variable, a value from the interval $[0;1]$ is determined: L – $[a_1; a_2]$, BA – $[a_3; a_4]$, A – $[a_5; a_6]$, AA – $[a_7; a_8]$, H – $[a_9; a_{10}]$. For example: L – $[0; 0,2]$, BA – $[0,2; 0,4]$, A – $[0,4; 0,6]$, AA – $[0,6; 0,8]$, H – $[0,8; 1]$. After that, based on the linguistic variable and the confidence of the expert's reasoning regarding their assignment, one normalized score is calculated:

$$\vartheta_e^g = a_t + \frac{1}{100} \cdot \mu(F_e^g) \cdot (a_{t+1} - a_t). \quad (8)$$

Where a_t is the interval value for the linguistic variable L_e^g , $t = \overline{1,9}$. ϑ_e^g is the normalized numerical value of the criterion for the influence factor on the predicted level of process control during project implementation, which is adjusted for the confidence of the expert's reasoning $g = \overline{1,4}$, $e = \overline{1, m_g}$.

Next, there is one aggregated predicted assessment within the influence factors f_1, f_2, f_3, f_4 . In this case, for each criterion within the influence factors g there are weighting factors $g \{ \alpha_{g1}, \alpha_{g2}, \dots, \alpha_{gm_g} \}$ from the interval $[1; 10]$. Similarly, within the influence factors g , normalized weighting factors for each criterion are determined:

$$\beta_{ge} = \frac{\alpha_{ge}}{\sum_{e=1}^{m_g} \alpha_{ge}}, g = \overline{1,4}, e = \overline{1, m_g}, \beta_{ge} \in [0; 1]. \quad (9)$$

After that, one predicted aggregated estimate is calculated within the influence factors f_1, f_2, f_3, f_4 of the level of process control during project implementation:

$$\varphi_g = \sum_{e=1}^{m_g} \beta_{ge} \cdot \vartheta_e^g, g = \overline{1,4}. \quad (10)$$

The obtained estimates $\varphi_g \in [0; 1]$ characterize the quantitative level that determines the predicted controllability of the processes during the implementation of the project according to the studied influence factors. The larger the value of $\varphi_g \in [0; 1]$, the better the predicted level of process control during project implementation.

So, at the end of the second stage, aggregated predicted estimates $\varphi_1, \varphi_2, \varphi_3, \varphi_4$ of factors affecting the level of controllability of project implementation processes were obtained in relation to the opinions of expert E .

The third stage is to find generalized quantitative estimates and linguistic levels of the possibility of financing cross-border cooperation projects, considering factors influencing the level of manageability of project implementation processes.

For an adequate interpretation of the dependence of assessments of cross-border cooperation projects on the possibility of their financing and considering the quantitative forecast level, which determines the controllability of processes during the implementation of the project on the studied factors, the following function of belonging is built:

$$\Delta_g = \begin{cases} 0, & m(P) < 0; \\ (m(P))^{\varphi_g}, & 0 \leq m(P) \leq 1; \\ 1, & m(P) > 1. \end{cases} g = \overline{1,4}. \quad (11)$$

Thus, estimates $\Delta_1, \Delta_2, \Delta_3, \Delta_4 \in [0; 1]$ are obtained, which characterize the level of the possibility of financing the project of cross-border cooperation in terms of factors affecting the control of processes during the implementation of this project.

Let DM set the weighting coefficients $\{p_1, p_2, p_3, p_4\}$ for each influence factor f_1, f_2, f_3, f_4 from the interval $[1; 10]$. Next, to obtain a generalized quantitative assessment of the possibility of financing a cross-border cooperation project, a weighted average convolution is used, while the weighting coefficients are normalized:

$$y_{TP}(P) = \frac{1}{p_1 + p_2 + p_3 + p_4} (p_1 \cdot \Delta_1 + p_2 \cdot \Delta_2 + p_3 \cdot \Delta_3 + p_4 \cdot \Delta_4). \quad (12)$$

In the end, TP is derived - the linguistic level of the possibility of financing cross-border cooperation projects, considering the future level of management of their implementation processes. For this, the obtained estimate $y_{TP}(P)$ is compared to one variable of the term sets $TP = \{tp_1, tp_2, \dots, tp_5\}$ with the following content: $y_{TP}(P) \in (0,89; 1] - tp_1$ = “high level regarding the possibility of financing a cross-border cooperation project”; $y_{TP}(P) \in (0,77; 0,89] - tp_2$ = “level regarding the possibility of financing a cross-border cooperation project – higher average”; $y_{TP}(P) \in (0,65; 0,77] - tp_3$ = “average level regarding the possibility of financing a cross-border cooperation project”; $y_{TP}(P) \in (0,54; 0,65] - tp_4$ = “low level regarding the possibility of financing a cross-border cooperation project”; $y_{TP}(P) \in [0; 0,54] - tp_5$ = “very low level regarding the possibility of financing a cross-border cooperation project”.

Demarcations between levels rely on the system analyst, using own experience and real data from cross-border cooperation projects.

It is noted that this study used the evaluations of one expert to evaluate the project. It is known that several experts $E = \{e_1; e_2; \dots; e_q\}$ work in tender commissions. In this case, the evaluation procedure is repeated q times. At the output of the project P , q estimates are obtained: $(y_{TP}(P))_1, (y_{TP}(P))_2, \dots, (y_{TP}(P))_q$. Then it is necessary to derive one aggregated estimate for the cross-border cooperation project, considering the opinions of all experts. To do this, you can use the intelligent analysis of knowledge using multidimensional membership functions or the convolutional approach described above.

4. Results

The decision-making support system regarding the possibility of financing cross-border cooperation projects was tested and verified on real and test data. Real data were obtained from the organization that implements cross-border cooperation projects, namely: the Agency for Regional Development and Cross-Border Cooperation of Transcarpathia (Ukraine). For the possibility of reproducing the research by other scientists, an example of the evaluation of a regular project of cross-border cooperation is shown below: P - Visual control of the functioning of checkpoints (PBU1/0240) [16]. The main goal of the project is the use of unmanned aerial vehicles intended for monitoring violations of the airspace of the state border, water bodies, and forests, which can take aerial photography and conduct search activities.

Next, a three-stage evaluation using a mathematical fuzzy model of evaluation of cross-border cooperation projects is presented.

The first stage.

Suppose that the cross-border cooperation project P underwent expert evaluation according to the information model – K_{FO} , and received points for each evaluation criterion. After obtaining point estimates according to the criteria (O), it is necessary to normalize them ($\mu(O)$), using formula (2). Also, DM sets the weighting coefficients (v) for each evaluation criterion. Next, the normalized weighting coefficients (w) for each criterion are determined according to formula (3). All obtained and calculated data are presented in Table 1.

Table 1

Input and normalized evaluations according to project evaluation criteria P

Name of the group	Criterion	v	w	O	$\mu(O)$
Financial and operational capacity	K_1^2	10	0.078	8	0.9012
	K_2^2	9	0.07	8	0.9012
	K_3^2	8	0.062	10	1
	K_4^2	9	0.07	9	0.9753
Cross-border approach	K_5^2	9	0.07	7	0.7778
	K_6^2	10	0.078	8	0.9012
	K_7^2	9	0.07	7	0.7778
	K_8^2	9	0.07	9	0.9753
Sustainability	K_9^2	8	0.062	9	0.9753

Action and communication plan	K_{10}^2	8	0.062	5	0.3951
	K_{11}^2	9	0.07	7	0.7778
	K_{12}^2	7	0.054	8	0.9012
	K_{13}^2	8	0.062	10	1
	K_{14}^2	9	0.07	7	0.7778
	K_{15}^2	7	0.054	9	0.9753

In the next step, the evaluation of cross-border cooperation projects is aggregated regarding the possibility of their financing. Let the DM choose the average convolution according to formula (6): $m_3(P) = 0.868$.

The second stage.

In the second stage, an expert assessment takes place according to the information model for predicting the level of process control during project implementation – K_{PR} . The received input data of the evaluated project are shown in Table 2. Further, based on the linguistic variable (L) and the confidence of the expert's judgments ($\mu(F)$) regarding their assignment, one normalized evaluation (ϑ) is calculated according to formula (8).

Let DM determine the weighting coefficients α from the interval [1; 10] for each criterion within the influence factors g . Similarly, within the influence factors g , normalized weight coefficients β are determined for each criterion according to the formula (9). All calculation results are given in Table 2.

After that, one predicted aggregated estimate is calculated within the influence factors of the level of process control during project implementation: $\varphi_1=0.73$; $\varphi_2=0.765$; $\varphi_3=0.876$; $\varphi_4=0.712$.

The third stage.

At the final stage, generalized quantitative estimates and linguistic levels of the possibility of financing cross-border cooperation projects are found, considering the factors influencing the level of manageability of project implementation processes. First, the dependencies of estimates of cross-border cooperation projects are calculated according to formula (11) regarding the possibility of their financing and taking into account the quantitative forecasted level, which determines the controllability of the processes during the implementation of the project according to the studied factors: $\Delta_1=0.902$; $\Delta_2=0.898$; $\Delta_3=0.884$; $\Delta_4=0.904$.

Let DM set the weighting coefficients {9,9,8,10} for each influence factor from the interval [1; 10]. Next, to obtain a generalized quantitative assessment of the possibility of financing a cross-border cooperation project, a weighted average convolution is used, according to formula (12): $y_{TP}(P) = \frac{1}{9+9+8+10} (9 \cdot 0.902 + 9 \cdot 0.898 + 8 \cdot 0.884 + 10 \cdot 0.904) = 0.8$

In conclusion, TP is derived - the linguistic level of the possibility of financing cross-border cooperation projects, considering the future level of manageability of their implementation processes: $y_{TP}(P) \in (0.89; 1]$ – tp_1 = “high level regarding the possibility of financing a cross-border cooperation project”.

For the practical implementation of the decision-making support system regarding the possibility of financing cross-border cooperation projects, software in the C# programming language was developed. The main window of the software is shown in Fig. 2. Such software will be a useful tool for tender managers of cross-border projects, to increase the validity of decision support.

Table 2

Input and normalized estimates according to the information model – K_{PR}

A group of factors	Criterion	α	β	L	$\mu(F)$	ϑ
f_1	F_1^1	9	0.143	AA	80	0.76
	F_2^1	8	0.127	H	90	0.98
	F_3^1	10	0.159	H	70	0.94
	F_4^1	9	0.143	BA	90	0.38
	F_5^1	10	0.159	A	80	0.56
	F_6^1	9	0.143	AA	70	0.74
	F_7^1	8	0.127	AA	90	0.78
f_2	F_1^2	10	0.182	A	70	0.54
	F_2^2	9	0.164	H	80	0.96

f_3	F_3^2	10	0.182	H	90	0.98
	F_4^2	8	0.145	BA	60	0.32
	F_5^2	9	0.164	AA	80	0.76
	F_6^2	9	0.164	H	90	0.98
	F_1^3	9	0.2	H	90	0.98
	F_2^3	9	0.2	H	70	0.94
	F_3^3	10	0.222	AA	80	0.76
	F_4^3	10	0.222	AA	90	0.78
	F_5^3	7	0.156	H	80	0.96
	F_1^4	10	0.179	H	70	0.94
f_4	F_2^4	9	0.161	H	60	0.92
	F_3^4	10	0.179	H	70	0.94
	F_4^4	10	0.179	A	80	0.56
	F_5^4	9	0.161	AA	90	0.78
	F_6^4	8	0.143	H	90	0.98

Figure 2: The main window of the decision support system

To check the adequacy and determine the effectiveness of the proposed decision-making support system regarding the possibility of financing cross-border cooperation projects, the results of project evaluation were compared with test regulatory approaches, namely: the VIKOR method, the TOPSIS method, and the ELECTRE method [13]. For this, five projects of cross-border cooperation were evaluated with the involvement of 9 experts [14]. The experts gave their opinions on whether the selected project, based on normative methods and the proposed method in the work, would suit them. At the same time, each expert gave their priorities and importance for evaluation criteria based on multi-criteria evaluation models. As a result, a different combination of alternative solutions was obtained for various experts. Also, to improve the quality of the evaluation, each expert put a quantitative number of "reliability" of his reasoning, from the interval [0; 1]: 0 if the chosen assessment approach did not suit the expert at all, 1 – on the contrary, it suited the expert as much as possible. "the chosen method suits the expert" was chosen as the comparative criterion. Despite the slight differences in the evaluation by the normative methods and the proposed approach, the results were identical. So, this testifies to the adequacy of the proposed approach. At the same time, the average accuracy regarding the satisfaction of experts in choosing alternative projects was higher: by 8.86% compared to the VIKOR method; by 10.12% compared to the TOPSIS method; and by 4.34% compared to the ELECTRE method. Then it can be concluded that the satisfaction of experts in the selection of projects according to the proposed approach is higher by 8% compared to the average arithmetic result of the selected normative methods of multi-criteria evaluation.

5. Discussion

The work developed a decision support system regarding the possibility of financing cross-border cooperation projects under the conditions of guaranteeing the national security of the partner countries during the implementation of the project. The research used an adequate apparatus of fuzzy sets, intellectual analysis of experts' opinions, and the principles of a system approach, which together make it possible to increase the objectivity of expert evaluation and management decision-making.

The peculiarity of the study is that it allows to derivation of a generalized quantitative assessment and the linguistic level of the possibility of financing cross-border cooperation projects, considering the future level of manageability of project implementation processes. Based on the obtained initial data, knowledge about the possibility of financing cross-border cooperation projects is determined, which is aimed at increasing the degree of validity of decision-making regarding the selection of projects for the possibility of their financing. Such knowledge is aimed at protecting information security during the management of international cooperation projects based on guaranteeing the national security of partner countries implementing projects. The knowledge obtained in this study will be useful for various management subjects in the ecosystem of financing cross-border cooperation projects for the purpose of making management decisions.

The received expert data is processed with the help of a fuzzy model for evaluating cross-border cooperation projects regarding the possibility of their financing, considering the future level of controllability of processes during implementation. The openness of the set of criteria for evaluating projects and factors that affect the level of process management during project implementation allows adapting the decision-making support system to highly specialized projects.

The advantages of the decision support system are as follows: it uses the input linguistic variables and the confidence of the expert's reasoning regarding their assignment; increases the objectivity of assessment and the reliability of expert assessments; the level of process control during project implementation is predicted; the factors of guaranteeing the national security of the partner countries are considered during the implementation of the project. In addition, to check the adequacy and determine the effectiveness of the proposed decision support system, the results of the project evaluation were compared with test regulatory approaches. It was found that the satisfaction of experts in the selection of projects according to the proposed approach was higher by 8% compared to the average arithmetic result of the selected normative methods of multi-criteria evaluation.

For this study, the method of traditional fuzzy inference can also be applied. The main advantage of such a method is its ability to effectively manage vagueness and uncertainty in data, which allows you to make informed decisions under conditions of uncertainty. However, it is very important to correctly define the set of rules and input parameters to ensure adequate decision-making results. Such a set of rules should be built separately for each group of projects: micro-projects, regular projects, and large infrastructure projects. In addition, it is necessary to have enough data on successfully implemented projects for adequate construction of the knowledge base. Instead, the proposed three-stage fuzzy model does not require such efforts and can be easily implemented in the practical work of cross-border project tender managers.

A limitation of our study was the selection of criteria and factors whose impact on project implementation was considered the most significant, as well as the use of various types of membership functions and data fuzzification approaches, including convolutions to obtain overall estimates. Another limitation is the geographical coverage of the study and the selection of cross-border cooperation projects. That is when receiving more projects from different countries, it would be possible to better adjust the fuzzy model to obtain more accurate results. Instead, the application of the developed decision support system does not impose any restrictions on the country of origin of the cross-border project.

Such limitations may lead to ambiguity in the results, but at the same time, the effectiveness of the developed decision support system has been proven and confirmed by the reasonable use of mathematical theory and verification on both real and test data.

6. Conclusions

During the conducted research, for the first time, an information model was developed for forecasting the level of controllability of processes during project implementation. Based on theoretical-multiple generalization, the factors affecting the implementation of the project are classified, namely: factors of internal and external influences; risk-oriented influencing factors; the effects of human factors; and

factors guaranteeing the national security of the partner countries during project implementation. Based on these factors, an open set of evaluation criteria of a total number of 24 was proposed. Also, for the first time, a fuzzy model of evaluation of cross-border cooperation projects was developed regarding the possibility of their financing, considering the future level of controllability of processes during implementation. The model consists of three stages: finding an aggregated assessment of cross-border cooperation projects regarding the possibility of their financing; finding generalized quantitative estimates and linguistic levels of the possibility of financing cross-border cooperation projects and considering factors influencing the level of manageability of project implementation processes. An approbation example of the calculation is given. For the practical implementation of the decision-making support system regarding the possibility of financing cross-border cooperation projects, software was developed, which will be a useful tool for managers of cross-border project tenders, to increase the validity of decision-making support.

The obtained results demonstrate the scientific and applied value of the conducted research. Further research of the problem is seen in the development of other mathematical models and software support for the evaluation of cross-border cooperation projects under the conditions of guaranteeing the national security of the partner countries during the implementation of the project.

Acknowledgements

It was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I03-03-V01-00059.

The scientific research and preparation of the article took place within the framework of the scientific project of young scientists "Protection of information security in the management of international cooperation projects based on guaranteeing the national security of Ukraine" (DB-921M) with financial support of the Ministry of Education and Science of Ukraine. This publication is also the result of the project „New possibilities and approaches of optimization within logistical processes“, supported by Operational Program Integrated Infrastructure (ITMS: 313011T567), and the Slovak Research and Development Agency project PP-COVID-20-0002.

References

- [1] R. Tiganasu, T.D. Jijie, K. Kourtit, Effectiveness and impact of cross-border cooperation programmes in the perception of beneficiaries, Investigation of 2007–2013 Romania–Ukraine–Moldova programmes Regional Science Policy & Practice 12 (5) (2020) 867–891. doi:10.1111/rsp3.12342
- [2] A European strategy for data, 2020. URL: https://ec.europa.eu/info/sites/default/files/communication-european-strategy-data-19feb2020_en.pdf
- [3] Durand, A. Decoville, A multidimensional measurement of the integration between European border regions, J. Eur. Integrat. 42 (2) (2020) 163–178. doi:10.1080/07036337.2019.1657857
- [4] J. Scott, Bordering, border politics and cross-border cooperation in Europe, in: F. Celata, R. Coletti (Eds.), Neighbourhood Policy and the Construction of the European External Borders, 115, Springer, Cham 2015. doi:10.1007/978-3-319-18452-4_2
- [5] J.L. Wong Villanueva, T. Kidokoro, F. Seta, Cross-border integration, cooperation and governance: a systems approach for evaluating “good” governance in cross-border regions, J. Borderl. Stud. 37 (5) (2022) 1047–1070. doi:10.1080/08865655.2020.1855227
- [6] R. Crescenzi, S. Iammarino, Global investments and regional development trajectories: the missing links, Transitions in Regional Economic Development (2018) 171–203. doi:10.4324/9781315143736-9
- [7] J. Kurowska-Pysz, R.A. Castanho, J.M. Naranjo Gómez, Cross-border cooperation: the barriers analysis and the recommendations, Polish Journal of Management Studies 17 (2) (2018) 134–147. doi:10.17512/pjms.2018.17.2.12
- [8] G.D.B. Sir, E. Çalışkan, Assessment of development regions for financial support allocation with fuzzy decision making: a case of Turkey, Soc. Econ. Plann. Sci. 66 (2019) 161–169. doi:10.1016/j.seps.2019.02.005

- [9] A.A. Oliinyk, S.A. Subbotin, The decision tree construction based on a stochastic search for the neuro-fuzzy network synthesis, *Optical Memory and Neural Networks (Information Optics)* 24(1) (2015) 18-27. doi:10.1007/s10559-012-9405-z.
- [10] S. Sobhi, S. Dick, An investigation of complex fuzzy sets for large-scale learning, *Fuzzy Sets and Systems* 471 (2023) 108660. doi:10.1016/j.fss.2023.108660.
- [11] S. A. Subbotin, Data clustering based on inductive learning of neuro-fuzzy network with distance hashing, *Radio Electronics, Computer Science, Control* (4) 71 (2022) 71-85. doi:10.15588/1607-3274-2022-4-6
- [12] S.R. Vijayakumar, P. Suresh, K. Sasikumar, K. Pasupathi, T. Yuvaraj, D. Velmurugan, Evaluation and selection of projects using hybrid MCDM technique under fuzzy environment based on financial factors, *Materials Today: Proceedings* 60(3) (2022) 1347-1352. doi:10.1016/j.matpr.2021.10.138.
- [13] Chao Tian, Juan-juan Peng, Shuai Zhang, Jian-qiang Wang, Mark Goh, A sustainability evaluation framework for WET-PPP projects based on a picture fuzzy similarity-based VIKOR method, *Journal of Cleaner Production* 289 (2021) 125130. doi:10.1016/j.jclepro.2020.125130.
- [14] M. Skare, B. Gavurova, V. Polishchuk, Fuzzy multicriteria evaluation model of cross-border cooperation projects under resource curse conditions, *Resources Policy* 85(B) (2023) 103871. doi:10.1016/j.resourpol.2023.103871.
- [15] The Hungary-Slovakia-Romania-Ukraine ENI CBC programme 2014, 2020. URL: <https://huskroua-cbc.eu/about/programme-description>
- [16] Cross-border cooperation projects Poland-Belarus-Ukraine 2014-2020, 2021. URL: https://pbu2020.eu/files/librarynews/file/c17c2e26-94bc-4940-99e2-246885e1c8c0/Regular_Projects_UA_05_2021.pdf

CBR Method for Decision-making Support in Operation Efficiency Ensuring of Complex Technical Systems

Vladimir Vychuzhanin¹, Alexey Vychuzhanin¹ and Nickolay Rudnichenko¹

¹ Odessa Polytechnic National University, Shevchenko Avenue 1, Odessa, 65001, Ukraine

Abstract

The sequence of decision-making, using the precedent-based reasoning method of the CBR cycle, which takes into account the operations of processing and structuring data based on precedents within the framework of the functioning of the developed application software system, ensures high performance, adaptability, work with incomplete information, versatility and learning ability when supporting decision-making to ensure efficiency operation of complex technical systems. The proposed information intelligent system allows, taking into account partial and complete failures of the operability of subsystems, components, elements and mutual connections of critical use complex technical systems to increase the efficiency of their operation as a result of system equipment pre-failure technical condition identifying signs at an early development stage.

Keywords

Information intelligent system, risk assessment, forecasting, technical condition, complex technical system, CBR cycle structure, similarity degree algorithm, complex systems pre-failure maintenance, performance

1. Introduction

Currently, there is an increase in the use of decision-making technologies for the effective operation of complex technical systems (CTS) [1, 2]. CTS are hierarchical structures with non-trivial internal connections, multiple functional subsystems, components, elements and their mutual connections, which are in various states of failure. The operation of CTS is associated with uncertainties [3]. One of the main causes of man-made accidents associated with the operation of CTS used in transport, aviation, energy, etc. remains the failure of their subsystems, components, and elements [4]. In this regard, such CTS are classified as systems of critical use. As the safety requirements for expensive CTS increase, the requirements for their efficiency also increase, depending on: the ability to operate the systems in the event of partial failures; time and resource during operation. In order to identify signs of pre-failure technical condition (TC) of CTS equipment of critical use at an early development stage it is necessary to use intelligent information technologies to support decision-making [5, 6]. The core of such information technologies are conceptual stochastic models and methods for diagnosing vehicles, effective intelligent information systems (IIS) for assessing, forecasting and controlling vehicles. The use of reasoning method based on CBR (Case-Based Reasoning) precedents simplifies decision-making for various types of uncertainties in the initial CTS data and expert knowledge, as well as in cases of emergency situations [7].

2. Description of problem

In order to increase the operational efficiency of critical ship CTS, solving the necessary decision-making problem is complicated due to the need to take into account a significant number of various factors. First of all, this is a large amount of information required to make an appropriate decision. When assessing, predicting and controlling a vehicle, it is necessary to take into account the mutual influence and different degrees of subsystems failure, components,

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ icst_nuop@ukr.net (V.Vychuzhanin); vychuzhanin.o.v@op.edu.ua (A. Vychuzhanin); nickolay.rud@gmail.com (N. Rudnichenko)

📄 0000-0002-6302-1832 (V.Vychuzhanin); 0000-0001-8779-2503 (A. Vychuzhanin); 0000-0002-7343-8076 (N. Rudnichenko)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

elements and their parameters on each other.

The operation of shipboard CTS is subject to uncertainties that are difficult or impossible to fully describe, understand or predict. The presence of uncertainties affects onto vehicle's assessment, forecasting and management, which is due to the lack of clear data on the consequences of failures in the structural and functional relationships of vehicle's subsystems, components and elements makes problem more relevant.

In accordance with the requirements of the Register of Maritime Shipping, all ships must be equipped with systems based on artificial intelligence technology [4]. To implement such technology, appropriate algorithmic and software tools are required to provide support for decision-making that is adequate to the goal. However, analysis and assessment of the technical characteristics of CTS equipment and the development of control actions in most cases are often carried out by ship operators on the basis of heuristic rules. It is impossible for personnel servicing CTS of critical applications to analyze and evaluate the TC of system equipment. This problem can be solved by using methods used in IIS systems.

The development of IIS systems for assessment, forecasting and control vehicle's subsystems, components, elements in order to ensure ship CTS survivability under adverse impacts and damaging factors is one of the promising areas in ensuring CTS operation safety. Such IIS systems can be implemented both in the form of separate stand-alone solutions and in the form of modules that complement ready-made general-purpose control and decision-making systems with the necessary functionality. They will allow us making quick decisions at the stage of eliminating the consequences of adverse impacts and damaging factors, and ensure the efficiency ship CTS operation due to the ability to evaluate, predict and control their vehicles [8, 9].

Well-known models, methods, algorithms and software packages reflect, to one degree or another, the facts of successful solution of problems of using IIS systems for assessments, forecasting and management of CTS vehicles, having advantages and disadvantages. The main complexities of modern technologies and methods which are useful for IIS systems lie in solving complex formalized problems arising from insufficient efficiency: solving problems of training, customization and adaptation to the problem area; processing fuzzy and multi-criteria source information sets; data interpretation and expert knowledge (experience) accumulation. These limitations in IIS systems are addressed by using CBR [6, 7].

Methods for finding solutions in IIS systems based on precedents are an approach based on the use of analogies with previously solved problems to find and adapt solutions to new situations. Similar methods include the steps that form a CBR cycle:

- capturing precedents from the precedent library (PL);
- indexing (organizing precedents to search for similar cases);
- search for the most suitable precedents for a new task.
- adaptation (modification of the found precedent to suit the current task);
- evaluation and implementation (checking the adapted solution for its suitability and, if necessary, implementation).

CBR advantages: adaptability; working with incomplete information; versatility; learning ability. Use cases can be represented in ways including textual description, diagrams, tables, prototypes and use cases, modeling via UML. Each of these methods can be effective depending on the context and goals of the project.

From the analysis it follows that the task of ensuring the efficient operation of ship CTS is still far from a final solution. The problem of creating IIS systems used to solve complex, difficult-to-formalize problems for assessing and predicting the TC of complex ship systems of critical use remains relevant [10, 11].

Thus, taking into account the specifics and existing problems in ensuring the efficiency of operation of ship CTS, an information approach to the procedures for creating models, methods, algorithms, and software packages for IIS systems that ensure the functioning of complex systems for TC seems promising. The development and research of an IIS system using CBR in order to increase the efficiency learning mechanisms implementation and problem specifics adaptation to the current environment for applied specific applications, as well as to increase decision makers (DM) efficiency based on the results of assessing and predicting ship complex systems TC is important and relevant.

3. Conceptualization and formalization of data and knowledge

The development and research of the IIS system, which uses the method of reasoning based on precedents when assessing failures in CTS, was carried out taking into account: partial and complete failures subsystems operability, components, elements and their mutual connections; precedent knowledge model based on the vehicle dynamics model from a serviceable state to complete failure. The vehicle dynamics model takes into account cause-and-effect relationships and the hierarchical structure of the vehicle system, consisting of: elements (E); components (C); subsystems (S); CTS.

During operation, subsystems, components, elements of the CTS in emergency scenarios, taking into account Harrington's generalized desirability function, can be in one of the following TC [12]: 0 - 0.2 - the level of risk and consequences are minimal, not affecting the operation of the CTS (RMi) ; 0.2 - 0.37 - the risk level is acceptable and the consequences are insignificant, allowing the CTS to be operated without repair (RA); 0.37 - 0.63 - the level of risk is maximum, the consequences are significant, but allow the operation of the CTS when performing repair work (RMa); 0.63 - 1.0 - the risk level is critical, the consequences are catastrophic, preventing the operation of the CTS (RC). Taking into account [13] for the hierarchical structure of the CTS, the transitions of the TC can be represented in the form of a TC matrix (Fig. 1).

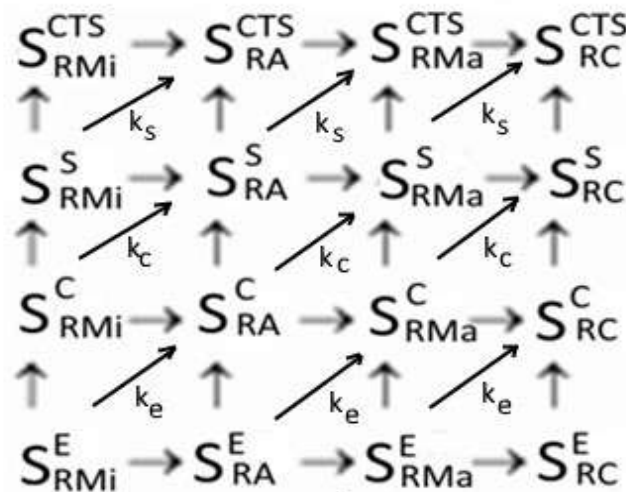


Figure 1: TC CTS matrix

k_e , k_c , k_s – coefficients of weight (significance) of an element, component, subsystem in the corresponding CTS structures.

The description of the problematic situation during the CTS operation consists of the partial or complete functionality loss consequences of subsystems, components, elements and their mutual connections

4. Method of presenting precedents

The method of presenting precedents is implemented as follows. In the proposed CBR cycle (Fig. 2), to support the exchange of knowledge, the initial task generation block receives a set of input diagnostic parameters of the complex system's vehicle and an array of ontologies representing a structured description ship's CTS domain area. As a result, the structure of a new precedent object is generated and content is extracted using the nearest neighbor method based on the results of assessing the degree of similarity (proximity) of the scenario under consideration with the vehicle and taking into account the data in the knowledge base (KB). Based on the implementation of this procedure, a solution object is formed, which can be changed for its target adaptation, taking into account all aspects of the considered scenarios of partial and complete failures CTS subsystems, components, elements using the transformation method [14]. The updated precedent is checked for logical consistency based on the use of

predicative productions using the ontological constructing logical inference Hermit method [15]. The solution obtained as a result of the performed actions is exported as a separate object containing recommendations for decision-makers (DM) and metadata. After this, the precedent is saved in the precedent database, which is an integral KB part (Fig. 2).

The sequence of decision-making (Fig. 3) using the proposed CBR cycle, taking into account the operations of processing and structuring data according to precedents within the framework of the application software system operation is carried out next way

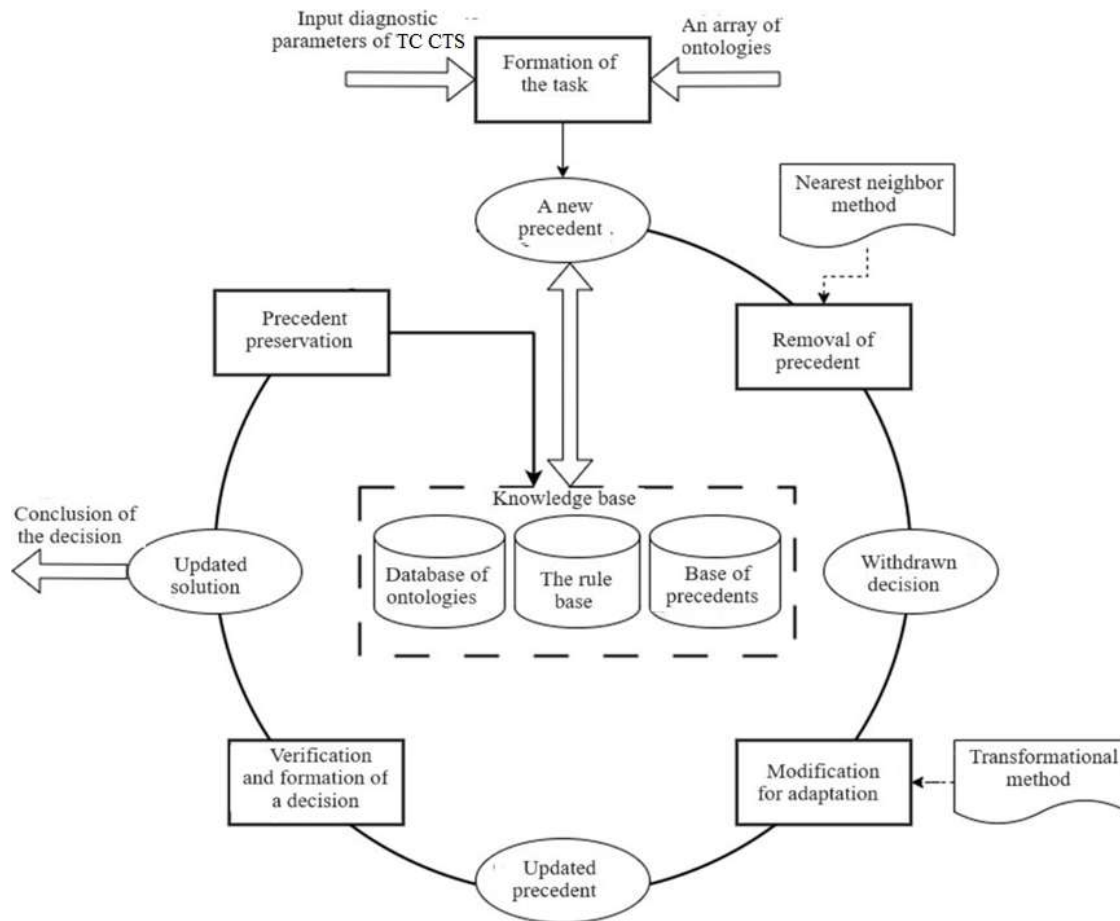


Figure 2: CBR cycle structure

When the software system is launched, the main form of the user interface is initialized, which provides the ability to import input data for generating a task. Then control parameters and options are set to configure the operation of all modules involved in the information processing cycle, in particular the data processing (DPMoD), case extraction (PEMoD) and adaptation (AMoD) modules. Next, a request is made to transfer the generated data arrays to DPMoD, in which the procedures for processing the data are carried out step by step (including checking them for consistency and breaking them into fragments), forming a collection for storing precedents, taking into account the presence of metadata (such as a short text description of the precedent purpose, its identifier, date of formation and some statistical indicators). After this, a request is made to obtain a specific use case in PEMoD, in which metric estimation actions are performed based on the nearest neighbor's method. The result is sent to DPMoD as a collection based on an associative array. DPMoD, after checking and verifying the results, sends the generated collection to AMoD for the purpose of carrying out the adaptation procedure. For this purpose, a transformation method is used based on comparison of a precedent with a rules set that take into account the correspondence logical products. As a result, the updated collection of precedents is sent to DPMoD to form a list of final decisions and validate them. To display the results in the form of text records and graphical representation, the resulting solution in serialized form (in json format) is sent to the main form of the interface.

This is necessary for the subsequent initialization of the process of sending precedent data, storing it in the knowledge base and issuing an information notification to the user about transaction results.

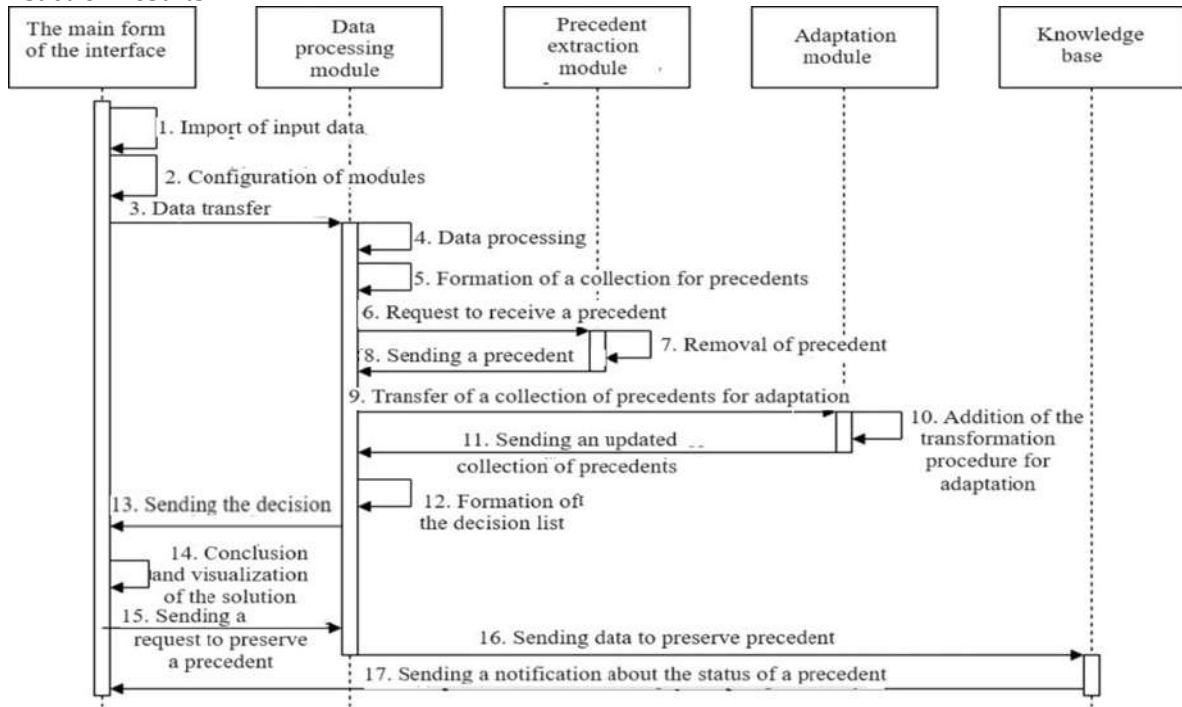


Figure 3: Decision sequence diagram

To represent precedents, a simple parametric representation is sufficient, i.e. presentation of a precedent in the parameters set form and some specific values for solution (decision maker's diagnosis and recommendations)

$$CASE = (R, P, D, W_{\vartheta_{S(C,E)n(m)}}^f, W_{\omega_{IS(C)a(z)}}^f, RE, SR, DR), \quad (1)$$

where R, P, D , - parameters describing the use case;

$R \left\{ R_{S(C,E)n(m)}, R_{IS(C)a(z)} \right\}$ - set of assessments of subsystem's failures risk components, elements and their interrelations CTS and DM recommendations;

$P \left\{ P_{S(C,E)n(m)}, P_{IS(C)a(z)} \right\}$ - set of assessments of subsystems failure probabilities, components, elements and their interrelations CTS and DM recommendations;

$D \left\{ D_{S(C,E)n(m)}, D_{IS(C)a(z)} \right\}$ - - set of damage assessments from failures of subsystems, components, elements and their interrelations CTS and DM recommendations;

$W_{\vartheta_{S(C,E)n(m)}}^f$ - performance assessment (partial or complete) of subsystems, components, elements and DM recommendations;

$W_{\omega_{IS(C)a(z)}}^f$ - performance assessing (partial or complete) of intercomponent (intersystem) connections and DM recommendations:

RE – set of refined specific estimates TC parameters of subsystems, components, elements and their CTS mutual connections, decisions made ($re_1, \dots, re_N \in RE$);

SR – saved values of many updated estimates technical system parameters of subsystems, components, elements and their CTS mutual connections, decisions made;

DR - diagnosis and DM recommendations [13];

S, C, E – subsystem, component, element of CTS;
 IS, IC – intersystem, intercomponent communication;
 n, m – number, hierarchical level in CTS;
 a – intercomponent number;
 z – interconnection number.

$$R_{S(c,e)_{n(m)}} = \{r_{S(c,e)_{n(m)}} \mid s(c,e) = \overline{1}, \overline{S(C,E)}, n_{S(c,e)} = \overline{1}, \overline{N_{S(C,E)}}, m_{S(c)} = \overline{1}, \overline{M_{S(C)}}\}, \quad (2)$$

$$R_{I_{S(c)a(z)}} = \{r_{I_{S(c)a(z)}} \mid i_{S(c)} = \overline{1}, \overline{I_{S(C)}}, a = \overline{1}, \overline{A}, z = \overline{1}, \overline{Z}\},$$

where $r_{S(c,e)_{n(m)}}$ - CTS failures risk of subsystems, components, elements;

$r_{I_{S(c)a(z)}}$ - CTS failures risk of intersystem, intercomponent connections;

$n_{S(c,e)}$ - number of the subsystem, component, element in the CTS;

$m_{S(c)}$ - intersystem hierarchical level CTS intersystem, intercomponent number;

$N_{S(C,E)}$ - CTS subsystems, components, elements number;

$M_{S(C)}$ - hierarchical levels CTS intersystem, intercomponent connections number;

A - intercomponent connections number;

Z - intersystem connections number.

$$P_{S(C,E)_{n(m)}} \cdot \lambda(t)_{S(C,E)_{n(m)}} = \frac{\alpha_{S(C,E)_{n(m)}} \cdot \exp(-\alpha_{S(C,E)_{n(m)}} \cdot T_{S(C,E)_{n(m)}})}{\exp(-\alpha_{S(C,E)_{n(m)}} \cdot T_{S(C,E)_{n(m)}})} = \alpha_{S(C,E)_{n(m)}}, \quad (3)$$

$$P_{I_{S(C)a(z)}} \cdot \lambda(t)_{I_{S(C)a(z)}} = \frac{\alpha_{I_{S(C)a(z)}} \cdot \exp(-\alpha_{I_{S(C)a(z)}} \cdot T_{I_{S(C)a(z)}})}{\exp(-\alpha_{I_{S(C)a(z)}} \cdot T_{I_{S(C)a(z)}})} = \alpha_{I_{S(C)a(z)}}$$

where λ - failure rate;

α – distribution parameter, $\alpha \approx 1/\hat{T}_o$, $\hat{T}_o$ – mean time to failure estimation.

Quantifying damage from failure $n(m,e)$ subsystem, component, element to determine the risk of failure:

$$D_{S(C,E)_{n(m)}} = \{d_{S(c,e)_{n(m)}} \mid s(c,e) = \overline{1}, \overline{S(C,E)}, n = \overline{1}, \overline{N}, m = \overline{1}, \overline{M}\}, \quad (4)$$

where $d_{S(c)_{n(m)}}$ - damage from failure of a component subsystem, CTS element.

Quantifying damage from failure $a(z)$ intersystem (intercomponent) communication to determine the risk of failure:

$$D_{I_{S(C)a(z)}} = \{d_{I_{S(c)a(z)}} \mid i_{S(c)} = \overline{1}, \overline{I_{S(C)}}, a = \overline{1}, \overline{A}, z = \overline{1}, \overline{Z}\}, \quad (5)$$

where $d_{I_{S(c)a(z)}}$ - damage from failure of intersystem (intercomponent) communication.

Performance of subsystems, components, elements at different degrees of its loss:

$$W_{\nu_{S(C,E)_{n(m)}}}^f = \{W_f^{<n_{S(c)}, m_{S(c)}>} \mid f = \overline{0}, \overline{1}; n_{S(c,e)} = \overline{1}, \overline{N_{S(C,E)}}; m_{S(c)} = \overline{1}, \overline{M_{S(C)}}\} \quad (6)$$

Performance of intercomponent (intersystem) communication at different degrees of its loss:

$$W_{\omega_{IS(C)a(z)}}^f = \{W_f^{<a,z>} \mid f = \overline{0,1}; a = \overline{1,A}; z = \overline{1,Z};\} \quad (7)$$

There are known methods for extracting precedents and their modifications. The most common methods are: determination of the nearest neighbor (NN – Nearest Neighbor) [16]; extracting precedents based on decision trees [17]; extracting precedents based on knowledge [16, 17]; extracting precedents taking into account their applicability [18]. The listed methods use different metrics.

Among the main metrics, the method of determining the nearest neighbor was used, which makes it possible to easily calculate the degree of similarity of the current problem situation and precedents with BP. The nearest neighbor method uses a simple coordinate-by-coordinate comparison of the current situation with a precedent, where each parameter for describing precedents is considered as one of the coordinates. The DCT distance between the point corresponding to the current situation and the point corresponding to the precedent is determined. The effectiveness of the nearest neighbor method largely depends on the choice of metric. If the case C and the current problem situation T are given in an n-dimensional property space, then the degree of similarity or proximity $S(C,T)$ of the case C and the current situation T can be determined using one of the metrics that determine the distance between two points x_i^C and x_i^T , in particular, the Euclidean distance:

$$D_{CT} = \sqrt{\sum_{i=1}^n (x_i^C - x_i^T)^2} \quad (8)$$

To determine the similarity degree (SIM) value, the maximum distance is calculated D_{max} in the selected metric using the boundaries of parameter ranges to describe use cases. After this, the value of the degree of similarity is determined using the boundary of the parameter ranges to describe the initial and final precedents, $i = 1, \dots, n$. The degree of similarity value can be calculated as follows:

$$SIM = 1 - D_{CT} / D_{max} \quad (9)$$

5. IIS system with CBR implementation

The implementation of the IIS system with CBR (Fig. 4) connects the vehicle diagnostic results with a database (DB), KB, expert system containing calculated, experimental, and data obtained by experts during CTS operation.

IIS contains: interface module; KB with a library of precedents and a DB; request formalization module; module of recommendations for ensuring the effectiveness of the CTS; libraries of structural diagrams of ship CTS; module for formalizing expert assessments; knowledge formalization module. The implementation of the developed strategy in the IIS system is ensured by targeted actions in accordance with decision support to search for failures of subsystems, components, elements and their mutual connections based on vehicle established assessments.

The KB model is production-based; in terms of implementing its software functionality, it is object-oriented. The developed knowledge base is represented by rules obtained on the basis of data mining (multi-level hierarchical structure of the knowledge base tree), expert assessments, results of applying diagnostic models for the CTS TC, and functioning in accordance with the developed decision-making sequence (Fig. 3) and taking into account the CTS TC matrix (Fig. 1). The entire list of data and expert assessments are received from the database upon request to the KB. As a result, at the output of the knowledge base, estimates of the TC of subsystems, components, elements and their mutual connections are formed. TC assessments go to the module of recommendations for ensuring the effective operation of the CTS and then to the decision maker for managing the TC of a complex system. The database contains: CTS DB; DB of

failure risk criteria; DB TC CTS; DB of degradation processes; DB of measures to reduce the risk of CTS failures.

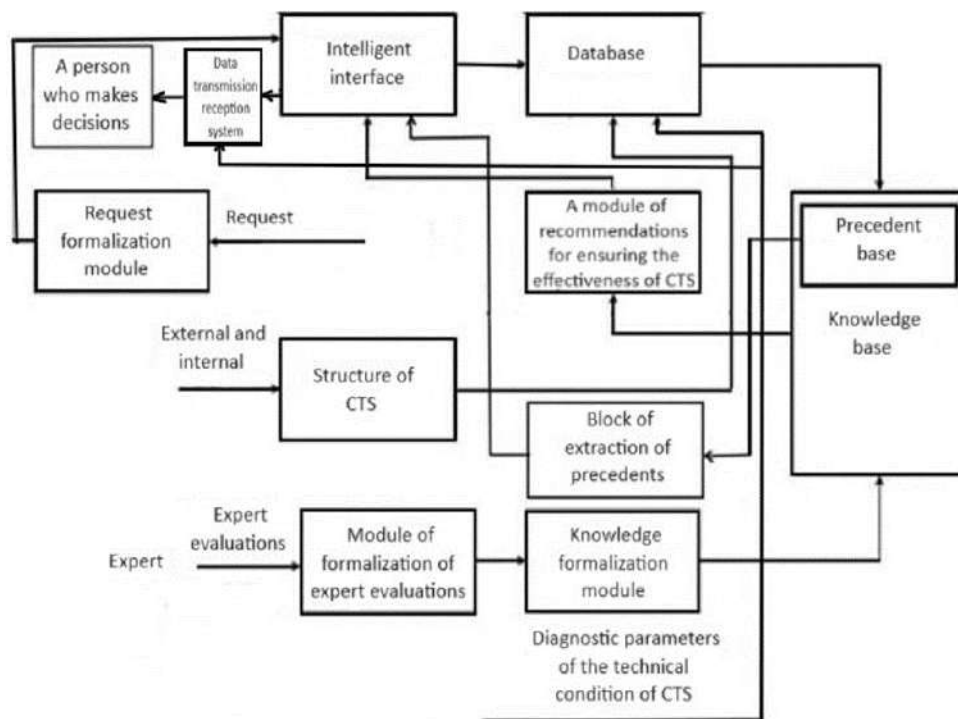


Figure 4: Block diagram of the IIS system with CBR assessment, forecasting and control of the CTS vehicle

The library of precedents consists of rule base (RB) of incidents, RB of emergency situations. Diagnosis data of a problem situation (complete or partial failures of equipment and their mutual connections between CTS and operability), obtained separately from the IIS, enters the IIS with CBR assessment, forecasting of the CTS TC. Taking into account the diagnostic data, the knowledge base and RB receive an established assessment of the technical system of a complex system, and recommendations are formed for decision-making by the D. Based on the assessment results, the TC of a complex system is predicted. The output may include a list of completed actions, additional comments, and links to other use cases.

The hierarchical structure of the key program logic of fragments of data processing modules according to precedents and solutions is shown in Fig. 5. To form a level of abstraction and ensure polymorphism when implementing the behavior of objects that implement different stages of data processing, three interfaces are implemented.

IData - forming a path to the location input data set, initializing data structures and collections, converting data to a single normalized form with checking for missing rows, setting up a structure. IOntology - storage of structures of attributes, classes and relationships, formation of an assembly of the ontology structure with its verification and validation. IPrecedent - operating with the properties situations scenarios, problems and solutions, as well as the formation of processes for assembling a precedent, saving and serializing it, checking for logical consistency. Each has a different implementation of the `logger()` method to provide processes for logging the intermediate results of computational operations over time. The DataLoader class implements the IData interface, performing the task of loading data into the system and the operations of forming precedents collections, checking data integrity, performing the necessary transformations, filtering and aggregation, as well as setting the structure and issuing notifications based on the results of completed actions in the status bar.

The OntologyMaker class implements the IOntology interface, operating a private collection dynamic array ontology for aggregating individual ontology elements and is intended to form the logical basis of the system structure when constructing each individual use case, providing visualization of the ontology in a graph-oriented form.

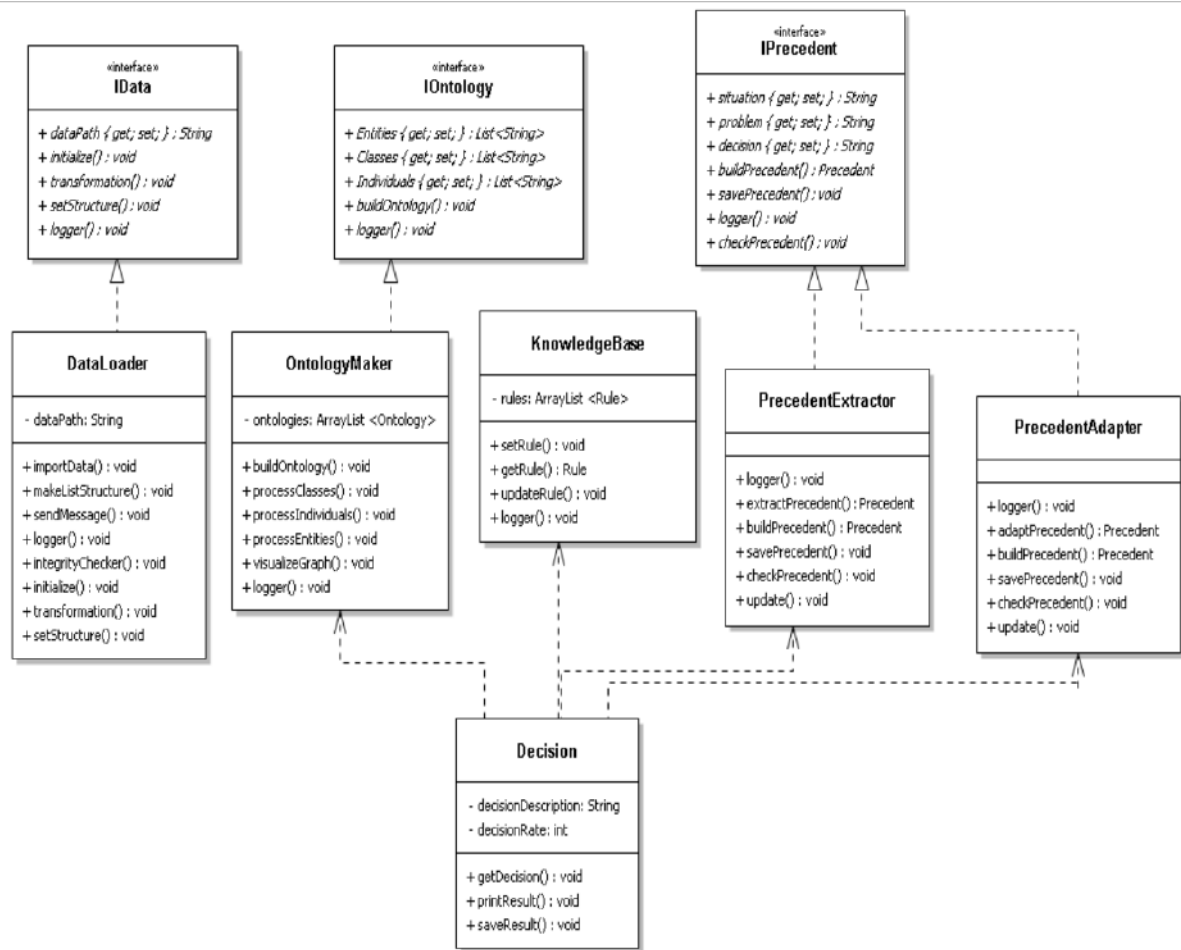


Figure 5: Key program logic hierarchical structure of data processing modules fragments by precedents and solutions

The PrecedentExtractor and PrecedentAdapter classes implement the IPrecedent interface, overriding methods for managing precedent data for use in the extraction and adaptation procedures performed in the extractPrecedent() and adaptPrecedent() methods, respectively, the result is a Precedent object. The KnowledgeBase class is separately implemented to provide CRUD operations for managing rules. Based on the use of these class instances a Decision object is formed, the state of which is described using the private properties decisionDescription and decisionRate, and the behavior is expressed using public methods for generating a decision object, outputting it, and saving the results.

For the software implementation of the IIS system with a CBR cycle, the Visual Studio development environment, the .NET 4.7 framework, the technology for creating WinForms graphical interfaces, as well as functional libraries for supporting Hermit ontologies and working with JSON were used [19]. The interface of the main form of the software system with a tab for managing the process of creating precedents within the framework of the proposed CBR cycle for assessing, forecasting and managing a vehicle, using the example of a ship's CTS [20] is shown in Fig. 6.

A main menu is provided for navigation between the processes of connecting data sources and knowledge base (File item), data management and processing (Data processing item), selection of computational operations and their implementation (Operations item), configuration of system modules and settings for its operation (Settings item), issuing reference information on the operation of the system (Help item). The interface provides four tabs for managing use cases, ontologies, rules and solutions. The precedents tab contains two panels.

The left panel contains a component for a structure hierarchical representation of the considered ship CTS and their subsystems, components and elements in the form of a tree of nodes. A field is provided to display a brief set of metadata for the precedent selected in the tree. The form contains buttons for opening a window for a detailed description of parameters,

constructing a descriptive summary cross-table for all values of ship CTS and their equipment, as well as buttons for initializing the processes of opening forms for extracting, adapting and validating (checking) compiled precedents. To launch the procedure for saving data based on the use case and serializing objects, use the Save button.

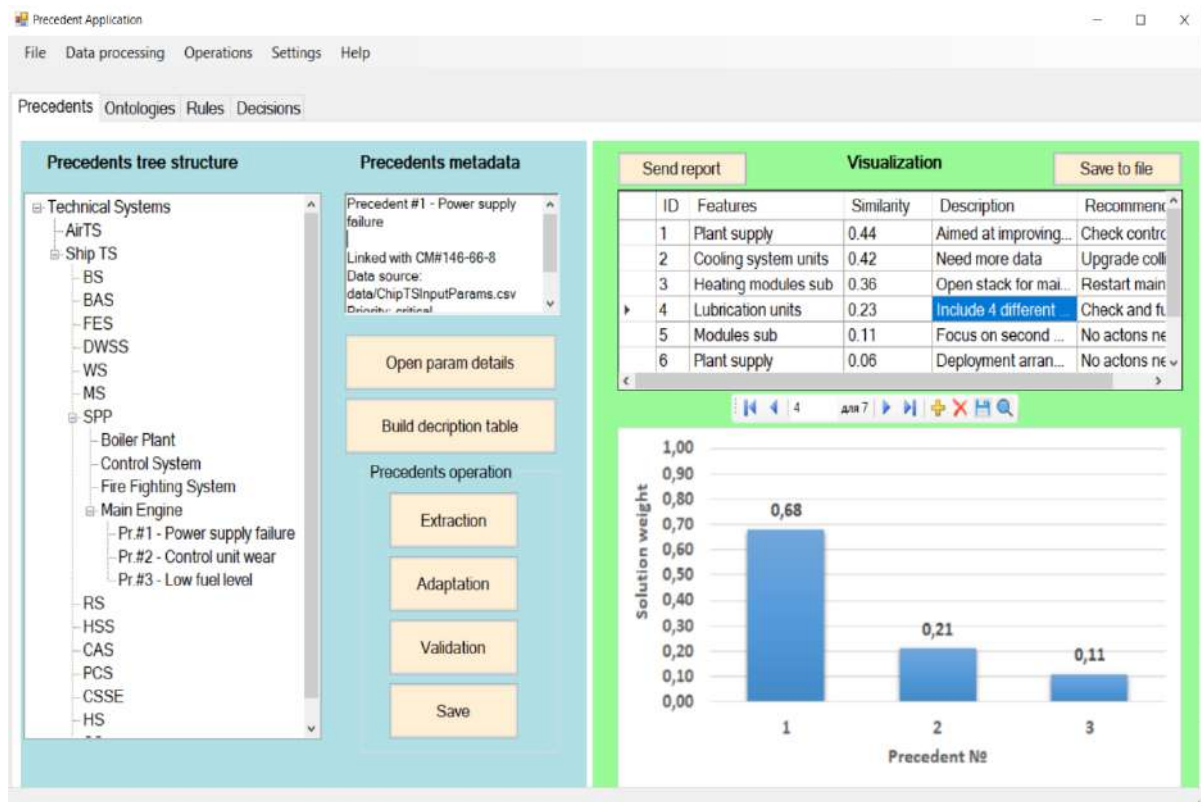


Figure 6: Software system main form interface with a tab for managing creating precedents process

The right panel of the form is intended to display a table of generated precedents with an assessment of their degree of similarity, description and a short set of standard recommendations. For ease of management, a component for quick table navigation with support for CRUD operations and search is provided. Below is a visualization component of the highest priority options for adapting precedents to a specific operating scenario for the CTS in question after all analytical procedures have been completed. The buttons for sending a report and saving to a file provide the ability to locally save visualization results in pdf and csv formats, respectively.

6. Experimental system research

The ship power study subsystems risk of failures assessment results formed on the basis of the compiled precedents, are shown in Fig. 7.

The results of functional elements and functional connections failure risk prediction results, for example, for the ship power plant main engine subsystem, can be seen in the block of the failure risk prediction view interface when navigating to the predictions web page (Fig. 8).

The disadvantage of the method of precedents with a CBR cycle is an increase in the time of searching for the nearest precedents. Therefore, a comparative analysis of the search time for the nearest precedent was conducted depending on precedent database size based on data caching when the data structure was initialized in collection form, which is based on an associative array.

The graph reflecting the time of determining the TC of a complex system from the number of precedents is shown in Fig. 9. Time spent on finding the nearest precedent with 10,000

precedents in the BZ was about 370 ms. The first closest precedent out of 5000 precedents was obtained in about 50 ms.

Precedents Ontologies Rules Decisions Risk assessment		
<i>System name</i>	<i>Risk assessment value</i>	<i>Details</i>
BS	<u>77</u>	
BAS	<u>65</u>	
FES	<u>14</u>	
DWSS	<u>21</u>	
WS	<u>54</u>	
MS	<u>12</u>	
SPP	<u>67</u>	Boiler Plant - 16% Control System - 22% Fire Fighting System - 11% Main Engine - 51%
RS	<u>33</u>	
HSS	<u>21</u>	
CAS	<u>26</u>	
PCS	<u>40</u>	
CSSE	<u>29</u>	
HS	<u>11</u>	
SS	<u>5</u>	

Figure 7: Interface of the risk assessment form for the analyzed subsystems

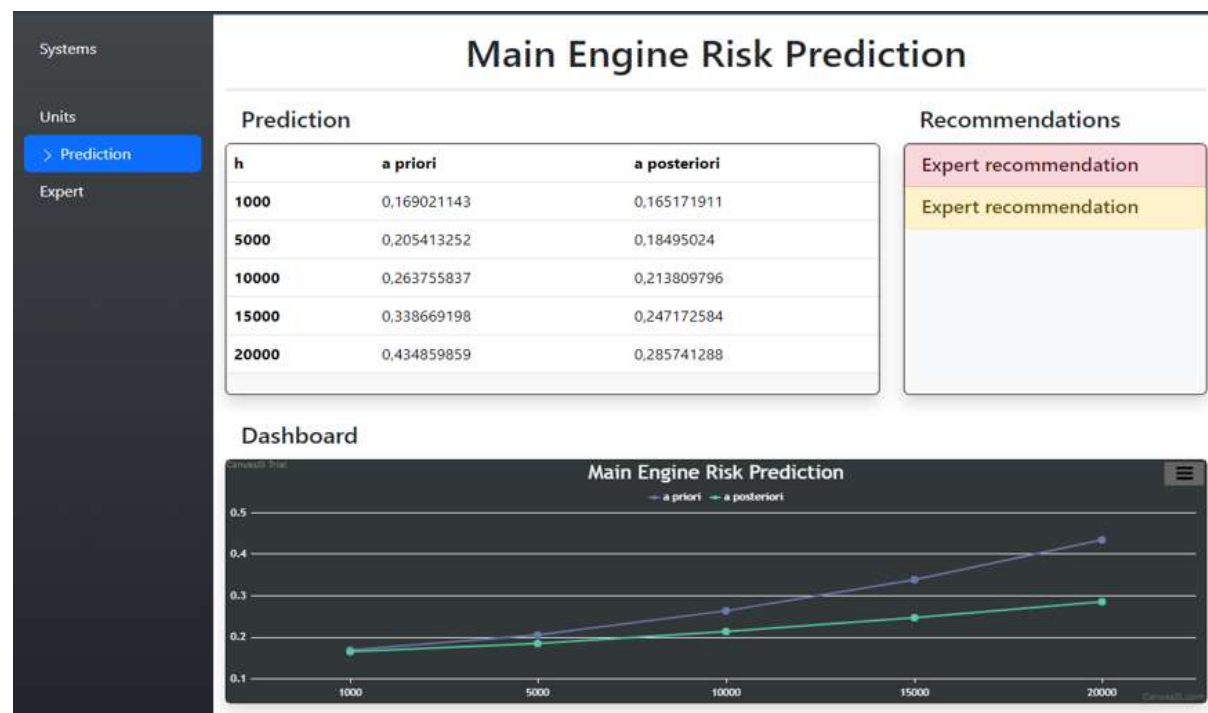


Figure 8: Main engine subsystem failure risk prediction interface unit

With an increase in the precedents number in the precedents library determining CTS TC time increases, but it does not significantly affect the overall time spent on evaluating the TC of the researched SPP subsystems.

Despite such a shortcoming, studies have shown the possibility of applying the method of reasoning based on CBR precedents, and it is expedient to use it for decision-making in real operating conditions. The developed IIS has high performance.

In order to estimate the time spent on the KB formation within the implementation framework of the proposed method by IIS means a comparison of the execution time of

computational processes at the start of the system in three modes (Fig. 10): single-threaded (1 flow), two-threaded (2 flow) and four-threaded (4 flow) was carried out.

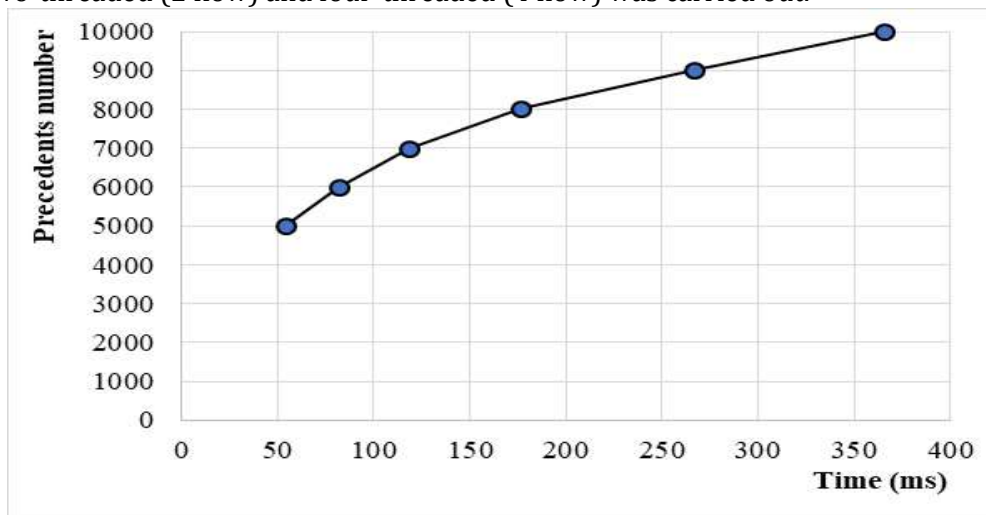


Figure 9: The time to determine the TC of a complex system is based on the number of precedents

It should be noted the general exponential nature of the time dependence of computational processes estimations execution forecasting SPP TC on the KB precedents number.

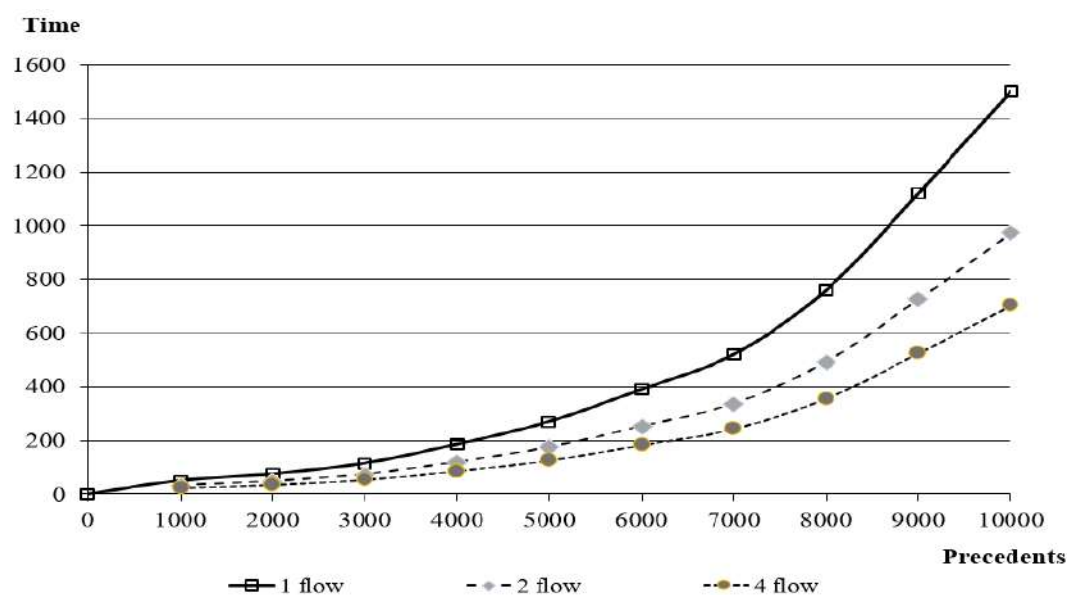


Figure 10: Execution time of computational processes depending on the number of generated precedents

Within distributed mode it becomes possible to reduce the time spent by up to 28% when using two mutually isolated data streams, and up to 42% in the case of dividing the computing load into four separate streams.

7. Conclusion

An IIS with CBR is proposed, designed for effective assessment and prediction of the TC of complex systems for critical applications by ensuring IIS speed. The effective functioning of IIS with CBR is based on the use of case-based reasoning. IIS with CBR consists of: interface module; KB with a library of precedents and a DB; request formalization module; module of recommendations for ensuring CTS effectiveness; CTS block diagram libraries; modules for formalizing expert assessments and formalizing knowledge. IIS experimental studies for assessing and predicting the technical characteristics of complex systems showed that the time

required to find the nearest precedent with 10,000 precedents in the knowledge base was about 370 ms. The decision-making sequence, using the proposed IIS system with a CBR cycle, taking into account the operations of processing and structuring data according to precedents within the framework of the functioning of the developed application software system, has high performance, ensures work with incomplete information, versatility and learning ability when supporting decision-making. According to the distributed computing mode, it becomes possible to reduce time costs by up to 28% when using two data streams isolated from each other, and up to 42% when the computing load is divided into four separate data streams.

When operating the proposed IIS, partial and complete failures of the operability of subsystems, components, elements and their mutual connections in CTS are taken into account. Increased operational efficiency of the CTS is ensured by CTS speed evaluation and forecasting, as well as by the actions of decision makers aimed at making decisions on pre-failure maintenance complex systems equipment at early development failure stages.

References

- [1] Benin D.M. Support systems accept solutions. Triada, 2022.
- [2] E.A. Gavriluk, Information Decision-Support System on the Basis of the Method of Diagnostics and Control of the Technical State of Industrial Equipment Engineering, Computer Science Dynamics of Systems, Mechanisms and Machines (Dynamics) (2018). DOI: 10.1109/Dynamics.2018.8601472
- [3] V. Vychuzhanin, N. Rudnichenko, Metody informatsionnykh tekhnologiy v diagnostike sostoyaniya slozhnykh tekhnicheskikh sistem, monografiya, 2019.
- [4] P. Zhang, Z. GaoCao, L. Dong, Marine Systems and Equipment Prognostics and Health Management. Systematic Review from Health Condition Monitoring to Maintenance Strategy. Machines 10 (2022) 72. DOI:10.3390/machines10020072
- [5] R. Garcia, M. Parades-Valverde, Exploring Intelligent Decision Support Systems, Current State and New Trends, Springer International Publishing AG, (2018).
- [6] V. Vychuzhanin, N. Rudnichenko, Z. Sagova, M. Smieszek, V. V. Cherniavskiy, A. I. Golovan, M. V. Volodarets, Analysis and structuring diagnostic large volume data of technical condition of complex equipment in transport. IOP Conference Series: Materials Science and Engineering, Volume 776, 24th Slovak-Polish International Scientific Conference on Machine Modelling and Simulations - MMS 2019, 3-6 September 2019, Liptovský Ján, Slovakia, 2019 pp.1-11. DOI:10.1088/1757-899X/776/1/012049.
- [7] C. Shaoli, Yi. Jianjun, J. Hui, Z. Xiaomin, Ontology and CBR based automated decision-making method for the disassembly of mechanical products. Advanced Engineering Informatics 30 3 (2016) 564-584.
- [8] M. Siepermann, Risk Cost Accounting and Bottom Price Calculation – A Risk Management Information System, in International Conference on the Economics of GrIIS, Clouds, Systems, and Services, September, Springer, Cham, 2017, pp. 43-55.
- [9] R. Majid, S. A. Mir, Advances in Statistical Forecasting Methods: An Overview. Economic Affairs 63 4 (2018) 815-831.
- [10] F. Lan, Y. Jiang, Performance Prediction Method of Prognostics and Health Management of Marine Diesel Engine, in Proceedings of the 2020 3rd International Conference on Applied Mathematics, Modeling and Simulation, Shanghai, China, 20–21 September 2020.
- [11] V. Vychuzhanin, N. Rudnichenko Complex Technical System Condition Diagnostics and Prediction Computerization. CMIS-2020 Computer Modeling and Intelligent Systems. Ceur-ws.org. 2608.2020., 2020, pp.1-15.
- [12] E.C. Harrington, The desirability Function. Industrial Quality Control, 21 10 (1965) 494–498.
- [13] A. F. Berman, Method of synthesis and analysis of fault trees based on the concepts of mechanism and kinetics of events. Problems of risk analysis, 15 3 (2018) 62-77.
- [14] De. Sumana, B. Chakraborty, A Novel CBR-Decision Tree Based Intelligent Car Fault Diagnosis System (CFDS). International Journal of Recent Technology and Engineering 8 (2019) 2180-2193. DOI:10.35940/ijrte.B2390.078219.

- [15] D. Glukhikh, I. Shchinnikov, I. Glukhikh, Using hybrid-CBR for intelligence monitoring and decision-making systems on SMART grid. *Intelligent Decision Technologies*. 16 (2022) 1-8. DOI:10.3233/IDT-210239.
- [16] G. Luger, *Artificial intelligence: structures and strategies for complex problem solving*, (2009).
- [17] I.D. Watson, F. Marir, Case-based reasoning: A review. *The Knowledge Engineering Review*, 9, 4 (1994) 355-381.
- [18] B. Smyth, M. Keane, Retrieving adaptable cases. *The role of adaptation knowledge in case retrieval*. Springer Berlin 837 (1994) 209-220.
- [19] Y. Shunkun, B. Chong, Li. Xing, T. Lin, T. Dongxiao, Optimized fault diagnosis based on FMEA-style CBR and BN for embedded software system. *The International Journal of Advanced Manufacturing Technology*. 94 (2018). DOI:10.1007/s00170-017-0110-y.
- [20] V. Vychuzhanin, N. Shibaeva, A. Vychuzhanin, N. Rudnichenko, Intellectualization Method and Model of Complex Technical System's Failures Risk Estimation and Prediction. *Ceur-ws.org*, 3392, CMIS-2023. pp.130 – 140, <https://ceur-ws.org/Vol-3392>.

Rational Air Transportation Technologies Resources Recombination on Condition of the Generalized Values

Andriy V. Goncharenko and Viktor V. Iliushyn

National Aviation University, 1, Liubomyra Huzara Avenue, Kyiv, 03058, Ukraine

Abstract

The presented paper is dedicated to the attempt of a theoretical description of the aviation transport technologies elements resources rational use. The maximal value of the generalized “positive” effect is chosen in the view of the optimization criterion. For two competing resources their optimal recombination is found at the segment of the mathematically linearly described solution obtained on condition of the fixed total amount of the production of the aviation transport technologies services. Such fixed total production amount acts like iso-perimeter constraint condition in the stationary problem with the constant capacities/productivities of the competing resources. On the other hand, the generalized “positive” effect parameter is also considered having two competing components, which values ensure yielding the optimal resources recombination. Illustrative example simulation is performed. Necessary diagrams are plotted.

Keywords

Aviation, transport, technology, resource, recombination, entropy, preferences, optimization, simulation

1. Introduction

Computer simulation of the aviation transport technologies elements is an urgent task. Airlines’ and aviation business enterprises’ resources rational use is important in regards with both aircraft technical operation [1] and the aircraft engines reliable functioning [2]. Based upon the ideas of [1, 2]—the aircraft and its powerplant elements reliability and maintenance require decent resources distributions and combinations. A significant amount of the resources are spent upon keeping the predictable level of the aviation risks in the acceptable margins [3, 4].

Analyzing the objectives of the aviation transport technologies elements functioning, it is possible to conclude that, in any case of the airlines operation, the management of the air companies pursues the goals that have some “positive” effects in the view of the managerial endeavors leading to the expected utilities of the individuals’ choice [5, 6].

The required resources recombination, when running an aviation business, also happens at the circumstances of the more or less uncertainties of the different kinds, such as e.g., the aircraft types used, their number, applicable aviation transport technologies elements resources etc.. Those kinds of uncertainties can be evaluated with the use of the entropy approaches, likewise in the references of [7 – 9], having become very popular recently [10].

In conjunction with the economical models of [11], the entropy methods of [7 – 10] resulted in the subjective analysis theory [12], which allows solving various types of the applicable problems, similar to the stated in the references of [13 – 16].

Air transportation management and aviation transport technologies elements resources recombination basic problems have already been set in [17 – 20] implementing the provisions of the previous researches [1 – 12].

According with the described concepts of [1 – 12, 17 – 20] the scientific gap, which needs its resolution, is an elaboration of a plausible theoretical approach to the actual and important problems of the rational aviation transport technologies resources recombination setting,

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ andygoncharenko@yahoo.com (A. V. Goncharenko); 8732463@stud.nau.edu.ua (V. V. Iliushyn)



0000-0002-6846-9660 (A. V. Goncharenko); 0009-0006-2923-1752 (V. V. Iliushyn)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

especially in regards with the analytical decision making process supporting based upon the computer modeling advantages.

The problem setting of the presented study is about the theoretical research centered upon the rational aviation transport technologies resources recombination, following the reference of [18].

Thus, the goals of the paper are generally describe possible optimization of the resources recombination and computationally illustrate the theoretically obtained solutions.

2. Resources recombination possible optimization

Optimization approach of the presented research is close, in the ideological considerations, to the conditional optimization, analogously to the problem settings implying the problem statements subject to the Lagrange uncertainty multipliers constraints. Nevertheless, herewith, the study uses acceptable simplifications that differ from variational problem of [18], yielding the same theoretical result, though, for the potentially possible resources distributions and combinations.

2.1. Basic concept

Amongst the multiple number of the aviation transport technologies elements is proposed to select and generalize such two competing ones that allow theoretically accumulate resources in considerations of the resources rational recombination.

In assumption that the mentioned above aviation transport technologies elements are determined in terms of the key resources, one can construct the following model, [18]:

1. The total amount of the aviation transport technologies services production:

$$N. \tag{1}$$

2. The capacity/productivity of the first competing aviation transport technologies elements resource is

$$p_1. \tag{2}$$

3. The capacity/productivity pertaining with the second competing aviation transport technologies elements resource is

$$p_2. \tag{3}$$

Obviously, using the introduced parameters of (1) – (3), one can obtain the following equations of the stationary case:

$$\frac{N}{p_1} = a, \tag{4}$$

where a symbolizes the number of the resources of the first nature required for the aviation transport technologies services production in order to realize the air transport technologies services total amount of N : parameter (1), on condition of the first nature resources capacity of p_1 : parameter (2).

Analogously to (4), there is a possibility to describe the stationary process as

$$\frac{N}{p_2} = b, \tag{5}$$

where b portrays correspondingly the number of the resources of the second kind required, on the competing bases, for the aviation transport technologies services production in order to realize the air transport technologies services total amount of N : parameter (1), on the condition that the second type resources capacity is of p_2 : parameter (3).

The discussed generalized case of (1) – (5) has a well-known linear solution [18] for a possible combination of the resources of both types.

Indeed, let us say the total amount of N : (1), is realized/produced with/by the resources of the two types together (simultaneously). That means

$$N = N_1 + N_2, \tag{6}$$

where N_1 is the portion of the total amount realized/produced with the help, or by the use, of the first competing kind of the resources; and N_2 is the rest of the total amount realized by the second type resources correspondingly.

Then, according to (2) and (3),

$$\frac{N_1}{p_1} = n_1, \quad (7)$$

where n_1 is the number of the units of the first competing kind of the resources that took part in the realization/production of the corresponding portion of N_1 : (6), of the total amount of N on the stationary condition of their production/capacity abilities of p_1 : parameter (2).

On the other hand, likewise in (7),

$$\frac{N_2}{p_2} = n_2, \quad (8)$$

where n_2 is the number of the units of the second competing kind of the resources that was required in/for the realization/production of the rest amount of N_2 .

Substituting components obtained from (7) and (8) for their values into (6), it can be notated that

$$N = p_1 n_1 + p_2 n_2. \quad (9)$$

At the fixed values of (1) – (3), the relation of (9) allows researching dependencies between two variables of n_1 and n_2 .

Considering one of those variables, for example n_1 , as the independent variable, the other n_2 can be expressed from (9) as a function of it.

Namely,

$$n_2(n_1) = \frac{N - p_1 n_1}{p_2}. \quad (10)$$

In terms of (4) and (5) the equation of (10) can be rewritten as [18, (5)]:

$$n_2(n_1) = b - \frac{b}{a} n_1. \quad (11)$$

2.2. Optimization technique

The simplest solution of (1) – (11) opens a wide range of conditional optimization problems for the segment of $[b, a]$.

Rational air transportation technologies resources recombination on condition of pertaining to the segment of $[b, a]$ can be conducted based upon the options effectiveness assessments.

Using the options effectiveness assessments parameter in the view of

$$P = D - R, \quad (12)$$

where P is the “positive” effect parameter value; D is the “developmental” component; R is the “regressive” component.

The models implemented for the equation (12) components can of different nature and content; similar to, for instance, the proposed in [18, (6) – (11)] for the activities splitting.

As to the presented study

$$D_i = s_i p_i n_i, \quad (13)$$

where D_i is the i^{th} resource kind component, this refers to the corresponding feature of the additive factor for both resources, i.e.

$$D = D_1 + D_2; \quad (14)$$

s_i is the i^{th} resource type component coefficient/function influencing the “developmental” components of the resources; p_i and n_i are the parameters of (1) – (5) and (7) – (11) in respect.

As to the values entering (13), those can be functions of different interdependencies.

For the developed herewith computer simulation it is proposed the following

$$s_i = k_{s_i} \prod_{j=1}^m (n_i - n_{i_j}), \quad (15)$$

where k_{s_i} is an augmentation factor/index (also possibly a coefficient/function), for the i^{th} resource type component, scaling the effect of the parameter of n_i change/variation; n_{i_j} is the parameter of n_i value symbolizing the “zero” effect for the “developmental” components.

Concerning the “regressive” component R of the dependence (12), there are a few contemplations of the next sort.

$$R = \sum_{i=1}^2 R_i, \quad (16)$$

where R_i is, similarly to (14), the “regressive” component of R in respect to the resources taken into consideration.

However, in regards with the “regressive” components constructions, the assumed models emphasize their features as

$$R_i = k_{R_i} n_i, \quad (17)$$

where k_{R_i} is a growth factor/index (also possibly a coefficient/function), for the i^{th} resource kind component, scaling the effect of the parameter of n_i change/variation.

In turn, the escalation factor/index of k_{R_i} , of formula (17), implies the peculiarities of its own possibilities to change depending upon the n_i change/variation.

The proposed model is

$$k_{R_i} = k_{R_i}^{(0)} + \Delta k_{R_i} k_{r_i}^{n_i}, \quad (18)$$

where $k_{R_i}^{(0)}$ is a boundary/basic value of the growth factor/index; Δk_{R_i} is the range/diapason of the growth factor/index possible change/variation; $k_{r_i}^{n_i}$ is the variated component of k_{r_i} raised to the power of the parameter of n_i value.

In order to investigate the uncertainty of the optimal recombination of the aviation transport technologies resources choice, it is proposed to apply the entropy approach developed and implemented in the works of [17, 18] based upon the entropy paradigm of [7 – 9, 12].

It implies the use of the relative hybrid pseudo-entropy function, i.e.

$$H_r(n_1) = \frac{H_{\max} - H(n_1)}{H_{\max}} \frac{\pi_1(n_1) - \pi_2(n_1)}{|\pi_1(n_1) - \pi_2(n_1)|}, \quad (19)$$

where $H_{\max}$ is the maximal value of the entropy, in the considered case it equals $\ln(2)$; $H(n_1)$ is the current value of the entropy defined by the expression of

$$H(n_1) = - \sum_{i=1}^2 \pi_i(n_1) \ln[\pi_i(n_1)], \quad (20)$$

where $\pi_i(n_1)$ are the preferences functions:

$$\pi_i(n_1) = \frac{e^{\beta P_i(n_1)}}{\sum_{j=1}^2 e^{\beta P_j(n_1)}}, \quad (21)$$

where β is the cognitive parameter essential to forming the objective functional with the preferences functions [7 – 9, 12, 17, 18]; $P_i(n_1)$ are the resource-centered effectiveness, defined as (12).

2.3. Simulation

The initial data for computer modeling are as follows:

$$N = 1000, \quad p_1 = 10, \quad p_2 = 20. \quad (22)$$

Then, implementing the procedure of (1) – (11) one can obtain the linear solution results presented in the Figure 1.

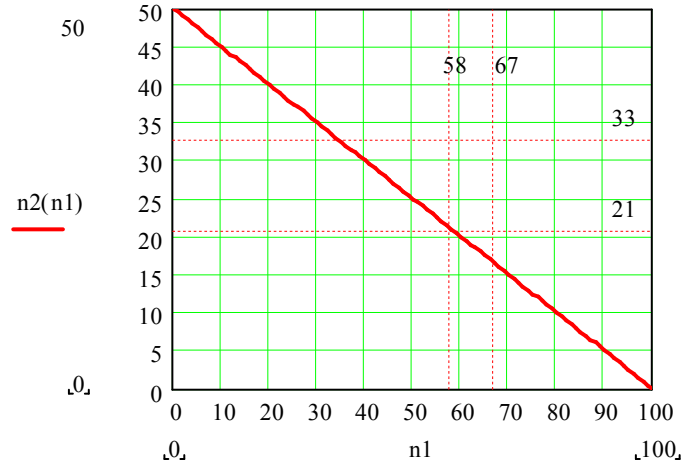


Figure 1: The resources interdependence

The segment of $[b, a]$ of the linear solution of (10) or (11) represented in the Figure 1 has the values of (4) and (5).

$$\frac{N}{p_1} = a = 100 \quad \text{and} \quad \frac{N}{p_2} = b = 50, \quad (23)$$

correspondingly depicted in the abscise n_1 and ordinate n_2 axes.

Now, the problem is to determine the optimal recombination of the resources along the solution segment of $[b, a]$.

In order to find the desired optimal air transport technologies resources recombination one can use the models of (12) – (18).

Concerning the “developmental” components of D , the accepted and used data are following:

$$k_{s_1} = -1, \quad n_{1_1} = 0, \quad n_{1_2} = 100, \quad k_{s_2} = -2, \quad n_{2_1} = 0, \quad n_{2_2} = 50. \quad (24)$$

The curves plotted for the augmentation functions of s_i , calculated by the formulae of (15), are shown in the Figure 2.

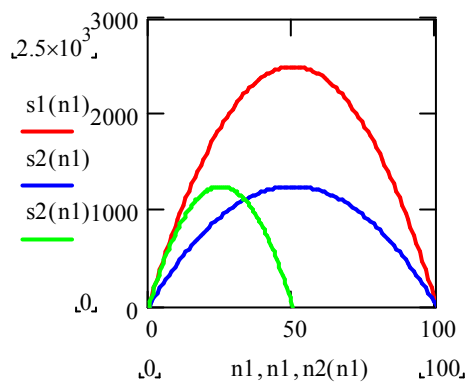


Figure 2: The augmentation functions for the “developmental” components

The third curve of $s_2(n_1)$, presented in the Figure 2, is plotted, as a phase portrait/curve (phase diagram) in the phase coordinates/space of $s_2(n_1)$ - $n_2(n_1)$, depending upon the phase coordinate of $n_2(n_1)$.

In such circumstances as (22) – (24), the computer modeling with the help of (13) and (14) gives the results illustrated in the Figure 3.

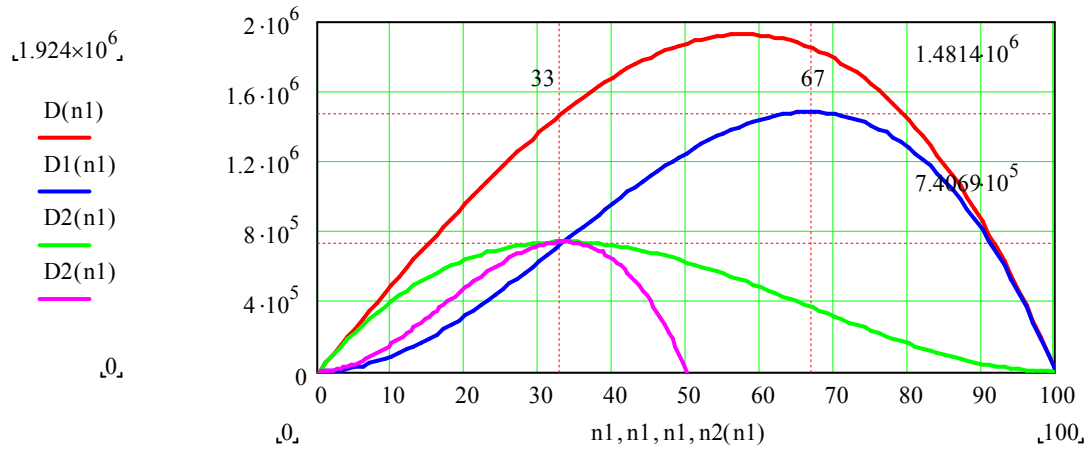


Figure 3: The “developmental” components of the aviation transport technologies resources recombination

The fourth curve of $D_2(n_1)$, presented in the Figure 3, is plotted, as a phase portrait/curve (phase diagram) in the phase coordinates/space of $D_2(n_1)$ - $n_2(n_1)$, depending upon the phase coordinate of $n_2(n_1)$.

It is also shown in the Figure 3, the extremal values of $D_1 = 1.4814 \cdot 10^6$ and $D_2 = 7.4069 \cdot 10^5$ which are ensured by the values of $n_1 = 67$ and $n_2 = 33$ correspondingly.

Those values are noticeable in the Figure 1 as well.

Next up are the “regressive” components of R .

The accepted data for the calculation experimentations with the parameters of the equations of (16) – (18) are following:

$$k_{R_1}^{(0)} = 2 \cdot 10^3, \quad \Delta k_{R_1} = 6 \cdot 10^3, \quad k_{r_1} = 0.954, \quad k_{R_2}^{(0)} = 3 \cdot 10^3, \quad \Delta k_{R_2} = 8 \cdot 10^3, \quad k_{r_2} = 0.9. \quad (25)$$

Thus, the results of the conducted computer modeling with the equations of (18) and data of (25) are presented with the diagrams in the Figure 4.

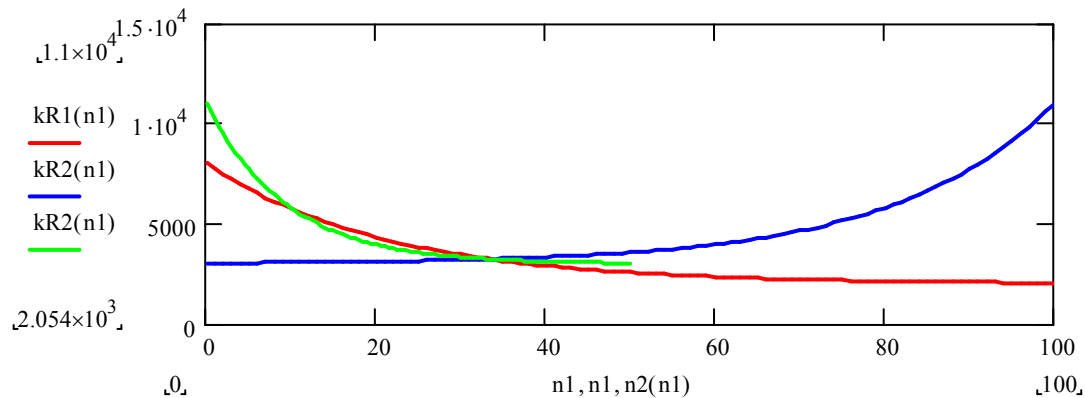


Figure 4: The escalation factors/indices of the “regressive” components of the aviation transport technologies resources recombination

The escalation factors/indices (as the functions of the aviation transport technologies resources units numbers) of k_{R_1} , of formulae (17) and (18), for k_{R_2} , depicted as $kR2(n_1)$ in the Figure 4, are shown in both dependencies; namely, for both, with respect to the only, so far, independent variable of n_1 and $n_2(n_1)$. The latter case, the curve number three as $kR2(n_1)$ in the Figure 4, is symbolized with the phase portrait/curve (phase diagram) in the phase coordinates/space of $kR2(n_1)$ - $n_2(n_1)$, depending upon the phase coordinate of $n_2(n_1)$.

Then, the diagrams of the “regressive” components of R , computed with the help of the formulae of (16), and (17), are shown in the Figure 5.

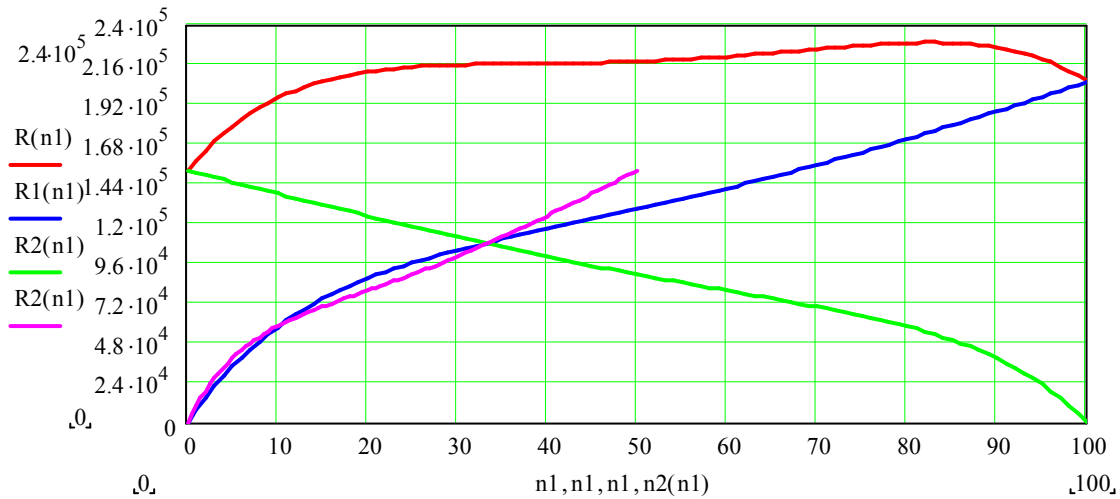


Figure 5: The “regressive” components of the aviation transport technologies resources recombination

The fourth curve of $R2(n1)$, presented in the Figure 5, is plotted, as a phase portrait/curve (phase diagram) in the phase coordinates/space of $R2(n1)$ - $n2(n1)$, depending upon the phase coordinate of $n2(n1)$.

At last, the “positive” effect parameter values of P , computed by the equation (12) components, are illustrated in the Figure 6.

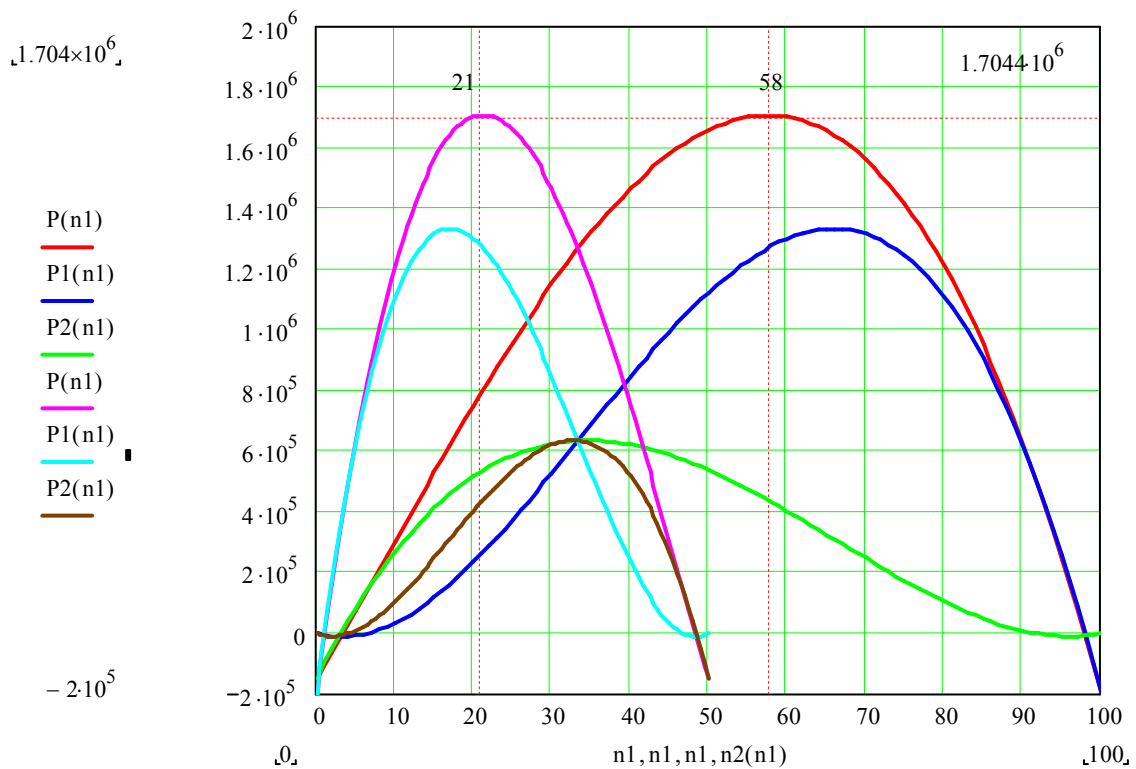


Figure 6: The “positive” effect parameter values of the aviation transport technologies resources recombination

The last three curves of $P(n1)$, $P1(n1)$, and $P2(n1)$, presented in the Figure 6, are plotted, as the phase portraits/curves (phase diagrams) in the phase coordinates/spaces of $P(n1)$ - $n2(n1)$, $P1(n1)$ - $n2(n1)$, and $P2(n1)$ - $n2(n1)$, depending upon the phase coordinate of $n2(n1)$.

It is visible from the plots illustrated in the Figure 6 that at a certain aviation transport technologies resources recombination (circumstances) the “positive” effect parameters values turn to the “negative” zone.

The optimal recombination of the aviation transport technologies resources is also depicted in the Figure 6, i.e. the extremal values of $P(n_1)=1.7044 \cdot 10^6$ are ensured by the values of the optimal recombination for $n_1 = 58$ and $n_2 = 21$ correspondingly.

This optimal recombination of the aviation transport technologies resources is shown in the Figure 1 as well.

Computer modeling of the uncertainty situation with the help of (19) – (21) yields the results represented in the Figures 7 – 9.

The data accepted for the simulation are as follows:

$$\beta = 3.6 \cdot 10^{-6}. \quad (26)$$

The resource-centered effectiveness: $P_i(n_1)$, are defined by (12), (14), and (16) as

$$P(n_1) = D(n_1) - R(n_1) = D_1(n_1) - R_1(n_1) + D_2(n_1) - R_2(n_1) = P_1(n_1) + P_2(n_1). \quad (27)$$

Preferences by (21) and entropy by (20) are shown in the Figures 7 and 8 respectively.

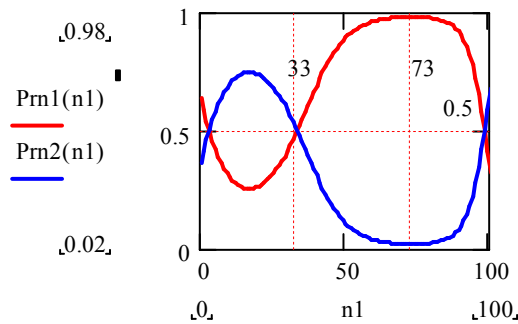


Figure 7: Preferences functions

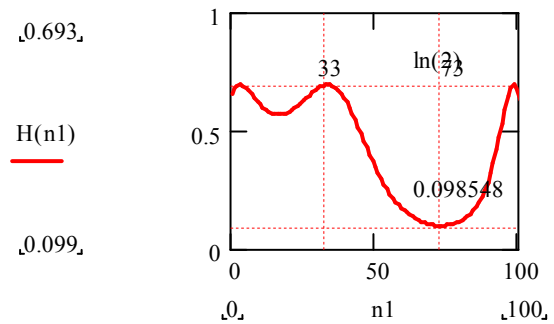


Figure 8: Entropy of preferences functions

The relative hybrid pseudo-entropy function by (19) in comparison to the traditional entropy is illustrated in the Figure 9.

3. Discussion

The demonstrated approach of (1) – (21) with the accepted data of (22) – (27) has a number of the generalized simplifying, although plausible, assumptions.

The simplified models make it easier finding the aviation transport technologies elements resources optimal recombination. It is accepted undisputable that the aviation transport technologies services production resources have their own subjectively preferred rational use as that was claimed in the references dealing with the individuals' choices, like in [5, 6], with respect to the economical issues marked at [11], subjective individual preferences uncertainty emphasized in [12].

One of the significant simplifications of the discussed model presented herein with this paper is the stationary condition. Dynamics of the resources optimal recombination must inevitably have its own consequences. Other important simplifications of the study are the suppositions of just one independent variable n_1 and independence of the resources capacities of p_i : (2) and (3) upon it. Introducing other independent variables, e.g. n_2 , as well as implying all other parameters dependences upon n_i in more elaborated model constructions, it will predictably change the models relations (1) – (11) and results of simulation shown in Figures 1 – 6.

Since it has been supposed the additive properties of (6), (9), and (12), and their components, both in respect to the generalized resources types and kinds of the effect parameter values, the expected solution will be complicated as well if all that have possible multiplicative effects (see and compare Figures 2 and 4).

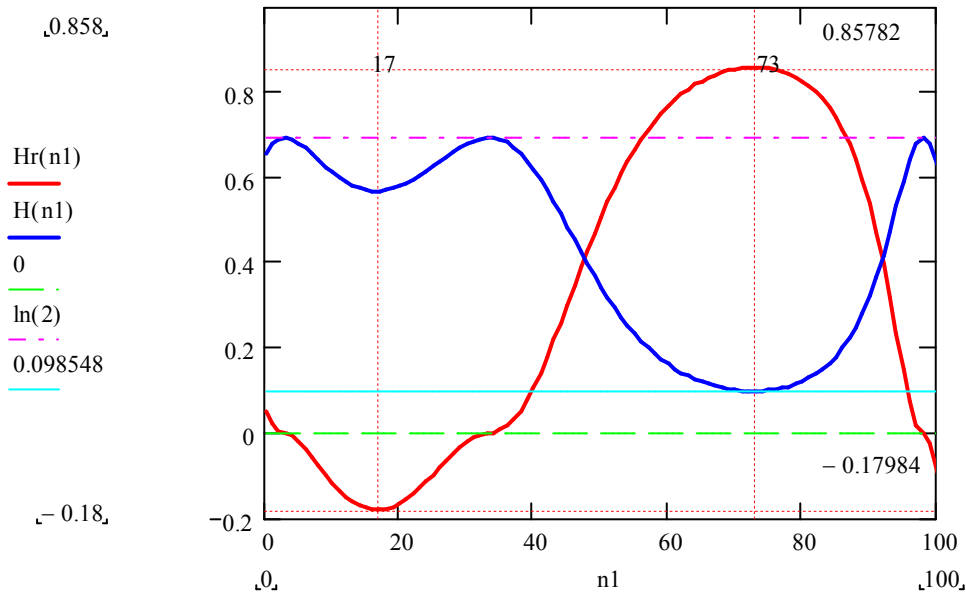


Figure 9: Entropy of preferences functions

Models of (13), and their components can differ in mathematical expressions. Herein it has been studied the “parallel” models, identically mathematically expressed.

Multiplicative effects are probable with the consideration of the individuals’ subjective preferences [12] in combinations with the dynamical issues. The study of the subjective preferences entropy influence upon the desired optimal air transport technologies resources recombination looks like a huge separate problem with the internal applicative incentives.

In this sense, the potential challenges and limitations associated with the implementing the proposed model in the diverse aviation contexts can be applicably demonstrated in the optimal distributions of the aircraft numbers of a certain or specified types over the airlines’ fleets. Such problems are supposedly prospective in the studies that are going to be prolonged in the future considerations dealing with the problem settings of conditional optimization.

Thus, the rational recombination may not belong with the solution presented in the Figure 1.

In conditions of the stated problem the solution with the values of $n_1 = 67$ and $n_2 = 33$ is impossible since it is not in the solution fragment (see Figure 1). It happens because that is not in the compliance with the theoretically developed model dependencies (1) – (21) and accepted calculation data of (22) – (25) too; although the mentioned above impossible solution delivers maximal values to the “developmental” components of D (see and compare Figures 1 and 3).

As to the uncertainty of the optimal recombination of the aviation transport technologies resources choice, it is worth saying that (19) – (21) used the parameter of (26) on the purpose of the contrast distinguishing between two alternatives of (27); the corresponding values of the preferences functions and entropy are visible in the Figures 7 and 8.

Herewith, it should be noted that the traditional view entropy of (20): [7 – 9, 12], is incapable to distinguish the directions of the certainty or uncertainty between the alternative preferences around the designated points (see Figure 8).

Such a lack of the informative deficiency is supposed to be compensated with the relative hybrid pseudo-entropy, for instance, around the argument values of “17” and “73”; “bad”, that is “negative”, vs. “good”, for the represented case study computer modeling “positive”, relative certainty (see Figure 9). In the illustrated interpretation zero value of the pseudo-entropy corresponds to the situation of the complete uncertainty.

4. Conclusion

The developed theoretical approach allowed discovering the rational aviation transport technologies resources recombination as a kind of an optimal solution in regards with the

generalized values. The segment of the linear solution has a maximal value of the generalized “positive” effect parameter in the framework of the accepted suppositions for the technologies fixed production. Computational intelligence and modeling are the indispensable aids in the optimization techniques for taking into account the operational alternatives subjective preferences uncertainty.

The dynamics of the desired optimal air transport technologies resources recombination selection process as well as some more developed models implementing operational alternatives subjective preferences entropy are proposed to be investigated in the further studies.

References

- [1] M. J. Kroes, W. A. Watkins, F. Delp, R. Sterkenburg, *Aircraft Maintenance and Repair*, 7th. ed., McGraw-Hill, Education, New York, NY, 2013.
- [2] T. W. Wild, M. J. Kroes, *Aircraft Powerplants*, 8th. ed., McGraw-Hill, Education, New York, NY, 2014.
- [3] B. S. Dhillon, *Maintainability, Maintenance, and Reliability for Engineers*, Taylor & Francis Group, New York, NY, 2006.
- [4] D. J. Smith, *Reliability, Maintainability and Risk. Practical Methods for Engineers*, Elsevier, London, 2005.
- [5] R. D. Luce, D. H. Krantz, Conditional expected utility, *Econometrica* 39 (1971) 253–271.
- [6] R. D. Luce, *Individual Choice Behavior: A theoretical analysis*, Dover Publications, Mineola, NY, 2014.
- [7] E. T. Jaynes, Information theory and statistical mechanics, *Physical review* 106 (4) (1957) 620–630. doi:10.1103/PhysRev.106.620.
- [8] E. T. Jaynes, Information theory and statistical mechanics. II, *Physical review* 108 (2) (1957) 171–190. doi:10.1103/PhysRev.108.171.
- [9] E. T. Jaynes, On the rationale of maximum-entropy methods, *Proceedings of the IEEE* 70 (1982) 939–952. <https://ieeexplore.ieee.org/document/1456693>. doi:10.1109/PROC.1982.12425
- [10] F. C. Ma, P. H. Lv, M. Ye, Study on global science and social science entropy research trend, in: *Proceedings of the IEEE International Conference on Advanced Computational Intelligence, ICACI, Nanjing, Jiangsu, China, 2012*, pp. 238–242. <https://ieeexplore.ieee.org/document/6463159>. doi:10.1109/ICACI.2012.6463159
- [11] E. Silberberg, W. Suen, *The Structure of Economics. A Mathematical Analysis*, McGraw-Hill Higher Education, New York, NY, 2001.
- [12] V. Kasianov, *Subjective Entropy of Preferences. Subjective Analysis*, Institute of Aviation Scientific Publications, Warsaw, Poland, 2013.
- [13] O. Solomentsev, M. Zaliskyi, T. Herasymenko, O. Kozhokhina, Yu. Petrova, Data processing in case of radio equipment reliability parameters monitoring, in: *Proceedings of the International Conference on Advances in Wireless and Optical Communications, RTUWO, Riga, Latvia, 2018*, pp. 219–222. <https://ieeexplore.ieee.org/abstract/document/8587882>. doi:10.1109/RTUWO.2018.8587882.
- [14] A. M. B Pavani et al., "VISIR+ Project Follow-up after four years: Educational and research impact," 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA, 2023, pp. 1–8, doi: 10.1109/FIE58773.2023.10343298.
- [15] D. Shevchuk, O. Yakushenko, L. Pomytkina, D. Medynskyi, Y. Shevchenko, Neural network model for predicting the performance of a transport task, *Lecture Notes in Civil Engineering*. 130 LNCE (2021) 271–278. doi:10.1007/978-981-33-6208-6_27.
- [16] S. Subbotin, The neuro-fuzzy network synthesis and simplification on precedents in problems of diagnosis and pattern recognition, *Opt. Mem. Neural Networks* 22 (2013) 97–103. <https://doi.org/10.3103/S1060992X13020082>.
- [17] A. V. Goncharenko, Multi-optional hybridization for UAV maintenance purposes, in: *Proceedings of the IEEE International Conference on Actual Problems of UAV Developments, APUAVD, Kyiv, Ukraine, 2019*, pp. 48–51.

<https://ieeexplore.ieee.org/abstract/document/8943902>.
doi:10.1109/APUAVD47061.2019.8943902.

- [18] A. V. Goncharenko, Relative pseudo-entropy functions and variation model theoretically adjusted to an activity splitting, in: Proceedings of the IEEE International Conference on Advanced Computer Information Technologies, ACIT'2019, Ceske Budejovice, Czech Republic, 2019, pp. 52–55. <https://ieeexplore.ieee.org/abstract/document/8779876>. doi:10.1109/ACITT.2019.8779876.
- [19] A. Goncharenko, Development of a theoretical approach to the conditional optimization of aircraft maintenance preference uncertainty, Aviation 22 (2) (2018) 40–44. <https://journals.vilniustech.lt/index.php/Aviation/article/view/5929>. <https://doi.org/10.3846/aviation.2018.5929>.
- [20] A. V. Goncharenko, Airworthiness support measures analogy to the prospective roundabouts alternatives: theoretical aspects, Journal of Advanced Transportation Article ID 9370597 2018 (2018) 1–7. <https://www.hindawi.com/journals/jat/2018/9370597/>. <https://doi.org/10.1155/2018/9370597>.

Evaluation of the effectiveness of machine learning methods for detecting disinformation in Ukrainian text data

Khrystyna Lipianina-Honcharenko¹, Mariana Soia¹, Khrystyna Yurkiv¹, Andrii Ivasechko¹.

¹ West Ukrainian National University, Lvivska str., 11, Ternopil, 46000, Ukraine

Abstract

In today's world, where information spreads with unprecedented speed, disinformation poses a serious challenge to public trust and information security. The full-scale invasion of Ukraine by Russia in 2022 activated the use of disinformation as a tool of hybrid warfare, highlighting the need for effective methods of identification and control. This article focuses on evaluating the effectiveness of various machine learning methods for detecting disinformation in Ukrainian text data, using a dataset that includes news headlines collected during the conflict. The study encompasses the analysis of logistic regression, support vector machines (SVM), random forest, gradient boosting, KNN, decision trees, XGBoost, and AdaBoost. Model evaluation was performed using standard metrics: precision, recall, F1-score, overall accuracy, and confusion matrix. The results indicate significant potential for using machine learning in the fight against disinformation, particularly the random forest model demonstrated the highest effectiveness. The study emphasizes the importance of adapting and optimizing classifiers for the specific task of disinformation analysis, paving the way for further research in this field.

Keywords

Disinformation, machine learning, classification, text data.

1. Introduction

In the modern information space, a vast amount of data is generated daily, a significant portion of which is news content. In the context of increasing globalization and accessibility of information technologies, information spreads rapidly through the network, making it a powerful tool for influencing public opinion. However, this also paves the way for the mass dissemination of disinformation, which can have significant consequences for society, politics, and international relations. Navigating this flow of information and distinguishing reliable data from false has become an increasingly important task.

The war that began with Russia's full-scale invasion of Ukraine in February 2022 is a striking example of the use of disinformation as a weapon in hybrid warfare. This has created a need for the development of effective tools for analyzing and classifying informational content, with the goal of identifying and counteracting disinformation.

In this context, machine learning and natural language processing methods play a key role in the detection and analysis of fake news. The application of these technologies allows for the automation of the disinformation detection process, providing fast and efficient processing of large volumes of data. At the same time, the development of effective machine learning models for information classification requires a deep understanding of data specifics, preprocessing methods, and model optimization.

This article is devoted to the analysis of a dataset containing news headlines collected during the Russo-Ukrainian conflict, aimed at identifying disinformation. It considers the application of various machine learning methods, including logistic regression, support vector machines (SVM), random forest, gradient boosting, KNN, decision trees, XGBoost, and AdaBoost for the classification of text data. The concluding section is dedicated to evaluating the effectiveness of these models using standard evaluation metrics, including precision, recall, F1-score, overall

Proceedings Acronym: Proceedings Name, Month XX-XX, YYYY, City, Country

✉ xrustya.com@gmail.com (K. Lipianina-Honcharenko); mariankasoa@gmail.com (M. Soia);

kh.yurkiv@wunu.edu.ua (K. Yurkiv); Andrewivasechko@gmail.com (A. Ivasechko).

ORCID 0000-0002-2441-6292 (K. Lipianina-Honcharenko); 0009-0001-3719-6460 (M. Soia); 0009-0007-4917-3251 (K. Yurkiv); 0009-0002-8623-7828 (A. Ivasechko).



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

accuracy, and the confusion matrix, which allows determining the most effective methods for combating disinformation in the context of information warfare.

2. Related Work

In contemporary research in the field of information security and media content analysis, the importance of detecting and analyzing disinformation, especially anti-vaccine content on social media platforms such as Twitter, has gained particular relevance. In [1], language-neutral models are developed for detecting such content on a large scale, using multifaceted representations of messages in networks. Meanwhile, [2] focuses on the challenges associated with pre-training graph neural networks for context-oriented detection of fake news, pointing out strategic and resource constraints. In [3], a comparative analysis of supervised and unsupervised machine learning algorithms for detecting fake news is conducted, demonstrating their performance, efficiency, and robustness.

Research covering the analysis of disinformation and public opinion on the Russo-Ukrainian War employs a variety of methodologies, including sentiment analysis, creation and analysis of datasets, and studies of the impact of war on language choice. One study models and clusters sentiment trends of different countries regarding the war [4], while another offers a detailed dataset of tweets related to the crisis [5]. The discourse on Twitter about the war is also analyzed, with a particular focus on language and the geographical origin of tweets [6]. Another study focuses on the challenges of labeling sensitive content and its psychological impact on annotators [7]. It is also examined how the war affects the language choice of Ukrainians on Twitter, analyzing changes in language preferences over time [8]. A separate study provides insights into the activity on subreddits related to the conflict, analyzing post volumes, comments, and the level of engagement [9]. Research [10] demonstrates that the application of the BERT model for fake news detection achieves an accuracy of 79.88% and an area under the ROC curve of 0.87, highlighting its potential in combating disinformation on social networks.

As confirmed by the aforementioned analysis, research in the field of disinformation detection, particularly fake news, is a significant direction in the context of information security and media content analysis. The need for the development and application of advanced technological solutions for effective fake news detection is particularly compelling. However, there is a noticeable lack of research focused on the analysis of disinformation in the Ukrainian language, which poses a challenge to the scientific community to expand the linguistic spectrum of research in this field. Considering this, the aim of this study is to develop intelligent methods for detecting disinformation, specifically fake news, with an emphasis on Ukrainian-language content. This will enhance the level of information security and ensure the integrity of the news space in the conditions of the modern information society.

3. Research methodology

3.1 Dataset Description

For data collection and processing, the dataset (Ukrainian language) [11] was used, which contains approximately 10,700 news headlines about the Russo-Ukrainian War, collected from February 24 to December 11, 2022, covering the period from the beginning of the full-scale invasion.

The dataset for the analysis of disinformation in the Ukrainian media space consists of two main files: "data_set_4.csv" (Table 1) and "news_data.csv" (Table 2), each containing news headlines classified as true ("True") or false ("False"). The "data_set_4.csv" file records 8,237 true and 2,498 false news items, while "news_data.csv" contains a significantly larger number of true news items — 48,006, compared to 2,024 false, reflecting a wide range of informational content collected for the study of the dissemination of disinformation during the Russo-Ukrainian War.

Both datasets (see Table 1,2) are used for the analysis of disinformation in the Ukrainian media space and include data that were collected from official and unofficial sources with the purpose of studying the spread of fake news in the context of the Russo-Ukrainian War.

Table 1
Structure of data_set_4.csv

Field	Description	Data Type
Text	News Headline	Text
Label	Label that marks the news as true ("True") or false ("False")	Boolean
Link	Link to the news source (available only in this file)	Text

Table 2
Structure of news_data.csv

Field	Description	Data Type
Text	News Headline	Text
Label	Classification of the news as true ("True") or false ("False")	Boolean

The graphs (Fig.1) of label distribution show that in both datasets, the number of true messages predominates over the false ones. For example, in the first dataset, the ratio of true messages to false is high, which indicates a focus on reliable information.

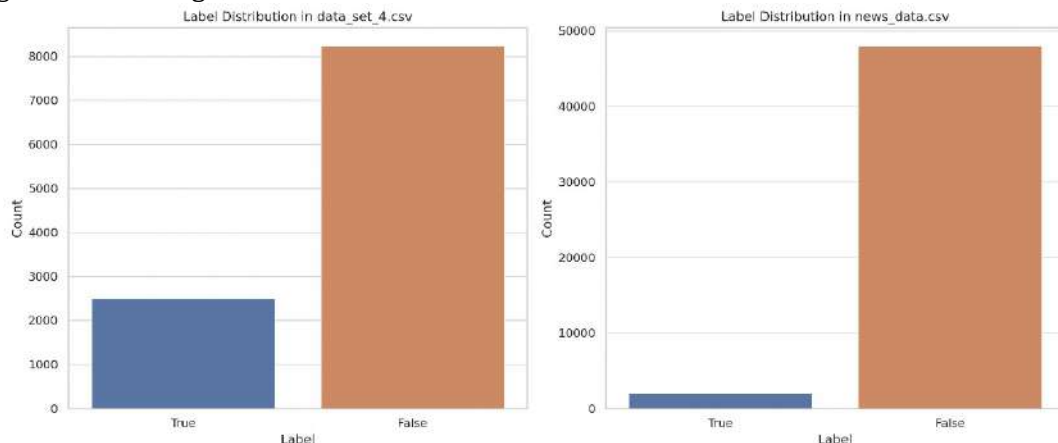


Figure 1: Label Distribution

The analysis of the most frequently used words (Fig.2) in the first dataset revealed words that appear hundreds of times. This allows identifying key topics of discussions or news, for example, the word "Ukraine" may occur most frequently, emphasizing the geographical or political focus of the collected data.

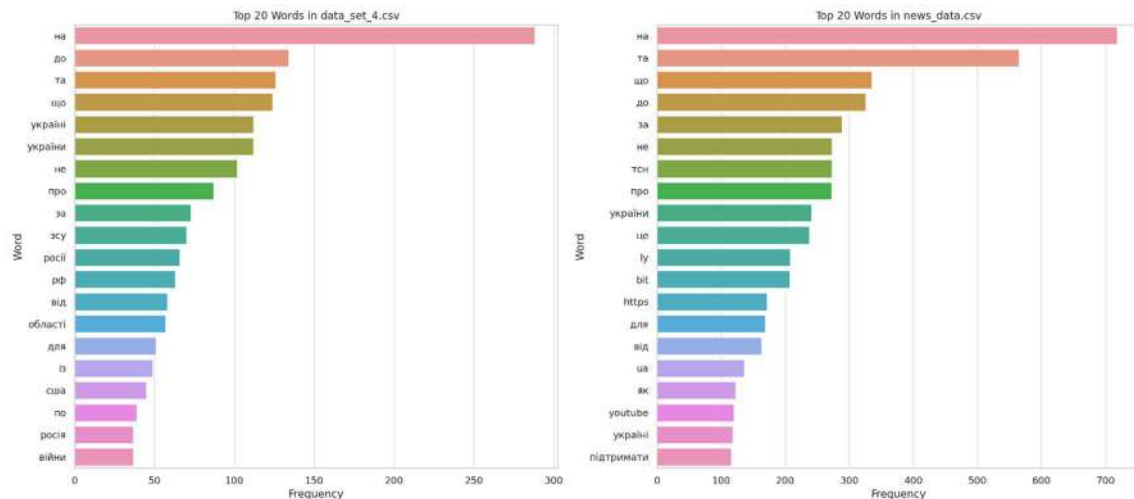


Figure 2: Top 20 Words

Most texts (Fig.3) in the first dataset have a length of 100 to 500 characters. Such information helps to understand the average volume of messages, which may indicate a prevalence of short news or overviews.

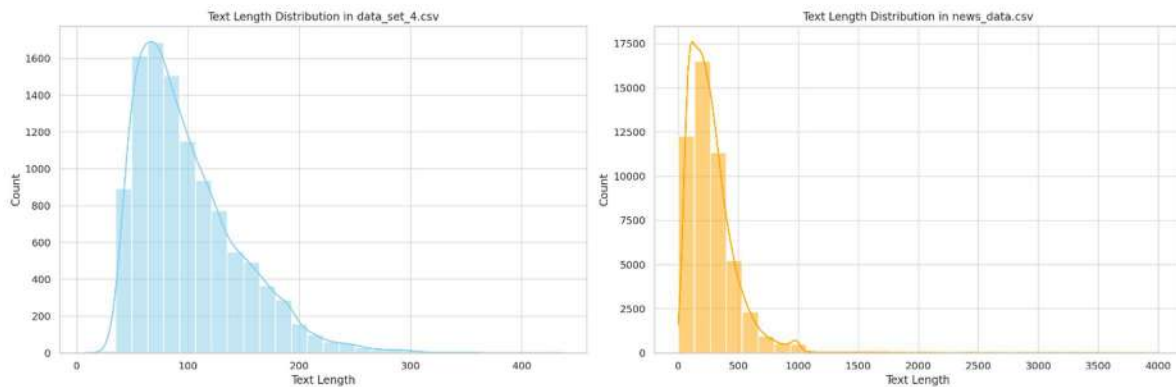


Figure 3: Distribution of Text Lengths

Analysis of the presented datasets covering headlines of news about the Russian-Ukrainian war demonstrates a significant advantage of credible information compared to false, emphasizing the focus on quality content. Considering that the dataset "news_data.csv" contains substantially more records compared to "data_set_4.csv", it is logical to use the former for training machine learning models as it provides a broader range of information and a greater number of examples for training. Meanwhile, the smaller dataset "data_set_4.csv" can serve as an excellent set for testing and evaluating the effectiveness of models on a smaller but specific data sample. This will allow assessing the model's ability to generalize learning on new, previously unknown data, emphasizing its practical value in real-world disinformation analysis conditions.

3.2 Description of used classifiers

In modern data analysis, especially in the context of detecting misinformation, the use of machine learning algorithms for classifying textual data becomes a key tool for developers and analysts. The diversity of classification methods [12], such as logistic regression, support vector machines (SVM), random forest, gradient boosting, k-nearest neighbors (KNN), decision tree, XGBoost, and AdaBoost, provides a wide range of approaches for data analysis and classification. Each of these algorithms has its unique advantages and limitations, making them more or less suitable for specific types of data and analytical tasks. The main goal is to select the optimal classifier that best fits the specificity of the task and the data we are working with.

Logistic regression [13] is a classification method used to predict the probability of two possible outcomes based on one or more independent variables. It transforms the linear combination of input data into probability using the logistic function. The advantages of logistic regression include simplicity in interpreting results, but it may be limited when analyzing complex relationships between variables and requires an assumption of linearity in relationships. Mathematically, logistic regression models the probability $P(Y=1)$ as a function of X , where Y is the dependent variable, and X is the set of independent variables. The probability is described by the equation:

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k)}}, \quad (1)$$

where e is the base of the natural logarithm, β_0 is the constant term (intercept), and $\beta_1, \dots, \beta_k$ are the coefficients of the independent variables. This equation allows us to estimate the probability that an observation belongs to class 1, depending on the values of the independent variables.

Support Vector Machine (SVM) algorithm [14] finds a hyperplane in a multidimensional space that best separates different data classes, maximizing the distance between the closest data points (support vectors) of different classes. The advantages of SVM include high accuracy in classification tasks, especially on relatively small datasets, and flexibility through the use of various kernel functions. However, its drawbacks include high computational resource

requirements for large datasets and complexity in interpreting the model. Mathematically, SVM seeks to solve the optimization problem: minimize $\frac{1}{2} ||w||^2$ subject to the constraints that

$$y_i (w \cdot x_i + b) \geq 1 \text{ to all } i, \quad (2)$$

where w is the weight vector of the hyperplane, b is the bias, x_i are the feature vectors, and y_i are the class labels.

The Random Forest algorithm [15] creates an ensemble of decision trees, training each tree on randomly selected subsets of the training dataset and features, which ensures high accuracy and model universality. The advantages of Random Forest include its ability to efficiently handle large datasets with high feature dimensionality and its lower tendency to overfit compared to individual decision trees. However, its drawbacks include relatively high computational resource requirements and the complexity of interpreting the model due to the large number of trees. The mathematical interpretation of Random Forest is based on the principle of "wisdom of the crowd," where the final model decision is determined by voting among the trees for classification tasks or averaging the outputs for regression tasks. More precisely, for classification:

$$Y = \text{mode}\{y_1, y_2, \dots, y_n\}, \quad (3)$$

and for regression:

$$Y = \frac{1}{n} \sum_{i=1}^n y_i, \quad (4)$$

where y_i is the prediction of each tree, and Y is the final prediction of the ensemble.

Gradient Boosting [16] is an ensemble machine learning method that improves predictions by sequentially training weak models, typically decision trees, to minimize a loss function. It is characterized by high prediction accuracy and flexibility in parameter tuning, but it can be prone to overfitting if not properly configured and requires more time and computational resources for training compared to other algorithms. Mathematically, Gradient Boosting performs optimization by adaptively reducing the difference between actual and predicted values using gradient descent, where the model update in the m -th iteration is defined as:

$$F_m(x) = F_{m-1}(x) + \alpha_m h_m(x), \quad (5)$$

where $F_{m-1}(x)$ is the prediction at the previous step, $h_m(x)$ is the weak classifier, and α_m is the learning rate.

The k -Nearest Neighbors (KNN) algorithm [17] classifies objects based on the nearest training examples in the feature space, where " k " indicates the number of neighbors considered to determine the class of a new object. The advantages of KNN include ease of implementation and its ability to effectively work with multi-class datasets. However, it requires significant computational resources to store training data and determine neighbors in large datasets, and it is sensitive to irrelevant features and data scaling. Mathematically, the classification of an object x in the KNN algorithm is determined by the majority vote of its neighbors, where each neighbor is weighted according to the inverse distance to x , typically using Euclidean distance:

$$d(x, x_i) = \sqrt{\sum_{j=1}^m (x_j - x_{ij})^2}, \quad (6)$$

where x is the point for classification, x_i is a point from the training dataset, and j varies from 1 to m , the number of features. The class of x is determined based on the most frequent class among the k nearest neighbors.

The Decision Tree algorithm [18] builds a predictive model in the form of a tree-like structure by partitioning the dataset into smaller subsets while simultaneously developing the associated decision tree. The advantages of decision trees include ease of interpretation, the ability to handle both numerical and categorical data, and no requirement for data normalization. However, decision trees are prone to overfitting, especially with deep trees, and can be unstable, meaning small changes in data can result in significantly different decision trees. Mathematically, decision trees use the concept of information gain or reduction in uncertainty (entropy) to select the attribute that best splits the dataset into subsets according to the target variable. The information gain for attribute A is calculated as:

$$IG(T, A) = Entropy(T) - \sum_{v \in \text{Values}(A)} \frac{|T_v|}{|T|} Entropy(T_v), \quad (7)$$

where T is the training set, $\text{Values}(A)$ is the set of all possible values of attribute A , T_v is the subset of T for which attribute A has the value v , and $Entropy(T)$ is the entropy of the training set T .

XGBoost (eXtreme Gradient Boosting) [19] is a highly efficient implementation of the gradient boosting algorithm, which optimizes both linear models and decision trees. The algorithm stands out for its high execution speed, ability to efficiently scale to large datasets, and built-in support for regularization, helping to mitigate overfitting. However, XGBoost can be challenging to tune due to the large number of hyperparameters and requires more time for training compared to simpler models. Mathematically, XGBoost minimizes losses using gradient descent, where the objective function includes both the loss function L and a penalty for model complexity Ω , adding regularization:

$$Obj = \sum_i L(y_i, \hat{y}_i) + \sum_k \Omega(f_k), \quad (8)$$

where y_i are the true labels, $\hat{y}_i$ are the predicted labels, f_k are the functions representing individual trees, and $\Omega(f_k)$ includes terms for regularization, such as the number of leaves and the sum of squares of node weights, thereby reducing the risk of overfitting.

AdaBoost (Adaptive Boosting) [20] is a machine learning algorithm that combines multiple weak classifiers to create a strong classifier, using an iterative approach to correct the errors of previous classifiers by assigning greater weight to observations that are harder to classify. The advantage of AdaBoost is its ability to improve prediction accuracy, ease of implementation, and automatic correction of underperforming classifiers. However, it can be prone to overfitting in the presence of outliers or highly noisy data and requires careful tuning of the number of iterations. Mathematically, AdaBoost adapts the weights of training observations, w_i , by increasing the weights of incorrectly classified observations. At each iteration step t , a classifier h_t is selected to minimize the weighted sum of errors. The final classifier is determined as a weighted sum of these classifiers.

$$H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x)), \quad (9)$$

where α_t is the weight assigned to classifier h_t , which depends on its accuracy.

Conclusions drawn from the description of the used classifiers underscore the importance of adapting the model to the specificity of the dataset and the analytical task. The effectiveness of each method depends on the size and quality of the data, the complexity of relationships in the dataset, as well as specific requirements for accuracy and interpretation of results. In the context of disinformation analysis, the choice between the simplicity of interpreting logistic regression and the high accuracy but complexity of tuning XGBoost or SVM may determine the success or failure in identifying fake news. Thus, careful selection and tuning of classifiers are critically important for developing effective tools to combat disinformation in the context of the Russian-Ukrainian war.

3.3 Evaluation metrics

For evaluating the effectiveness of models in classification tasks, key metrics such as precision, recall, F1-score, accuracy, and confusion matrix are utilized [21]. These metrics allow for a deeper analysis of the model's performance, identifying potential weaknesses, and optimizing the model for better results.

Precision is defined as the ratio of the number of true positive results to the total number of results classified as positive by the model. Mathematically, it is represented as:

$$P = \frac{TP}{TP+FP}, \quad (10)$$

where TP — true positives, and FP — false positives.

Recall measures the model's ability to identify all actual positive cases in the dataset. It is defined as the ratio of the number of correctly identified positive results to the sum of correctly identified positive results and instances that are actually positive but were missed by the model. The formula for calculation is:

$$R = \frac{TP}{TP+FN}, \quad (11)$$

where FN — false negatives.

The F1 Score is the harmonic mean between precision and recall, providing a balance between these two metrics. This is particularly useful in situations where class imbalances may cause biases in one metric over the other. F1 is defined as:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}. \quad (12)$$

Accuracy measures the percentage of cases correctly classified by the model and is defined by the formula:

$$A = \frac{TP+TN}{TP+TN+FP+F} , \quad (13)$$

where TN — true negatives.

The Confusion Matrix provides a visualization of classification results by representing the counts of TP, TN, FP, and FN in the form of a matrix. This allows us not only to determine the accuracy of the model but also to understand the types of errors made by the model.

3.4 Model training

The training procedure (Figure 4) on the training dataset is a fundamental step in the development of effective machine learning algorithms for text data classification. In this context, the use of datasets "news_data.csv" and "data_set_4.csv" for training and testing models, respectively, provides a valuable foundation for misinformation analysis. The initialization of the procedure begins with the import and preprocessing of data, including the removal of records without textual content, ensuring data cleanliness for subsequent processing stages.

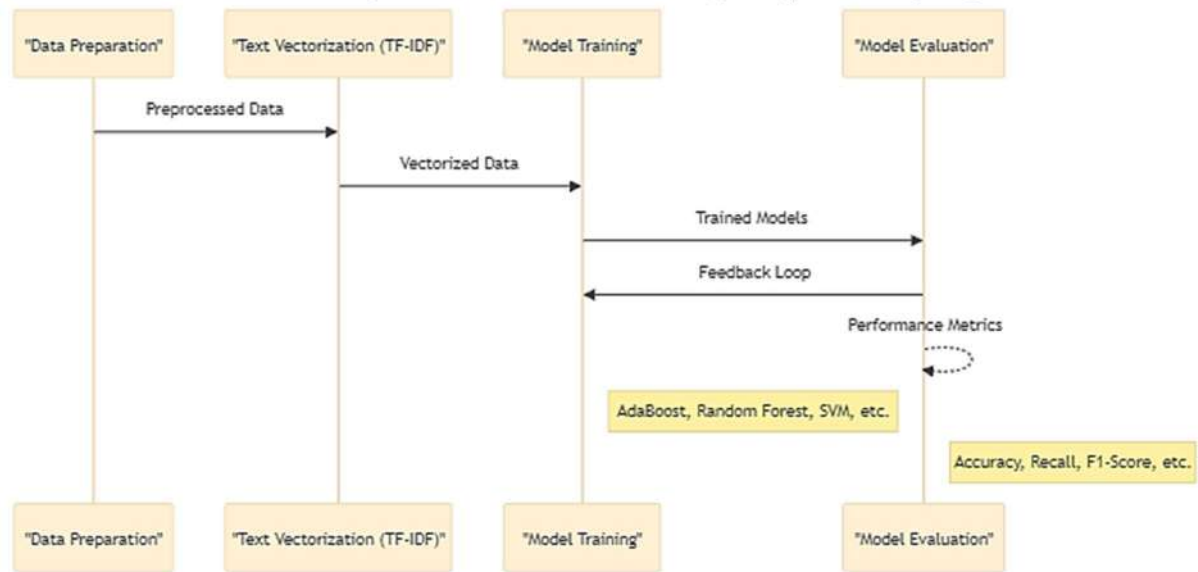


Figure 4: Model Training Process

Text vectorization using TF-IDF (Term Frequency-Inverse Document Frequency) is a key step in data preparation, as it transforms textual data into a numerical format, making them suitable for processing by machine learning models. This method considers not only the frequency of words in the text but also their uniqueness through the inverse frequency of documents, allowing the model to better identify important features in the text.

After preparing the vectorized training and testing data, the next stage involves training different models on the training dataset. This process requires the application of machine learning algorithms to the training dataset to form a model capable of effectively classifying text based on learned features. Each model adjusts its parameters to minimize errors on the training set while simultaneously ensuring the ability to generalize to new data.

Evaluation of the effectiveness of each trained model on the test dataset is crucial for determining its suitability and efficiency in classification tasks. The use of evaluation metrics such as accuracy, recall, F1-score, and overall accuracy allows for deep analysis and comparison of model performance. The evaluation results can be visualized for better understanding of model effectiveness, and the best-performing model can be selected for further use or saved for future analysis.

Thus, the model training procedure on the training dataset using pre-processed and vectorized text data is fundamental for developing reliable machine learning tools capable of effectively classifying and analyzing text for misinformation.

4. Research results

Further, we will discuss the implementation and evaluation of the effectiveness of various machine learning models in the task of classifying misinformation data in the Ukrainian media space, contextualized in the context of Russia's full-scale intervention. By analyzing a wide range of algorithms, we aim to identify the most effective methods for accurate detection and differentiation of misinformation incidents. This analysis is based on a comprehensive comparison of overall classification accuracy as well as specific metrics such as precision, recall, and F1-score for each class. The implementation was done in the Python programming language utilizing libraries for text analysis.

The logistic regression model showed (Figure 5) an overall classification accuracy of 86.4% on the test sample of 10,735 examples, demonstrating high effectiveness in identifying true values with an F1-score of 0.92 for the "True" class. However, the model was less accurate in identifying the "False" class, with a prediction accuracy of 0.96 and a low recall score of 0.44, indicating a significant number of false negatives, as reflected in the confusion matrix with 1,407 misclassified examples.

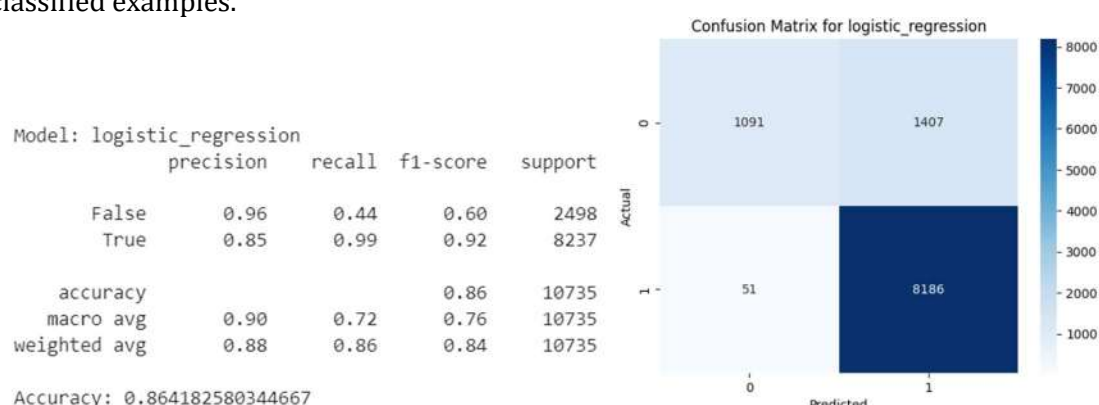


Figure 5: LogisticRegression Results

The SVM model demonstrated (Figure 6) high overall classification accuracy of 93.6% for the test sample, indicating its effectiveness in distinguishing between the "True" and "False" classes. Specific accuracy and recall metrics for the "False" class were 0.97 and 0.75, respectively, demonstrating the model's ability to identify negative cases well, albeit with some errors. At the same time, high metrics for the "True" class with precision of 0.93 and recall of 0.99 indicate minimal false negative classifications, confirmed by a low number of errors in the confusion matrix (618 for "False" and 68 for "True").

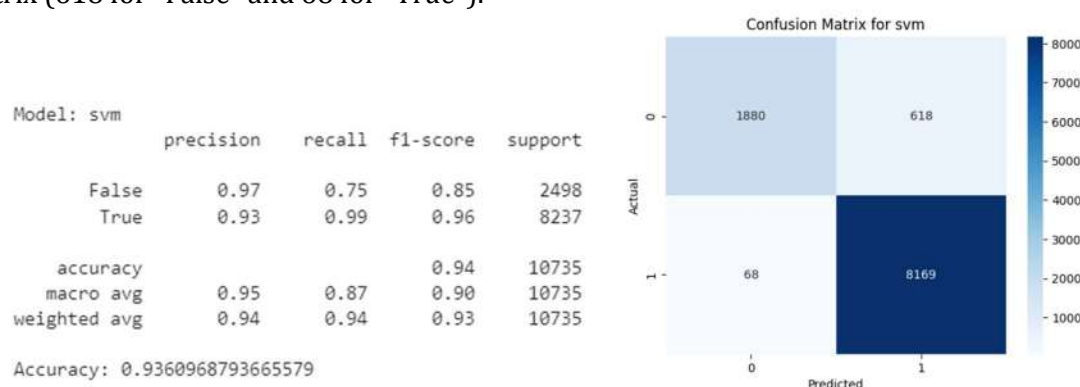


Figure 6: SVM Results

The Random Forest model demonstrated (Figure 7) high accuracy in classification with an overall accuracy of 95.3% on the test dataset, indicating the model's high effectiveness in recognizing both classes. For the "False" class, the model showed high accuracy (0.98) and a relatively high recall of 0.81, indicating the model's ability to effectively identify negative cases with a moderate number of errors. Conversely, extremely high accuracy (0.95) and almost perfect

recall (1.00) for the "True" class highlight the minimal number of false negative results, corroborated by the low number of errors in the confusion matrix.

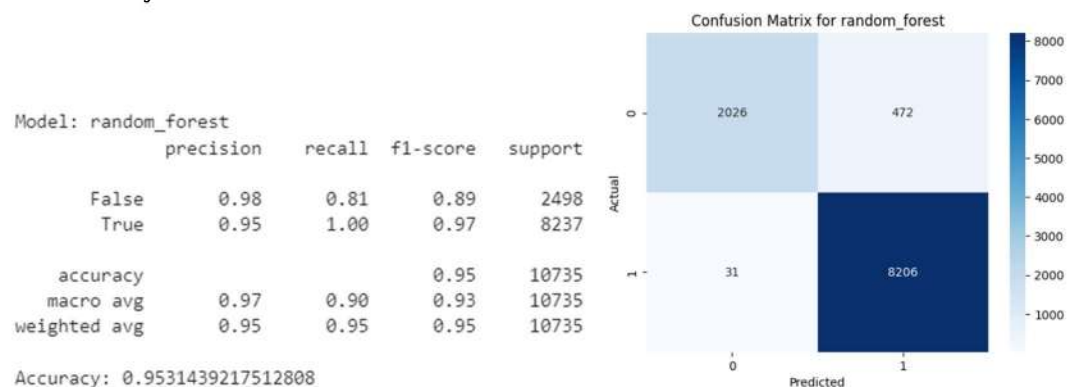


Figure 7: RandomForestClassifier Results

The Gradient Boosting model achieved (Figure 8) an overall classification accuracy of 87.3% on the test dataset, indicating the model's good ability to distinguish between the "True" and "False" classes. The model's precision for the "False" class was high (0.94), but the recall was only 0.48, indicating a significant number of type II errors, where negative cases are often misclassified as positive. Conversely, for the "True" class, the model showed impressive classification ability with a precision of 0.86 and a recall of 0.99, demonstrating its high effectiveness in identifying true positive cases with minimal errors.

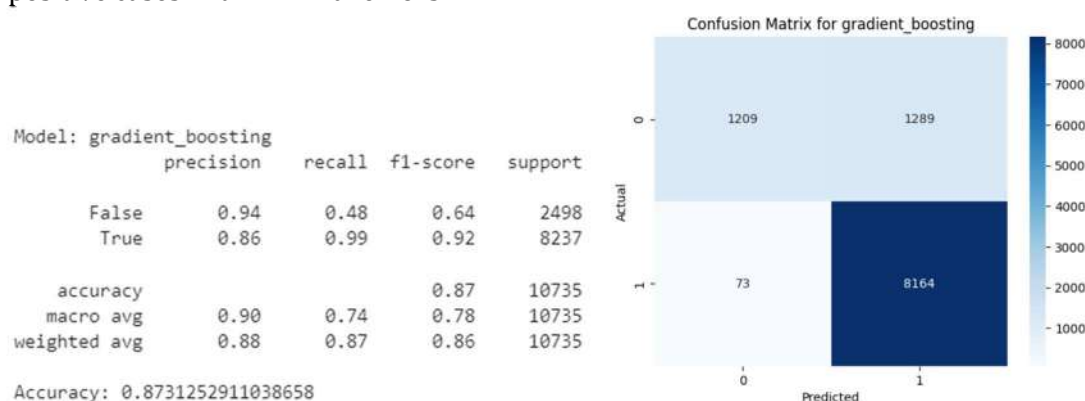


Figure 8: Gradient Boosting Results

The KNN (K-Nearest Neighbors) model demonstrated (Figure 9) an overall classification accuracy of 87.6% on the test dataset, highlighting its ability to effectively classify data. Although the model showed high precision (0.91) for the "False" class, the recall was only 0.51, indicating difficulties in identifying all negative cases. Conversely, for the "True" class, the model exhibited excellent precision (0.87) and a high recall (0.99), indicating its ability to identify positive cases with high confidence, albeit with some false positive errors, as seen in the confusion matrix.

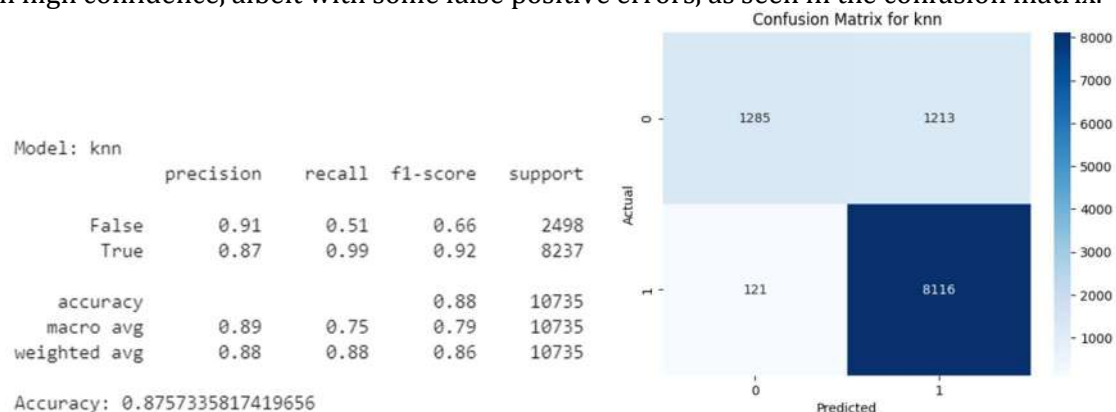


Figure 9: KNN Results

The Decision Tree model achieved (Figure 10) a high level of classification accuracy, 91.4%, on the test dataset, confirming its effectiveness in classifying data into "True" and "False" classes. For the "False" class, the model showed relatively high precision (0.81) and recall (0.83), indicating a balanced ability to identify negative cases with a relatively small number of errors. Meanwhile, for the "True" class, the model provided impressive precision (0.95) and recall (0.94), demonstrating its strong capabilities in identifying positive cases with minimal false negatives, as reflected in the confusion matrix.

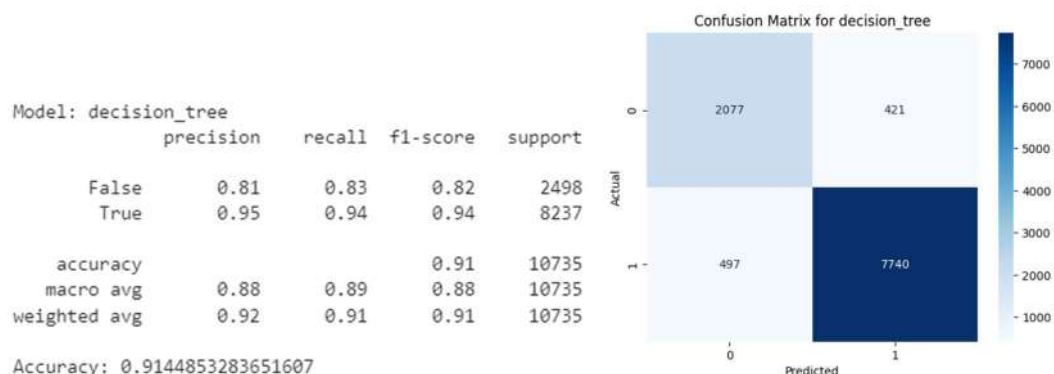


Figure 10: DecisionTreeClassifier Results

The XGBoost model demonstrated (Figure 11) an overall accuracy of 89.2% on the test dataset, indicating its ability to effectively classify the given dataset. The precision and recall metrics for the "False" class were 0.89 and 0.61, respectively, indicating the model's higher ability to correctly identify negative cases, albeit with some errors. Meanwhile, for the "True" class, the model showed high precision and recall (both 0.89 and 0.98), demonstrating excellent ability to accurately identify positive cases with minimal errors, as reflected in its high F1-score.

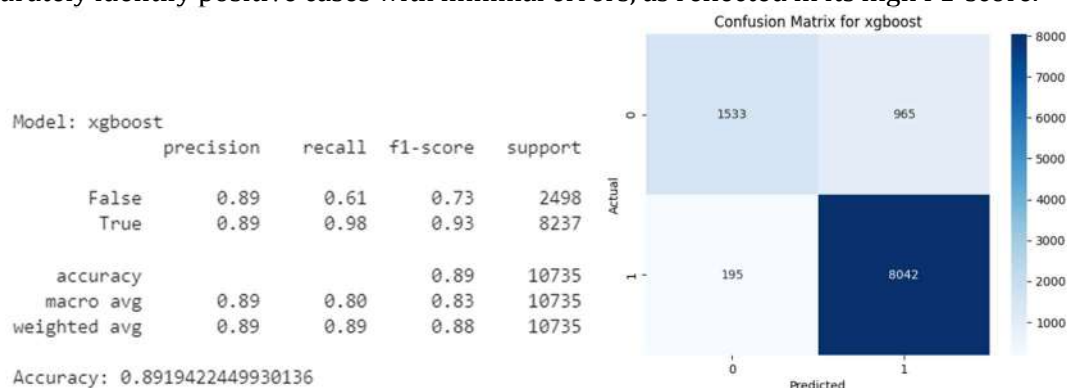


Figure 11: XGBoost Results

The AdaBoost model exhibited (Figure 12) an overall classification accuracy of 86.9% on the test dataset, indicating its effectiveness in recognizing data, albeit with some limitations. For the "False" class, the model achieved a precision of 0.88 with a recall of 0.50, indicating a relatively low ability to identify all negative cases. On the other hand, high precision (0.87) and recall (0.98) for the "True" class underscore the model's strong ability to detect positive cases, albeit with few errors, as reflected in the confusion matrix.

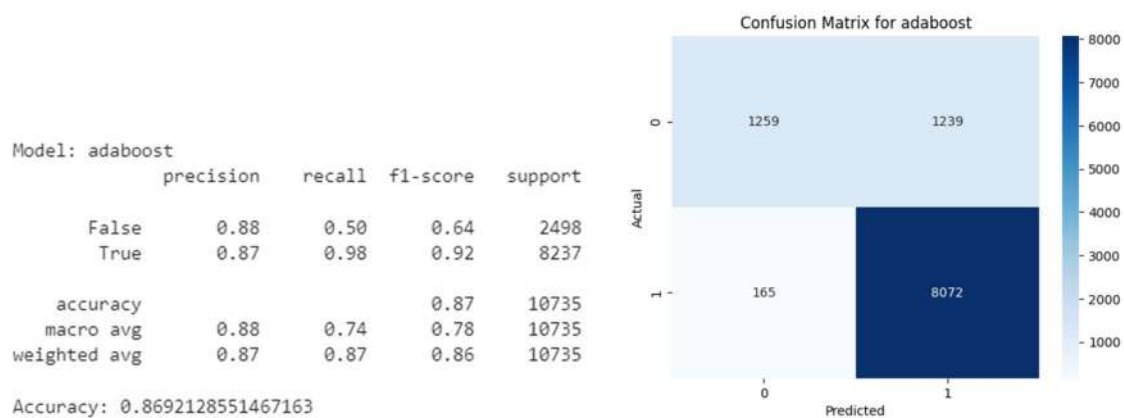


Figure 12: AdaBoostClassifier Results

Therefore, analyzing the results of applying various machine learning models to classify disinformation data in the Ukrainian media space after the onset of Russia's full-scale invasion, it can be noted that the Random Forest model proved to be the most effective with an accuracy of 95.3%, highlighting its ability to accurately detect and differentiate disinformation incidents. At the same time, models such as AdaBoost and logistic regression showed lower overall accuracy, which may indicate their limitations in identifying subtle cases of disinformation or more subjective aspects of information operations. This underscores the importance of choosing the appropriate model for the task of analyzing disinformation, where Random Forest may be more suitable for deep understanding and detecting complex patterns of disinformation in the context of information warfare.

5. Conclusion

The analysis of the effectiveness of various machine learning models applied to the task of classifying news headlines for misinformation in the Ukrainian media space demonstrates significant variations in accuracy, recall, and F1-score among the models. The considered algorithms, including logistic regression, SVM, random forest, gradient boosting, KNN, decision tree, XGBoost, and AdaBoost, showed different levels of performance in addressing the task.

The random forest model emerged as the most effective, achieving an overall accuracy of 95.3%, indicating its high capability to recognize and distinguish true and false messages. This is supported by high precision scores for both the "False" class (0.98) and the "True" class (0.95), as well as significant recall scores for both classes (0.81 for "False" and 1.00 for "True"), demonstrating its effectiveness in minimizing both false positives and false negatives.

These results underscore the importance of choosing the appropriate model for a specific misinformation analysis task. The model choice not only affects the overall classification accuracy but also the model's ability to minimize false positives or false negatives, which is crucial for developing effective tools to combat misinformation. Particularly, the random forest model, which demonstrated the best performance, can be recommended as the optimal choice for similar tasks, providing a high level of accuracy and the ability to effectively distinguish between true and false messages.

Further scientific research in the field of identification and analysis of misinformation in the Ukrainian media space requires deeper development and improvement of machine learning algorithms, with a particular focus on enhancing their ability to recognize subtle and complex forms of misinformation. The results of our study show that the random forest model, with an accuracy of 95.3%, proved to be the most effective, but there is potential for improvement, especially in accurately distinguishing between true and false messages. In the future, researchers may focus on developing hybrid models that combine the advantages of multiple algorithms, including deep learning and neural networks, to ensure greater adaptability and accuracy in different informational contexts. Additionally, an important direction will be the development of methods that allow models to better understand the semantic context and emotional tone of texts, which can significantly improve their ability to identify hidden

misinformation. Implementing such approaches will require not only technological innovations but also a deeper understanding of linguistic nuances and cultural-historical contexts on which misinformation is based.

References

- [1] Ashford, J.R. (2024). Detecting Anti-vaccine Content on Twitter using Multiple Message-Based Network Representations. arXiv preprint arXiv:2402.18335.
- [2] Donabauer, G., & Kruschwitz, U. (2024). Challenges in Pre-Training Graph Neural Networks for Context-Based Fake News Detection: An Evaluation of Current Strategies and Resource Limitations. arXiv preprint arXiv:2402.18179.
- [3] Kaur, S., & Ranjan, S. (2024). Comparative Analysis of Supervised and Unsupervised Machine Learning Algorithms for Fake News Detection: Performance, Efficiency, and Robustness. ResearchGate.
- [4] Vahdat-Nejad, H., Akbari, M. G., Salmani, F., Azizi, F., & Nili-Sani, H. R. (2023). Russia-Ukraine war: Modeling and Clustering the Sentiments Trends of Various Countries. arXiv preprint arXiv:2301.00604.
- [5] Haq, E. U., Tyson, G., Lee, L. H., Braud, T., & Hui, P. (2022). Twitter dataset for 2022 russo-ukrainian crisis. arXiv preprint arXiv:2203.02955.
- [6] Zia, H. B., Haq, E. U., Castro, I., Hui, P., & Tyson, G. (2023). An Analysis of Twitter Discourse on the War Between Russia and Ukraine. arXiv preprint arXiv:2306.11390.
- [7] Daria, S. (2023). When a Language Question Is at Stake. A Revisited Approach to Label Sensitive Content. arXiv preprint arXiv:2311.10514.
- [8] Racek, D., Davidson, B. I., Thurner, P. W., & Kauermann, G. (2023). The Politics of Language Choice: How the Russian-Ukrainian War Influences Ukrainians' Language Use on Twitter. arXiv preprint arXiv:2305.02770.
- [9] Zhu, Y., Haq, E. U., Lee, L. H., Tyson, G., & Hui, P. (2022). A reddit dataset for the russo-ukrainian conflict in 2022. arXiv preprint arXiv:2206.05107.
- [10] Padalko, H., Chomko, V., & Chumachenko, D. (2023). Misinformation Detection in Political News using BERT Model. Proceedings of the 3rd International Workshop of IT-professionals on Artificial Intelligence (ProFIT AI 2023) Waterloo, Canada, November 20-22, 2023. 117-127
- [11] Ukrainian news. (n.d.-a). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/zepopo/ukrainian-fake-and-true-news/data?select=news_data.csv
- [12] Lipyana, H., Maksymovych, V., Sachenko, A., Lendyuk, T., Fomenko, A., & Kit, I. (2020, August). Assessing the investment risk of virtual IT company based on machine learning. In International Conference on Data Stream Mining and Processing (pp. 167-187). Cham: Springer International Publishing.
- [13] Lipyana, H., Sachenko, S., Lendyuk, T., Brych, V., Yatskiv, V., & Osolinskiy, O. (2021, January). Method of detecting a fictitious company on the machine learning base. In International Conference on Computer Science, Engineering and Education Applications (pp. 138-146). Cham: Springer International Publishing.
- [14] Kalogridis, I. (2024). Robust and adaptive functional logistic regression. Computational Statistics & Data Analysis, 192, 107905.
- [15] Çakir, M., Yilmaz, M., Oral, M. A., Kazanci, H. Ö., & Oral, O. (2023). Accuracy assessment of RFerns, NB, SVM, and kNN machine learning classifiers in aquaculture. Journal of King Saud University-Science, 35(6), 102754.
- [16] Sun, Z., Wang, G., Li, P., Wang, H., Zhang, M., & Liang, X. (2024). An improved random forest based on the classification accuracy and correlation measurement of decision trees. Expert Systems with Applications, 237, 121549.
- [17] Wang, C., Xiao, W., & Liu, J. (2023). Developing an improved extreme gradient boosting model for predicting the international roughness index of rigid pavement. Construction and Building Materials, 408, 133523.

- [18] Çakir, M., Yilmaz, M., Oral, M. A., Kazanci, H. Ö., & Oral, O. (2023). Accuracy assessment of RFerns, NB, SVM, and kNN machine learning classifiers in aquaculture. *Journal of King Saud University-Science*, 35(6), 102754.
- [19] Gao, B., Zhou, Q., & Deng, Y. (2024). HIE-EDT: Hierarchical interval estimation-based evidential decision tree. *Pattern Recognition*, 146, 110040.
- [20] Niazkar, M., Menapace, A., Brentan, B., Piraei, R., Jimenez, D., Dhawan, P., & Righetti, M. (2024). Applications of XGBoost in water resources engineering: A systematic literature review (Dec 2018–May 2023). *Environmental Modelling & Software*, 105971.
- [21] Xing, H. J., Liu, W. T., & Wang, X. Z. (2024). Bounded exponential loss function based AdaBoost ensemble of OCSVMs. *Pattern Recognition*, 148, 110191.

System for Teaching Proper Toothbrushing Techniques using 6DOF Marker Pose Estimation and Machine Learning Methods

Dmytro Fedasyuk¹, Ostop Truba¹ and Tetyana Marusenkova¹

¹ Lviv Polytechnic National University, St. Bandery str, 28 a, Lviv, 79013, Ukraine

Abstract

A novel approach to synthesizing software systems for teaching toothbrushing techniques is proposed. This approach leverages augmented reality technology and machine learning methods to monitor toothbrush movements, recognize cleaning gestures, and analyze and evaluate the user's performance. This paper outlines the proposed approach, determines the optimal marker type, size, and brightness conditions required for precise positioning assessment, and highlights Kalman filtering for suppressing noise introduced by camera imperfections and swift toothbrush movements.

Keywords

Toothbrushing, pose estimation, fiducial markers, augmented reality, convolutional neural networks

1. Introduction

Diseases of the oral cavity represent one of the most significant health challenges for countries and populations worldwide. According to estimates by the World Health Organization (WHO) in 2019, oral cavity diseases affected nearly 3.5 billion people [1].

Among the primary oral cavity diseases are dental caries, gingivitis, oral cavity cancer, HIV infection, cleft lip and palate, and oral cavity and dental traumas. Additionally, research exists showing a correlation between deteriorating oral health and overall health conditions, which may be associated with diseases such as heart disease, endocarditis, and premature births [2].

Inadequate or improper oral hygiene ranks as one of the leading causes of the high prevalence and intensity of dental diseases. While addressing insufficient oral hygiene is relatively widespread and straightforward, identifying and rectifying improper hygiene practices can be challenging. Factors contributing to the development of incorrect oral hygiene habits include low levels of education, habit formation during childhood, and the confusing abundance and variety of tooth-cleaning recommendations [3, 4].

Therefore, the development of systems to educate individuals on proper oral hygiene practices, providing tools to cultivate correct habits in both adults and children, emerges as a prudent approach to prevent the mentioned oral cavity diseases. Furthermore, such systems serve as excellent solutions for promoting nationwide preventive measures.

2. Related work

Recently, the scientific community has increasingly focused on utilizing technological solutions for oral cavity health care. Solutions developed for teaching oral hygiene primarily employ the following approaches: wearable electronics (smartwatches and bracelets), the creation of smart brushes using MEMS sensors, and the use of augmented and virtual reality technologies.

In [5], the authors describe the development of smart toothbrushes for monitoring teeth cleaning effectiveness using a recurrent probabilistic neural network (RPNN). To address the problem, they propose using a modified toothbrush with an inertial measurement unit (IMU) to

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ dmytro.v.fedasyuk@lpnu.ua (D. Fedasyuk); ostop.truba.mnpzm.2022@lpnu.ua (O. Truba); tetiana.a.marusenkova@lpnu.ua (T. Marusenkova)

ORCID: 0000-0003-3552-7454 (D. Fedasyuk); 0009-0004-6177-623X (O. Truba); 0000-0003-4508-5725 (T. Marusenkova)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

determine its spatial position. The main part of this work is dedicated to proper processing and recognition of movements occurring during teeth cleaning, utilizing the RPNN model. The authors highlight the advantages of RPNN over convolutional neural networks (CNN) and long short-term memory networks (LSTM), such as low computational resource usage, high recognition accuracy, and efficiency.

In [6], the study aims to present a protocol for developing a serious game to motivate oral hygiene practice in children. Kinect hardware (from Microsoft) is employed to track human movements. The system requires users to perform specific tasks (moves), after which it evaluates their actions and provides recommendations for improving toothbrushing technique. The author bases the work on the Stillman and Fones toothbrushing techniques.

The research [7] addresses toothbrush monitoring using augmented reality technology. The authors use multiple AR markers attached to a dodecahedron base to collect positioning data. The work also discusses the correlation between the number of AR markers and the accuracy of results. It is recommended to use three markers to achieve 95% monitoring accuracy. The authors emphasize the usefulness of this method in developing oral hygiene training systems.

In [8], a smartwatch equipped with an accelerometer is proposed for monitoring movements. A feedforward neural network is used for gesture recognition, with a publicly available UCI repository selected as the dataset. The authors aim to detect human motion primitives using a triaxial accelerometer.

Motion tracking in [9] employs augmented reality technology. The software system attempts to solve two problems: monitoring toothbrush movements and identifying soiled tooth surfaces. OpenCV library is used for toothbrush movement tracking, with a single AR (ArUco) marker attached to the toothbrush for positioning data. The authors state that the soiled tooth surface detection accuracy is 98%.

In [10], a smartwatch equipped with a magnetic sensor and a modified toothbrush with tiny magnets attached is proposed for motion tracking. This setup allows for the transmission of user motion data to the watch. The system comprises two phases: a training phase and a working phase. During the training phase, the user must perform tooth cleaning several times using the Bass technique for calibration. Subsequently, in the working phase, the user continues to use the software, receiving feedback on the correctness of their technique. At the conclusion of the study, the author notes that the system usage significantly improves the tooth cleaning technique in respondents, resulting in more effective plaque removal. The accuracy of gesture monitoring with this approach is reported to be 85.6%.

After analyzing recent publications in this field, one can identify the following shortcomings:

- Use of IMU sensors based on MEMS technology: studies were conducted using sensors of high accuracy, which are impractical for commercial projects. The authors propose using MEMS technology due to its low cost. However, MEMS sensors have low accuracy [11, 12].
- Use of laboratory conditions: studies using computer vision for motion monitoring did not consider external conditions (e.g., lighting).
- Significant computational resource usage for real-time data processing, reducing system data accessibility.
- Some studies utilized no known toothbrushing techniques when evaluating process effectiveness, a crucial factor for developing educational systems.
- Proposed methods that require IMU sensors to work are cost-ineffective, which makes them inaccessible to most people.

The described problems hinder the creation of an effective and accessible training system for oral cavity hygiene. This research aims to address the mentioned problems.

3. System architecture overview

In general, the software system should function as follows. The user selects the toothbrushing technique they aim to learn. Subsequently, the system provides visual instructions for the individual to follow. The software continuously monitors user activity in real time through a webcam. It evaluates the user's actions and provides recommendations for enhancing their performance in the completed exercises. From this standpoint, several issues emerge that the prospective software system needs to tackle:

- Capturing video streams
- Recognizing human faces
- Determining the spatial position of a toothbrush at any given moment
- Identifying brush-cleaning areas
- Evaluating the user's proficiency in executing a specific tooth cleaning technique and offering guidance for improvement.

Hence, to address the mentioned challenges, we propose the system architecture shown in Figure 1. The system comprises five modules (and a GUI), each targeting the mentioned issues.

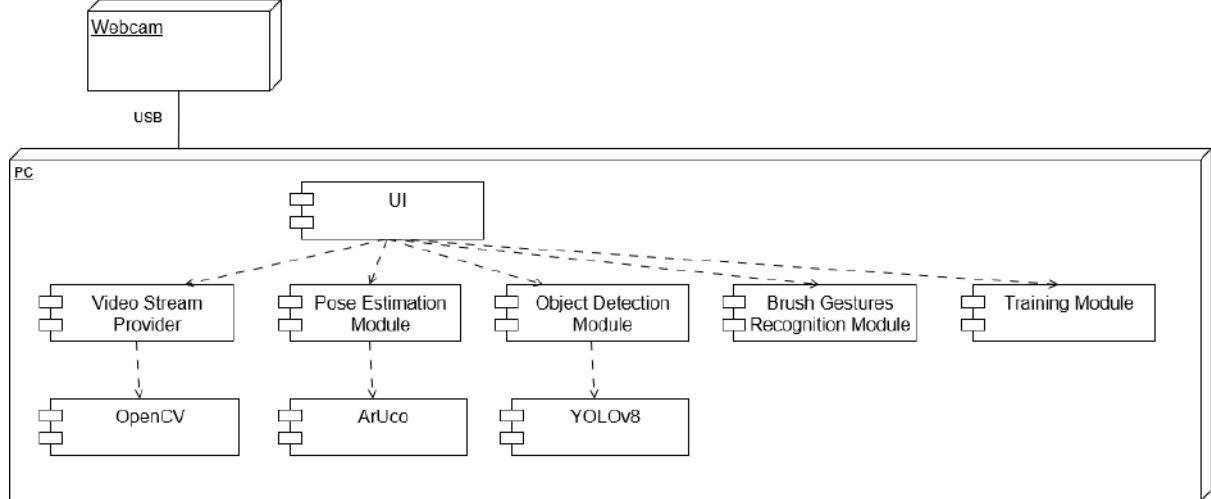


Figure 1: Deployment diagram of the proposed software system

Each module is designed to be interchangeable, facilitating the easy substitution of one implementation with another as needed (for instance, replacing one marker type with another). Furthermore, each module operates concurrently in a separate thread. It ensures that all computations can be executed in parallel, thus enhancing the overall system performance in terms of frames per second (FPS). Inter-module communication is facilitated through signals and slots. This mechanism is specific to QT and similar to the observer pattern. Data acquired from a module are transmitted to the main component (GUI) via signal emission. Subsequently, if necessary, the data is relayed to other components using the same principle.

The proposed architecture exhibits versatility and can be applied universally to similar application types, irrespective of the platform (Mobile, Web, PC). This adaptability underscores its potential utility across diverse technological environments.

4. Face detection

Recognition of the human face is primarily intended to check whether a person is visible and looking directly into the camera frame. It ensures that the system will not start/resume the training with no person present. It also guarantees that brush-cleaning areas are detected accordingly since the person looks directly into the camera.

For the object detection module implementation, we used a CNN specifically employing the YOLO (v8) single-stage detector [13]. We trained the model using an annotated open dataset comprising 1280 images of human faces (900 training and 380 validation). Each image had dimensions of 640x640 pixels. The number of epochs was 25.

5. Toothbrush pose estimation

5.1. Comparison of popular markers libraries

A fiducial marker system comprises planar (2D) markers positioned within a specific environment, intended to be detected by a camera across various applications. These markers enable the estimation of the object pose [7]. However, certain assumptions must be considered regarding the following factors:

- **Marker Placement:** Strategic positioning of fiducial markers within the environment is essential to maximize their visibility to the camera system while minimizing potential occlusions and obstructions. Also, it directly impacts marker pose estimation accuracy.
- **Lighting Control:** Environmental lighting conditions must be carefully controlled to minimize fluctuations that could deteriorate marker detection. Consistency in lighting setups is crucial for ensuring reliable performance.
- **Distance Consideration:** Variations in the distance between the camera and fiducial markers must be accounted for, as they can influence optimal marker size for the project as well as its detection.
- **Detection Speed:** In real-time application projects, the detection algorithm must work as fast as possible to ensure proper system response.

Given the metrics outlined above, an attempt can be made to evaluate and contrast existing marker libraries to identify the most suitable option for our project requirements. The selected candidates for comparison include ArUco, AprilTag, STag, CCTag, and ARTag (Figure 2).

Pose estimation and motion tracking efficacy highly depend on the marker type, size, tilt, the distance between the marker and camera, and marker occlusions. The smaller the marker size and the greater the mentioned distance, the slower marker detection. Besides, if the marker is partially occluded, it may be misrecognized. Not all the marker types provide the same results, i.e., the efficacy of pose estimation highly depends on the marker type. However, this issue is not covered in the literature. To choose the appropriate marker type, we designed a set of experiments with different marker sizes, distances from the camera, and percentages of occluded areas. The results are presented in Tables 1 – 5 (paragraphs 5.1.1 – 5.1.4).

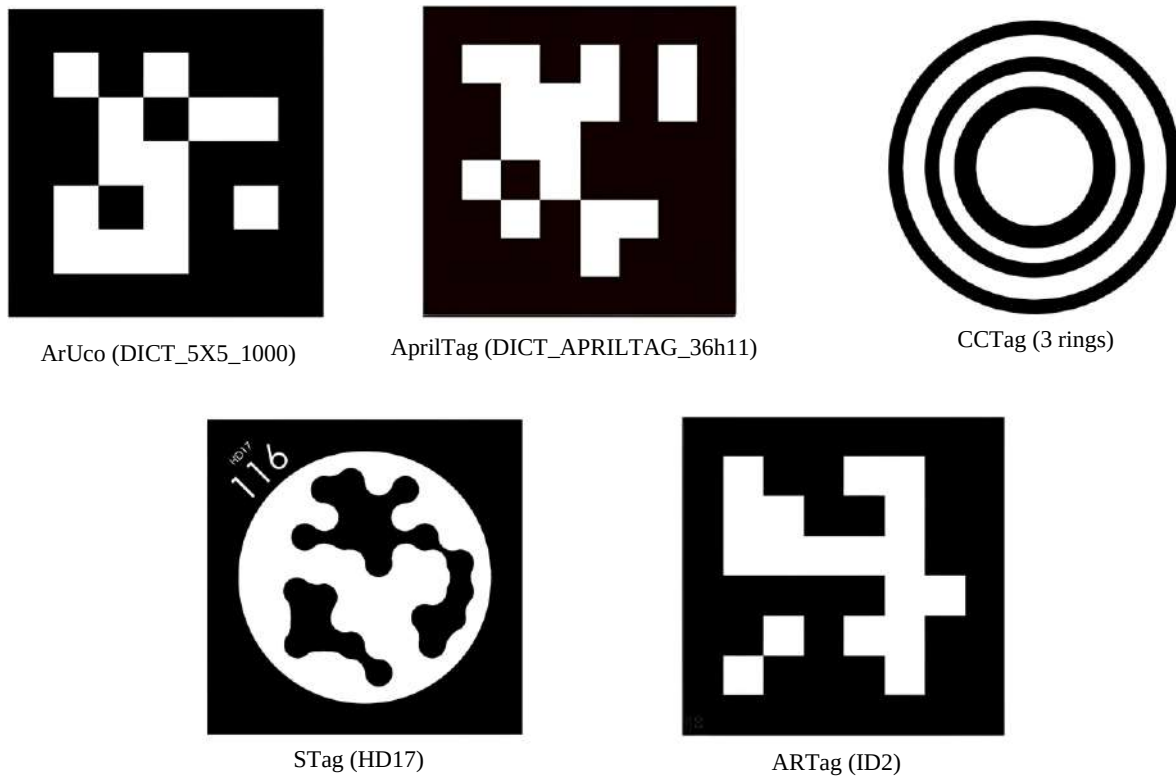


Figure 2: Fiducial markers used in our comparison with specified dictionaries

5.1.1. Marker size

Typically, the monitor should be within the range of distances from the eyes, specifically between the near (52 cm) and middle (73 cm) distances [14]. Therefore, it is imperative to determine the optimal marker size capable of detection within the range from 40 cm to 100 cm. The experimental setup involves placing the camera in a fixed position (XY) while the marker is positioned at various distances (with a 90-degree tilt; pitch rotation), including 40 cm, 60 cm, 80 cm, and 100 cm. Subsequently, the program executes 100 frames trying to detect the marker. We

performed ten experimental runs under the same lighting conditions and averaged the resulting data to derive the final marker detection rate. The results are resented in Tables 1 – 3.

Table 1

Comparing the marker detection rate with MSize = 10 mm

Marker/Distance	40 cm	60 cm	80 cm	100 cm
AprilTag	97%	69%	0	0
ArUco	98%	77%	0	0
STag	95%	75%	0	0
CCTag	0	0	0	0
ARTag	93%	38%	0	0

Table 2

Comparing the marker detection rate with MSize = 15 mm

Marker/Distance	40 cm	60 cm	80 cm	100 cm
AprilTag	100%	89%	62%	2%
ArUco	100%	95%	94%	7%
STag	100%	94%	84%	4%
CCTag	0	0	0	0
ARTag	100%	83%	42%	2%

Table 3

Comparing the marker detection rate with MSize = 20 mm

Marker/Distance	40 cm	60 cm	80 cm	100 cm
AprilTag	100%	100%	96%	93%
ArUco	100%	100%	100%	95%
STag	100%	100%	97%	95%
CCTag	3%	0	0	0
ARTag	100%	100%	92%	91%

5.1.2. Marker tilt angle

Marker tilt angle pertains to the degree of rotation or inclination of a fiducial marker relative to the camera's field of view. A tilted marker departs from its optimal alignment, potentially influencing its detectability and pose estimation accuracy. In the experimental configuration, the camera remains stationary (fixed in the XY plane) while tilted along different axes (pitch and yaw). Our findings indicate that most markers demonstrate satisfactory performance when tilted up to 65 degrees in both counter-clockwise and clockwise directions.

5.1.3. Marker detection speed

Several factors can impact the speed of marker detection, including the marker size, the detection algorithm efficiency, processing power, and lighting conditions. However, including such data in this paper may introduce bias given the variability introduced by these factors. Nonetheless, under stable and fixed conditions, such as consistent lighting, distance, and marker size, it would be beneficial to present speed detection data. In our experiments, we used the marker size of 16mm, positioned at a 40 cm distance. The webcam configuration is given in section 8. Table 4 shows the results.

Table 4

Comparing the marker detection speed

Marker	AprilTag	ArUco	STag	CCTag	ARTag
Time (s)	0.00425	0.00454	0.0143	0.2	0.00913

5.1.4. Marker occlusion

Marker occlusion refers to instances where a portion or the entirety of a fiducial marker is obstructed from the camera's view. Such obstruction can arise due to various factors, including

physical objects blocking the marker or the marker partially concealed behind another object. Some markers are specifically designed to exhibit greater resistance to occlusion.

To assess a marker's resistance to occlusion, we ran an experiment where a piece of paper covered a certain percentage (50%, 25%, 10%, 5%) of the total area of the marker. In this experiment, 20 mm markers were employed, and the camera was fixed along the Z-axis at a distance of 30 cm from the ground. The experiment findings are summarized in Table 5.

Table 5

Comparing the marker resistance to occlusion

Marker/ Occlusion	1% (4 mm)	5% (20 mm)	10% (40 mm)	25% (100 mm)	50% (200 mm)
AprilTag	Partial detection	0	0	0	0
ArUco	1	1	Partial detection	0	0
S-Tag	1	1	1	Partial detection	0
CCTag	1	1	1	1	1
ARTag	Partial detection	0	0	0	0

5.2. Markers placement

To ensure optimal accuracy and precision in pose estimation with markers, it is imperative to arrange them in a configuration where at least two markers are visible simultaneously. This precautionary measure is vital for several reasons:

- Pose estimation encounters challenges when only a single marker is visible within the frame, resulting in ambiguity in solving the Perspective-n-Point (PnP) problem [15]. This limitation arises from the insufficient information provided by a single marker, impacting the accuracy and reliability of the pose estimation.
- Single-marker pose estimation is constrained by limitations in accuracy and the range of viewing angles. Without multiple markers for reference, the system may struggle to determine accurately the pose of the object being tracked.
- When used for brushing the inner side of dental areas, a single marker is susceptible to occlusion, obstructing the marker's view and impeding accurate tracking.

Considering these factors, it is advisable to utilize 3D objects instead of plain markers, such as cubes, tetrahedrons, or octahedrons. However, numerous studies focusing on object position tracking using fiducial markers [7, 16, 17] advocate for the adoption of a dodecahedron-based design, which offers favorable properties for robust tracking and accurate pose estimation.

5.3. Finalizing toothbrush design

After completing all of our experiments, we have finalized the following configuration: an ArUco marker with a size of 15.8 mm coupled with a dodecahedron-based object featuring edges measuring 17 mm in length, produced using an off-the-shelf 3D printer. The ArUco marker was selected due to its superior detection rate and optimal performance compared to other fiducial markers. The marker size was determined through range testing considerations, aiming to strike a balance between detectability and user comfort, particularly regarding the toothbrush handle. Figure 3 depicts our modified toothbrush design. Figure 4 shows its usage in the developed application.



Figure 3: Our proposed modified toothbrush for 6DOF tracking

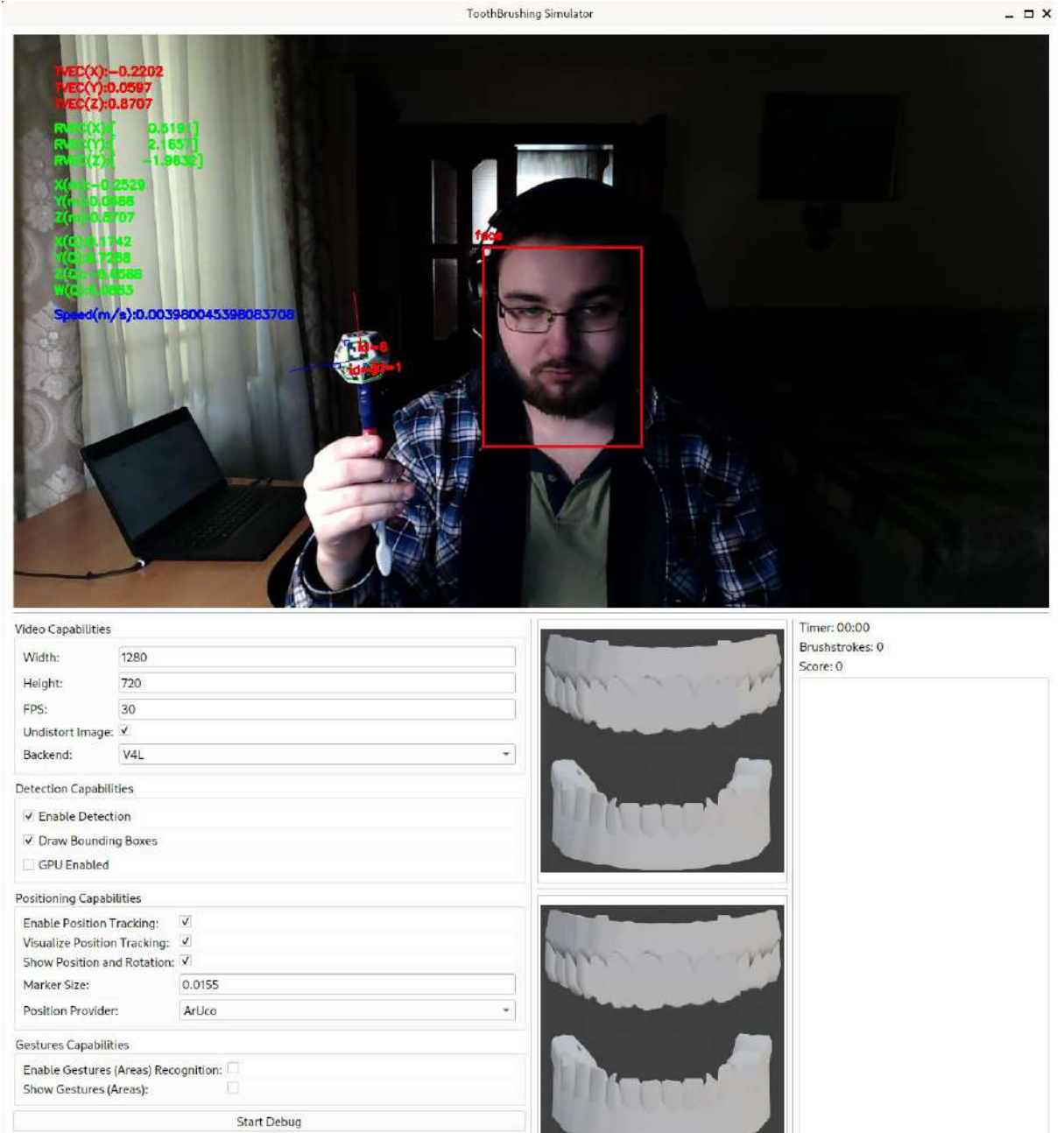


Figure 4: A sample of the marker usage in the developed application

5.4. Data filtration

Given the real-time nature of the system, characterized by swift movements of the marker and variations in camera quality, a considerable amount of noise is introduced within the captured frames. To somewhat mitigate this noise, we implemented a linear Kalman filter. This approach is expected to yield improved accuracy in tracking outcomes [18].

Initially, it is necessary to establish our state vector (1), comprising 18 states. These states encompass positional information (x, y, z) alongside their first and second derivatives (velocity and acceleration). Additionally, rotation is presented as three Euler angles (roll, pitch, yaw), accompanied by their respective first and second derivatives (angular velocity and acceleration).

$$X = (x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ \ddot{x} \ \ddot{y} \ \ddot{z} \ \psi \ \theta \ \varphi \ \dot{\psi} \ \dot{\theta} \ \dot{\varphi} \ \ddot{\psi} \ \ddot{\theta} \ \ddot{\varphi})^T \quad (1)$$

Next, one should determine the number of measurements. It amounts to 6. These measurements are derived from the rotation (R) and translation (t), yielding the positional coordinates (x, y, z) and the Euler angles (ψ , θ , φ). Furthermore, the number of control actions to apply to the system is specified, which, in this scenario, is zero. Lastly, we define the interval between measurements denoted as the differential time. In this instance, it is calculated as $1/T$,

where T represents the frame rate of the video. We picked some random values of the process noise, measurement noise, and error covariance matrix.

The matrix (2) represents the state transition model A of the Kalman filter, which is used to predict the evolution of the state vector from one time step to the next in a linear dynamic system.

$$\begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 & dt^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 & 0 & dt^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & dt^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & dt^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & dt^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & dt^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The matrix (3) represents the measurement model H of the Kalman filter, which relates the measurements obtained from sensors to the state vector. The first three rows in (3) indicate that the measurements directly correspond to the positional data (x, y, z). The next three rows indicate that the measurements directly correspond to Euler angles (ψ, θ, ϕ) in the state vector.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

6. Detecting brush cleaning area

The Brush Cleaning Areas Recognition module is tasked with identifying specific regions of the brushing area associated with particular brushing techniques. For instance, the Bass technique implies 15 areas, including upper and lower segments on the right and left sides, outer segments, and specific regions within the incisors.

When employing an IMU sensor with 9 Degrees of Freedom (9DOF), contemporary solutions often suggest using deep learning architectures such as CNNs, LSTMs, and RPNs, among others [5, 10]. However, integrating existing implementations with our 6 Degrees of Freedom (6DOF) tracking system poses considerable challenges due to several factors. The unpredictable positioning of the camera within the user's environment, coupled with the variable distance between the camera and the user (ranging from 40 to 70 cm), renders the utilization of X, Y, and Z positioning impractical. Consequently, the available data is limited to rotations around the X, Y, and Z axes (3DOF), presenting significant obstacles in accurately discerning the brushing region even with a comprehensive dataset.

As a result, the efficient adaptation of current methodologies for identifying brushing areas within the context of our tracking system presents notable difficulties, primarily due to the inherent limitations of rotational data and the unpredictable nature of the user's setup.

7. Training module

The Training module is structured to fulfill the following objectives:

- Monitoring and tallying the number of brushing strokes performed by the user.

- Initiating and terminating the brushing timer to track the duration of brushing sessions.
- Assessing the user's proficiency in tooth cleaning techniques based on available data.
- Managing the training session by pausing or resuming activities when the user is absent or present in the frame.
- Providing personalized guidance and recommendations for improvement.

The counting of brushing strokes can be accomplished through two proposed methods. The first method involves utilizing an acoustic sensor, such as a microphone, to detect the occurrence of brush strokes [19]. However, this approach implies that the user has a noise-free environment and relies on the brush strokes audible enough to be registered on the microphone. Alternatively, the second method entails leveraging positioning data obtained from the Pose Estimation Module. This data encompasses the X, Y, and Z coordinates between two consecutive frames, as well as changes in velocity, enabling the calculation of brush stroke occurrences.

Regarding the assessment of the user's proficiency in executing specific tooth-cleaning techniques, one can conduct a comprehensive analysis utilizing various data collected during the training session. These include but are not limited to the total count of brush strokes performed by the user, the average speed of brush strokes executed during the session, the duration of time allocated to brushing specific areas within the oral cavity, the ratio of correctly executed movements to the total number of movements performed. By incorporating these metrics into the evaluation process, one can attain a holistic understanding of the user's performance and adherence to the prescribed tooth-cleaning techniques.

8. Experimental setup

We evaluate each of the proposed and implemented system modules (specifically Face Detection and Toothbrush Pose Estimation), as well as some other mentioned results (described in section 5.1) on the following hardware/software configuration:

- Desktop: OS: Fedora Linux 39 (6.7.3 kernel); CPU: AMD Ryzen 5 5600X (12) @ 3.700GHz; RAM: 32Gb DDR4.
- SW Dependencies: OpenCV: 4.9.0.80; YOLOv8 (Ultralytics): 8.1.11; PySide6: 6.6.1; Python: 3.12.1; Numpy: 1.26.4.
- Webcam: Model: Asus Webcam C3; Resolution: 1280x720; Sensor Resolution: 2 Mpix; FPS: 30; Codec: MJPEG; Exposure: Auto Exposure is disabled, Exposure Time 1/60s; Auto Focus: Disabled.
- Marker (on dodecahedron): Size: 15.8 mm; Dictionary: DICT_5X5_1000; IDs Range: 0 – 11; Type: ArUco.

Camera calibration is of key importance [20, 21, 22]. Our camera is calibrated using a 10x15 chessboard pattern with a marker size of 18 mm, attached to a wooden board (Figure 5).

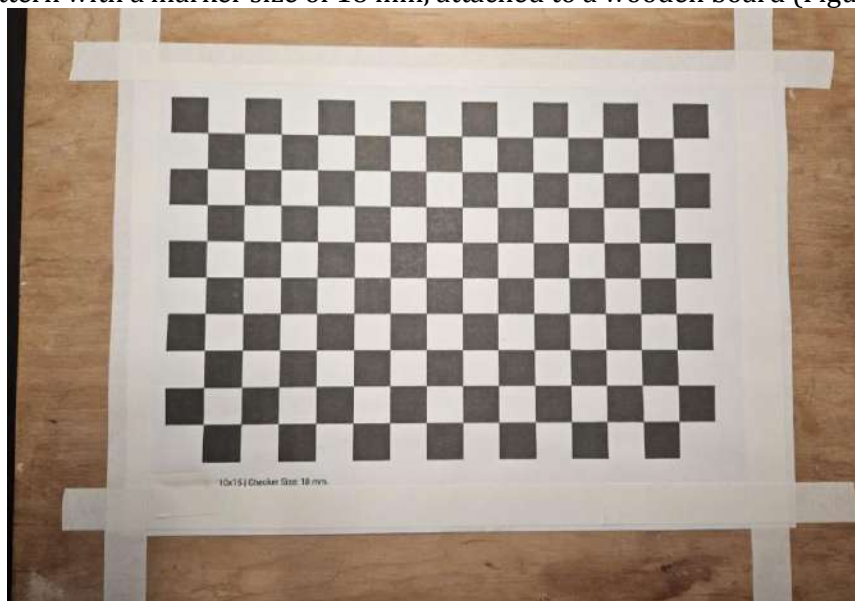


Figure 5: Calibration board (chessboard, 10x15, marker size 18 mm)

Root mean square (RMS) reprojection error – 0.223. The camera intrinsics (camera matrix and distortion coefficients) are the following:

$$\begin{bmatrix} 641.125534 & 0 & 628.572419 \\ 0 & 641.125534 & 351.051382 \\ 0 & 0 & 0 \end{bmatrix} = K$$

$$[0.0519576 \quad -0.0213721 \quad 0 \quad 0 \quad 0.0030762] = D$$

9. Results

9.1. Face detection accuracy and precision

In the assessment of our trained YOLO model, we have selected the following metrics:

- TP (True Positive): Refers to the tally of positively classified samples with accuracy.
- TN (True Negative): Represents the count of negatively classified samples accurately.
- FP (False Positive): Indicates the number of negatively classified samples inaccurately labeled as positive.
- FN (False Negative): Signifies the number of positively labeled samples inaccurately categorized as negative.
- Precision: Quantifies the ratio of TP to the total number of predicted positive instances.
- Recall: Measures the ratio of TP to the total number of actual positive occurrences.
- AP (Average Precision): Provides a measure for assessing the precision-recall curve.
- F1 Score: serving as an overarching performance indicator, reflects the harmonic mean of precision and recall. It is computed by doubling the product of precision and recall, then dividing by their sum.

When using the YOLOv8 model, all essential metrics for evaluating our model's performance are readily available. True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) can be extracted from the confusion matrix (Figure 6). Specifically, TP = 166, FP = 6, FN = 3, TN = 0. The precision is measured at 0.9651, recall at 0.9822, and the F1 score at 0.9736. The overall Average Precision (AP) stands at 0.993. Inference time – 28.3ms.

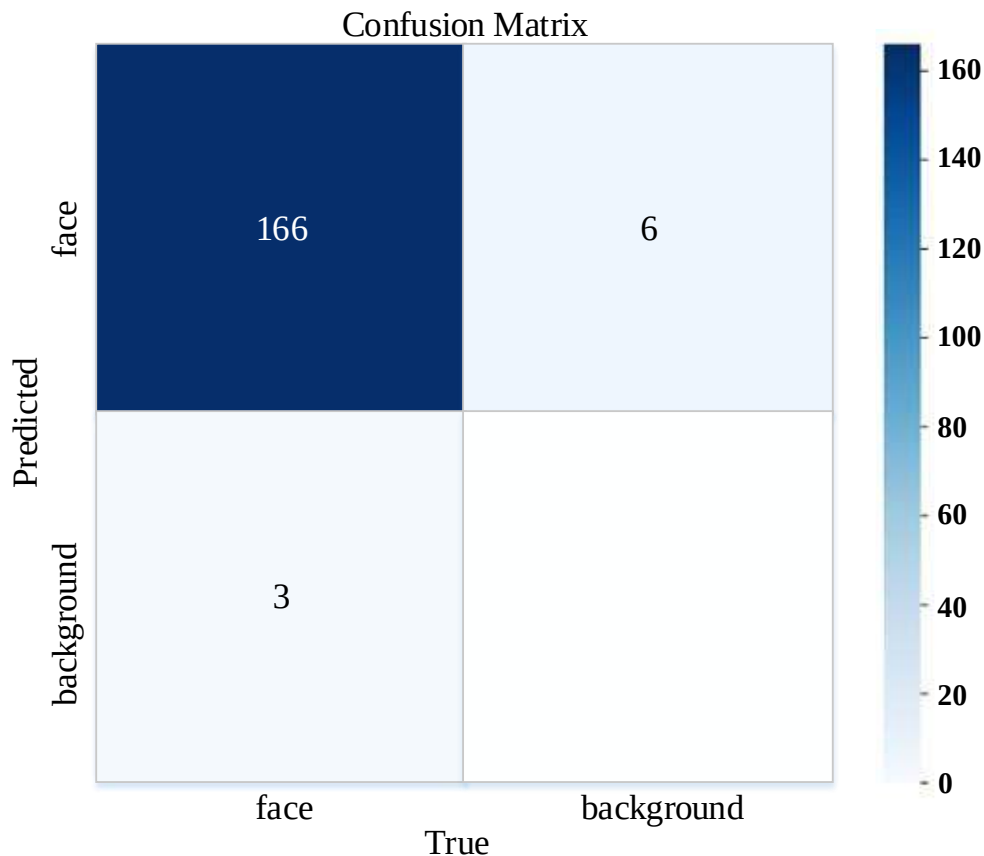


Figure 6: Confusion matrix

9.2. Pose estimation accuracy

To assess the pose estimation accuracy, we established an experimental setup using a grid pattern drawn on an A2 sheet of paper, with each square having a side length of 5 cm. The experiment involved moving a toothbrush incrementally in each plane direction by 5 cm and by 10 cm in the Z-direction while maintaining the camera's position at a fixed Z-coordinate of 1 meter. The experiment outcomes are presented in Table 6.

Table 6

Pose estimation accuracy

	Ground truth	Estimated values	Absolute error	Percentage
X (m) (at Z = 1m)	0.1	0.108	0.008	8%
	0.15	0.159	0.009	6%
	0.2	0.211	0.011	5.5%
Y (m) (at Z = 1m)	0.1	0.1075	0.0075	7.5%
	0.15	0.1611	0.0111	7.4%
	0.2	0.214	0.014	7%
Z (m)	0.8	0.824	0.024	3%
	0.9	0.929	0.029	3.2%
	1	1.031	0.031	3.1%

9.3. Usage of Kalman's filter on bad pose rejection and noise reduction

To evaluate Kalman's Filter efficacy for noise reduction and mitigation of undesired pose fluctuations, we conducted a series of experiments. They were meticulously designed to assess the filter's capability to reject false pose estimations and minimize the ambient noise impact:

1. Steady Movement along the X-axis in one direction: The toothbrush undergoes deliberate, unhurried motion exclusively along the negative direction of the X-axis.
2. Steady Movement along the X-axis in both directions: Similar to the preceding scenario, the toothbrush undergoes deliberate, unhurried motion along the X-axis in both directions.
3. Steady Movement along the X-axis with Marker Occlusion: As in the initial scenario, the toothbrush undergoes consistent motion along the X-axis. However, a distinct feature of this scenario is an intentional occlusion of the marker during the midway point of the trajectory.

In the absence of Kalman filtering (Figures. 7 and 8), the pose exhibits susceptibility to undesirable vibration noise, closely mirroring its fluctuations. Such behavior proves disadvantageous in our application, where stability in pose determination is crucial.

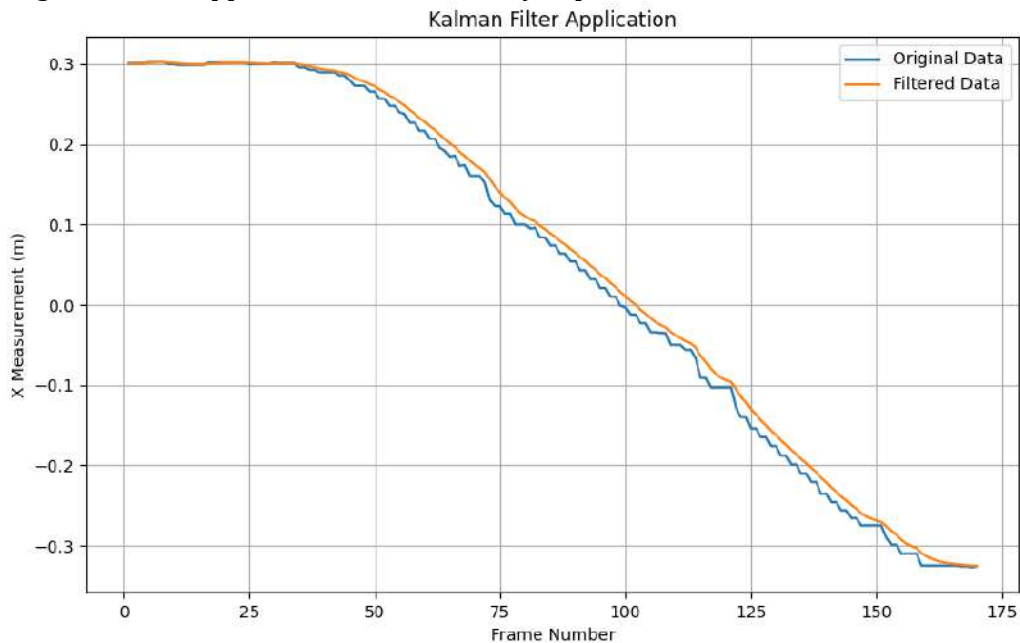


Figure 7: Fluctuations in unfiltered data

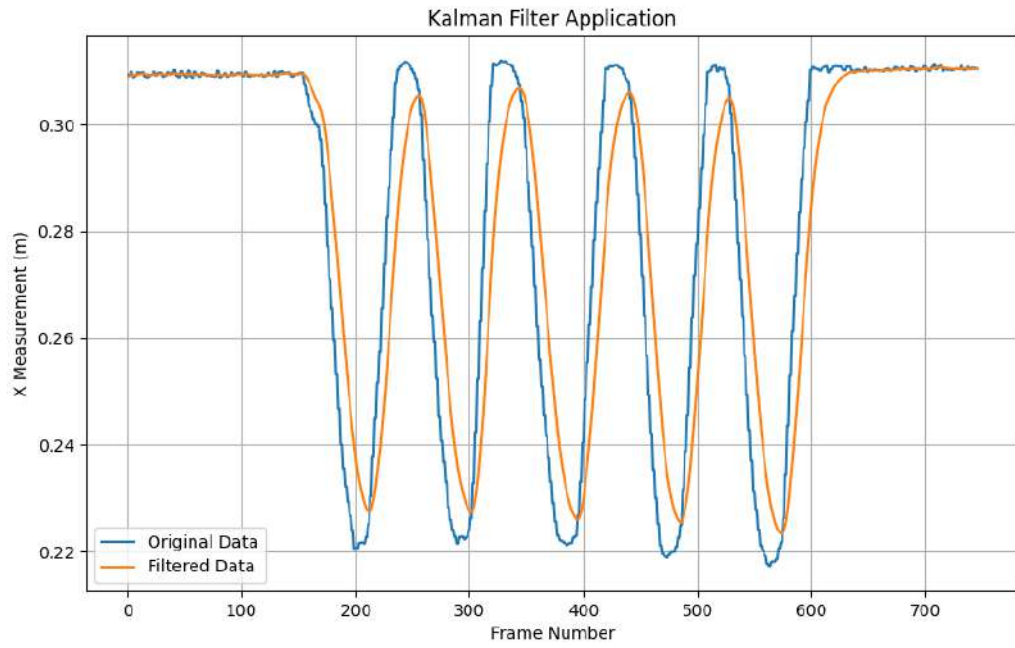


Figure 8: Filtered vs unfiltered data

Conversely, upon the Kalman filter usage, the pose maintains a relatively stable trajectory, mitigating the adverse effects of noise-induced fluctuations.

In scenarios where no pose information is accessible due to occlusion of the ArUco marker (Figure 9), the conventional methods fail to provide reliable estimations. However, with the integration of the Kalman filter into our approach, the system remains capable of inferring the state and anticipating marker locations during these occluded intervals. The outcomes of our analysis demonstrate the algorithm's efficacy in addressing occlusion challenges, yielding optimal estimations despite the absence of direct pose data.

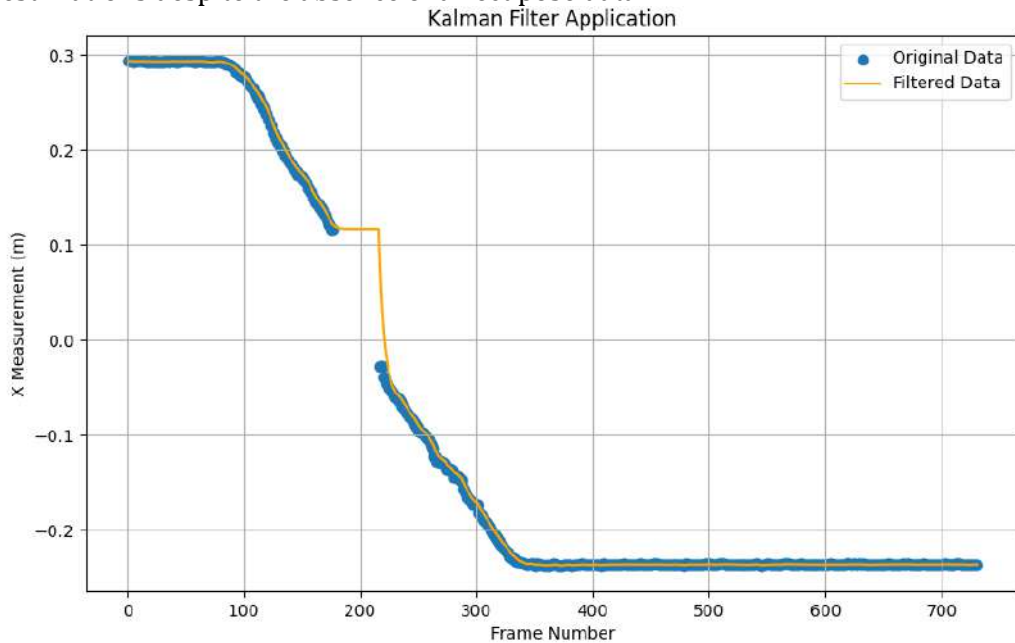


Figure 9: Filtering at ArUco occlusion

10. Discussion

In the proposed system architecture, which holds potential for universal application across similar projects, and exhibits commendable accuracy in pose estimation, several critical impediments to its real-world applicability have been identified.

Initially, the facial detection mechanism within the project predominantly functions as a superficial feature rather than a core component. Its primary use—to ascertain whether a user is

facing the camera, thereby enabling the pause and resume of training sessions and, theoretically, assisting in brush area identification—limits its practical utility. Moreover, the attainment of precise pose estimation is contingent upon fulfilling numerous prerequisites:

- **Camera Calibration:** Effective pose estimation necessitates user-initiated camera calibration, incorporating an automated procedure within our system. Although explored in existing studies, camera calibration is not a straightforward task. The inevitable variance in distortion coefficients is likely to introduce bias into the data collected, thereby compromising the accuracy of pose estimation.
- **Toothbrush Modification:** To utilize our system, users have to engage in a modification process for their toothbrush. It involves the creation of a dodecahedron using a 3D printer and affixing markers onto it. However, the accessibility of 3D printers remains limited, and the precise alignment and attachment of markers to the dodecahedron is crucial. Any inaccuracies in this process may compromise the accuracy of marker detection and pose estimation, thus impacting the overall system functionality.
- **Rolling-shutter Cameras:** Predominantly, contemporary webcams operate on a rolling-shutter mechanism, capturing images not instantaneously but by rapidly scanning the scene. This approach results in predictable distortions of swiftly moving objects or intense light fluctuations, leading to the erroneous detection of markers and inaccurate pose estimation. Solutions to this challenge encompass transitioning to cameras with a global shutter, increasing shutter speed—at the expense of potential exposure issues—and the implementation of advanced Image Processing Algorithms.
- **Lighting Conditions:** The marker detection is linked to lighting conditions, requiring an environment that is neither excessively bright nor dim. This issue is mitigated when employing an IMU solution but is exacerbated by reduced exposure times.

Lastly, the utilization of the proposed 6DOF marker pose estimation system introduces significant challenges in accurately determining the brushing area, a dilemma that demands a viable resolution as detailed in sections 6 and 7 of our analysis.

This comprehensive evaluation underscores the complexities and limitations inherent in the deployment of the proposed system within practical settings, highlighting the necessity for further refinement and adaptation to overcome these obstacles. The future work implies the data fusion of the obtained results with IMU data and gesture recognition [23].

References

- [1] T. A. Ghebreyesus, Global oral health status report: towards universal health coverage for oral health by 2030. regional summary of the African region, 2023. URL: <https://www.who.int/publications/i/item/9789240070769>.
- [2] J. A. Pieren, D. M. Bowen, Darby and Walsh dental hygiene e-book: theory and practice, 5th ed., Elsevier, Amsterdam, 2019.
- [3] D. Slot, L. Wiggelinkhuizen, N. Rosema, G. Van Der Weijden, The efficacy of manual toothbrushes following a brushing exercise: a systematic review: how effective are manual toothbrushes? *International Journal of Dental Hygiene* 10(3) (2012) 187–197. doi: 10.1111/j.1601-5037.2012.00557.x.
- [4] A. B. Londero, A. P. Reiniger, R. C. Tavares, C. M. Ferreira, U. M. Wikesjö, K. Z. Kantorski, C. H. Moreira, Efficacy of dental floss in the management of gingival health: a randomized controlled clinical trial, *Clinical Oral Investigations* 26(8) (2022) 5273–5280. doi: 10.1007/s00784-022-04495-w
- [5] C.-H. Chen, C.-C. Wang, Y.-Z. Chen, Intelligent brushing monitoring using a smart toothbrush with recurrent probabilistic neural network, *Sensors* 21(4) (2021). doi: 10.3390/s21041238
- [6] S. N. Amantini, A. A. Montilha, B. C. Antonelli, K. T. Leite, D. Rios, T. Cruvinel, N. Lourenço Neto, T. M. Oliveira, M.A. Machado, Using augmented reality to motivate oral hygiene practice in children: Protocol for the Development of a Serious Game, *JMIR research protocols* 9(1) (2020). doi: 10.2196/10987.
- [7] S. Hayakawa, G. Al-Falouji, G. Schickhuber, R. Mandl, T. Yoshida, S. Hangai, A method of toothbrush position measurement using AR markers, in: 2020 IEEE 2nd Global Conference

- on Life Sciences and Technologies, LifeTech, Kyoto Japan, 2020, pp. 91–93. doi: 10.1109/LifeTech48969.2020.1570619103.
- [8] M. Fahim, V. Sharma, T. Q. Duong, A wearable-based preventive model to promote oral health through personalized notification, in: 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC'2022, IEEE, Glasgow United Kingdom, 2022, pp. 4282–4285. doi: 10.1109/EMBC48229.2022.9871128.
 - [9] H. Kondo, K. Funahashi, AR tooth brushing system to promote oral care habits of children, in: 2021 Nicograph International, NicoInt'21, IEEE, Tokyo Japan, 2021, pp. 115–115. doi: 10.1109/NICOINT52941.2021.00033.
 - [10] Z. Hussain, D. Waterworth, M. Aldeer, W. E. Zhang, Q. Z. Sheng, J. Ortiz, Do you brush your teeth properly? An off-body sensor-based approach for toothbrushing monitoring, in: 2021 IEEE International Conference on Digital Health, ICDH'21, IEEE Chicago IL USA, 2021, pp. 59–69. doi: 10.1109/ICDH52753.2021.00018.
 - [11] D. Fedasyuk, R. Holyaka, T. Marusenkova, A tester of the MEMS accelerometers operation modes, in: 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT'19, IEEE, Lviv Ukraine, 2019, pp. 227–230. doi: 10.1109/AIACT.2019.8847840.
 - [12] D. Fedasyuk, R. Holyaka, T. Marusenkova, Method of Analyzing Dynamic Characteristics of MEMS Gyroscopes in Test Measurement Mode, in: 2019 9th International Conference on Advanced Computer Information Technologies, ACIT'19, IEEE, Ceske Budejovice, Czech Republic, 2019, pp. 157–160, doi: 10.1109/ACITT.2019.8780058.
 - [13] G. Yocher, A. Chaurasia, Ultralytics YOLOv8 Docs, 2023. URL: <https://docs.ultralytics.com/>.
 - [14] D. Rempel, K. Willms, J. Anshel, W. Jaschinski, J. Sheedy, The effects of visual display distance on eye accommodation, head posture, and vision and neck symptoms, *Human Factors* 49(5) (2007) 830–838. doi: 10.1518/001872007X230208.
 - [15] H. -Y. Tseng, P. -C. Wu, M. -H. Yang, S. -Y. Chien, Direct 3D pose estimation of a planar target, in: 2016 IEEE Winter Conference on Applications of Computer Vision, WACV, Lake Placid, NY, USA, 2016, pp. 1–9. doi: 10.1109/WACV.2016.7477640.
 - [16] P.-C. Wu, R. Wang, K. Kin, C. Twigg, S. Han, M.-H. Yang, S.-Y. Chien, DodecaPen: Accurate 6DoF tracking of a passive stylus, in: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST'17, Association for Computing Machinery, Québec City QC Canada, 2017, pp. 365–374. doi: 10.1145/3126594.3126664.
 - [17] P. García-Ruiz, F. J. Romero-Ramirez, R. Muñoz-Salinas, M. J. Marín-Jiménez, R. Medina-Carnicer, Fiducial objects: custom design and evaluation, *Sensors* 23(24) (2023). doi: 10.3390/s23249649.
 - [18] H. C. Kam, Y. K. Yu, K. H. Wong, An Improvement on ArUco Marker for Pose Tracking Using Kalman Filter, in: 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), IEEE, Busan Korea (South), 2018, pp. 65–69. doi: 10.1109/SNPD.2018.8441049.
 - [19] H. Huang, S. Lin, Toothbrushing monitoring using wrist watch, in: Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM, SenSys '16, ACM, Stanford CA USA, 2016, pp. 202–215. doi: 10.1145/2994551.2994563.
 - [20] H. Rezazadegan Tavakoli, H. R. Pourreza, An automated camera calibration framework for desktop vision systems, in: 2009 International Conference on Advances in Computational Tools for Engineering Applications, IEEE, Beirut Lebanon, 2009, pp. 96–100. doi: 10.1109/ACTEA.2009.5227921.
 - [21] L. Tan, Y. Wang, H. Yu, J. Zhu, Automatic camera calibration using active displays of a virtual pattern, *Sensors* 17(4) (2017). doi: 10.3390/s17040685.
 - [22] S. Su, W. Heidrich, Rolling shutter motion deblurring, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Boston, MA, USA, 2015, pp. 1529–1537. doi: 10.1109/CVPR.2015.7298760.
 - [23] L. Ivanska, T. Korotyeyeva, Mobile real-time gesture detection application for sign language learning, in: 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies, CSIT'22, IEEE, Lviv Ukraine, 2022, pp. 511–514. doi: 10.1109/CSIT56902.2022.10000440.

Hybrid Neural Network Identifying Complex Dynamic Objects: Comprehensive Modelling and Training Method Modification

Victoria Vysotska¹, Serhii Vladov², Ruslan Yakovliev² and Alexey Yurko³

¹ Lviv Polytechnic National University, Stepan Bandera Street 12, Lviv, 79013, Ukraine

² Kremenichuk Flight College of Kharkiv National University of Internal Affairs, Peremohy Street 17/6, Kremenichuk, 39605, Ukraine

³ Kremenichuk Mykhailo Ostrohradskyi National University, University Street 20, Kremenichuk, 39600, Ukraine

Abstract

The article to the development of mathematical models of complex dynamic objects (using the example of helicopter turboshaft engines) in the form of recurrent neural networks and their use in complex modelling to identify the parameters of automatic control, monitoring, and diagnostic systems is described. For the first time, a concept has been created for constructing neural network models of complex dynamic objects, in which, by increasing the robustness of the functioning of a trained neural network, it becomes possible at its output to increase the reliability of solving problems of identifying complex dynamic objects. The use of a hybrid neural network NARX with a radial-basis nonlinear layer is proposed. The introduction of a radial basis nonlinear layer into the NARX neural network is an effective addition when working with unstructured data such as images, audio signals, or text due to its ability to extract and represent complex patterns and features in these data. This is confirmed by the results of modelling the losses of the neural network, which turned out to be stable over 500 epochs of its training and did not exceed 0.025 (2.5 %). A comprehensive modification of the Levenberg-Marquardt training method is proposed, which consists of a particular application of the Broyden method for calculating the elements of the Hessian matrix, as well as an analytical description of the regularization parameter through the use of control coefficients for increasing or decreasing its value in the event of a neural network training error. The use of the modified Levenberg-Marquardt method made it possible to reduce the average training error of the NARX hybrid neural network with a radial basis layer by 33 % to the level of 0.025.

Keywords

Neural network, helicopters turboshaft engines, complex dynamic objects, hybrid neural network NARX, radial-basis nonlinear layer, Levenberg-Marquardt method, training, Broyden method, loss

1. Introduction

Experimental research and modelling of complex dynamic objects, for example, helicopter turboshaft engines (TE) and their control systems, are constant elements of knowledge of their behaviour throughout the entire life cycle. This starts from the design stage, fine-tuning, and certification, and ending with operation and disposal. Such studies require a special integrated modelling technology creation. It makes it possible to confirm the reliability, operability and required characteristics of systems both before putting them into operation and in operating modes [1, 2]. Today, the development of the industry is based on technologies of digital manufacturing, computer modelling, machine learning, cloud computing, and cyber-physical systems. The *digital twins* concept is being fully implemented. This is a virtual representation of a physical object not only at the stages of design, development, and commissioning but also throughout the entire life cycle, including operation and disposal [3, 4].

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ victoria.a.vysotska@lpnu.ua (V. Vysotska); ser26101968@gmail.com (S. Vladov); director.klk.hnuvs@gmail.com (R. Yakovliev); yurkoalexe@gmail.com (A. Yurko)

🆔 0000-0001-6417-3689 (V. Vysotska); 0000-0001-8009-5254 (S. Vladov); 0000-0002-3788-2583 (R. Yakovliev); 0000-0002-8244-2376 (A. Yurko)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The technology of semi-natural modelling of complex control objects has been used for a long time in many industries, where real systems are coupled with mathematical models of control objects. However, the methods and tools for creating such models often remain the same [5, 6].

One of the most pressing and important problems is to ensure the adequacy of the model of a complex dynamic object in the automatic control, monitoring, and diagnostics system. The operation of control, monitoring, and analysis algorithms at the same time can cause various *collisions* that need to be taken into account and modelled when developing and configuring a control system [7]. In addition, in the process, complex dynamic objects gradually exhaust their resources, and their characteristics begin to degrade [8, 9]. In the process of analysis and synthesis of automatic control systems, the need arises to correct and adapt the existing model of a complex dynamic object for its effective operation. To solve this problem, adaptive models are needed that are identified by the real characteristics of the object and its operating conditions.

The widespread application of intelligent technologies utilizing neural networks has been observed in the research and development of sophisticated control and monitoring systems tailored for complex dynamic objects [10, 11], including helicopters TE [12, 13]. However, the task remains of the adequacy and applicability of mathematical models of complex dynamic objects in operating modes, which are mostly presented in the form of fast-calculating piecewise linear dynamic models [14, 15].

The research aim is to increase complex modelling and testing efficiency of a real automatic control system (ACS) for complex dynamic objects, monitoring, and diagnostics through the use of nonlinear dynamic mathematical models and their systems in the form of neural networks (using the example of helicopters TE). A scientific concept is proposed for the neural network model constructing for a complex dynamic object (using the example of helicopters TE), including algorithms for training and identifying a mathematical model of the engine using real data with a choice of the structure and size of the neural network.

2. Related works

Currently, neural networks are an effective means of solving a wide range of problems in complex dynamic objects identifying and their control systems [16, 17]. They are distinguished by the simplicity of the architecture and high representative power. The quality of operation of these networks largely depends on the efficiency of data clustering, as a result of which the centres of activation functions and their dispersion are determined. The works of individual domestic and foreign scientific schools are devoted to the issue of automating the neural network architecture selection. Also, this work focuses both on local modifications of training algorithms [18, 19] and on the use of bionic models [20, 21] to optimize the number of neurons in the hidden layer [22, 23]. The latter has good potential. Because it leads to growing interest in the use of distributed intelligent systems to optimize neural network architecture [24, 25]. An alternative to the described solutions is methods based on special approaches to density clustering. As a result, in hidden layer the optimal neurons number is determined and their key characteristics are established [26, 27]. A common disadvantage of most known solutions is the requirement for the completeness of processed samples. This makes the use of such methods for working in systems with dynamically changing data, for example in control systems, ineffective compared to such specialized neuroarchitectures as Jordan networks [28, 29], Elman networks [30, 31] or recurrent multilayer perceptron [32, 33].

3. Methods and materials

To solve the above task of complex dynamic objects and their control systems, an intelligent system (Fig. 1) can be used that implements the Fault Detection and Identification Method (FDI) [34]. It is based on a neural network mathematical model of the research object and an identification block [35, 36]. Such a system makes it possible to detect and classify abnormal operating modes of the research object, measuring channels and actuators under operating conditions. The output parameters of the mathematical model can be used to diagnose abnormal operating conditions of the research object based on a comparison of the matching the computed parameters with the observed ones. Additionally, they used and also as to restore lost data of

measuring channels in the event of their failure being detected. Such a model should have several special properties, the most important of which are [37, 38]:

1. The model must describe the properties of the research object that determine the non-stationary nature of work processes. This means the need to use a dynamic model.
2. The structure of the mathematical model of the research object should provide the practical possibility of its functioning in combination with mathematical models of other elements of the system.

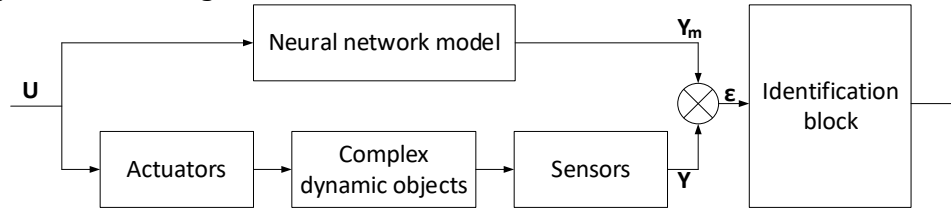


Figure 1: Structure of an intelligent system for identifying complex dynamic objects and their control systems based on the FDI method (author's research, based on [24–36])

The mathematical representation of the nonlinear dynamic model of complex dynamic objects, taking [39] into account, can be represented as a differential equations system:

$$\begin{cases} \frac{\Delta y_1(t)}{\partial t} = a_{11} \cdot y_1(t) + \dots + a_{1n} \cdot y_{n-1}(t) + b_{11} \cdot x_1(t) + \dots + b_{1m} \cdot x_m(t), \\ \frac{\Delta y_2(t)}{\partial t} = a_{21} \cdot y_1(t) + \dots + a_{2n} \cdot y_{n-1}(t) + b_{21} \cdot x_1(t) + \dots + b_{2m} \cdot x_m(t), \\ \dots \\ \frac{\Delta y_n(t)}{\partial t} = a_{n1} \cdot y_1(t) + \dots + a_{nn} \cdot y_{n-1}(t) + b_{n1} \cdot x_1(t) + \dots + b_{nm} \cdot x_m(t). \end{cases} \quad (1)$$

Currently, neural network models built on linear neural networks, for example, multilayer perceptron, have been acquired for the identification of complex dynamic objects [40, 41]. Neural network models based on linear neural networks (as multi-layer perceptron) may face several disadvantages when modelling complex dynamic objects. First, they are limited in their ability to capture complex nonlinear relations between variables. This can lead to insufficient accuracy in predicting or modelling. In addition, multi-layer perceptron can suffer from the problem of gradient damping when training deep models. This makes training difficult and can lead to low performance in practice. Also, they can be prone to overtraining in the presence of a limited amount of training data. This makes them less effective for processing real dynamic systems. Therefore, it is expedient to use dynamic recurrent neural networks, for example, recurrent multilayer perceptron (NARX). The justification for the transition from a multilayer perceptron to a recurrent multilayer perceptron (NARX) is based on several factors and is given in Table 1, and the scientific features of this transition are in Table 2.

Table 1

Factors for the transition from a multilayer perceptron to a hybrid NARX network for the identification of complex dynamic objects (author's research)

Factor	Description
Complexity of the system	Complex dynamical systems have non-linear behaviour and relations between different parameters. Multilayer perceptron may not be flexible enough to model such a complex system.
Taking into account the dynamics	The NARX hybrid network provides an opportunity to take into account the dynamic properties of the system based on time dependencies between inputs and outputs. This is especially important for predicting the parameters of the work process of the research object, which may change over time.
Handling uncertainty	Complex dynamic objects are exposed to various external influences and changes in operating conditions. Hybrid NARX networks can better adapt to data uncertainty and provide more accurate predictions under different conditions.
Use of contextual data	NARX hybrid networks allow you to include both current parameter values and historical data in the model. This can be useful for analyzing previous system states and identifying patterns of parameter changes

Table 2

Scientific substantiation of the peculiarity of the transition from a multilayer perceptron to a hybrid NARX network (author's research)

Factor	Description
Provision no. 1. Innovative modelling of complex dependencies through a combination of nonlinear autoregressive models and radial basis functions	
Combining nonlinear autoregressive models and radial basis functions	Nonlinear autoregressive NARX models are capable of capturing complex dynamic relationships between input and output variables. Nonlinear functions bring flexibility to the model, which allows you to adapt to different forms of data, which is especially important when modelling input parameters with a nonlinear nature.
Effective modeling of complex dependencies	The workflow parameters of complex dynamic objects can exhibit complex, non-linear dependencies that are best described by the hybrid NARX network. This makes it possible to more accurately predict changes in parameters under different operating conditions.
Provision no. 2. Feasibility of solving the problem of identification of complex dynamic objects using the NARX hybrid network and comparing its results with the use of a multilayer perceptron	
Modelling accuracy	Comparison of accuracy coefficients, such as root mean square deviation, on training and test samples.
Generalization of new data	It is estimated how well the model generalizes its knowledge to new data not used during training.
Stability	Analysis of the stability of models in various conditions, including changes in external parameters.
Provision no. 3. Simultaneous use of simulation results using multilayer perceptron and hybrid NARX network in operational conditions	
Using a multilayer perceptron	They are used to quickly assess current input parameters and provide a response to rapidly changing conditions.
Using the hybrid NARX network	They are used for deeper analysis of dynamic changes in parameters, taking into account time dependencies and predicting future values.
Conclusions	
The combined use of a multilayer perceptron and a hybrid NARX network allows combining the advantages of both models, providing more accurate and flexible real-time identification for complex dynamic objects.	

In [40], the use is justified by a modified version of a recurrent multilayer perceptron (NARX). It is a dynamic network characterized by the delay of output/input signals combined in a network input vector, with a radial basis nonlinear and a linear recurrent layer. It should be noted that [40] uses a Gaussian NARX framework with input data regressor selection using a modified gradient method from [41]. This modification is justified based on the outdated NARX models with outdated machine training models [42]. The modified NARX structure proposed in [40] consists of two parts: nonlinear and linear blocks (Fig. 2, where σ_i is i -th element radial function width; $c_{i1}, c_{i2}, \dots, c_{in}$ are i -th element coordinates centre; $u_1, u_2, \dots, u_n$ are the input signals).

Such a model in a neural network form with feedback makes it possible to take into account the nonlinear dynamic characteristics of an object and guarantee the structural and parametric adequacy of its analytical model. The vector $\mathbf{u}$ placed to the input has the form $u(t) = [1, u(t), u(t-1), \dots, u(t-N_u), y(t-1), \dots, u(t-N_y)]^T$, where N_u – is the input signal delays number, N_y – is the output signal delays number [40]. Depending on the complex dynamic object model, the vector $\mathbf{u}$ is formed according to the parameters specified in the technical specifications.

According to [33, 34, 37], the network output vector has the following mapping [40]:

$$y(t+1) = f(u(t), y(t-1), \dots, y(t-N_y), u(t-1) \dots, u(t-N_u)), \quad (2)$$

then the NARX hybrid network is characterized by a set of numbers (N_u, N_y, N_i) , where N_i is neurons in the i -th hidden layer number.

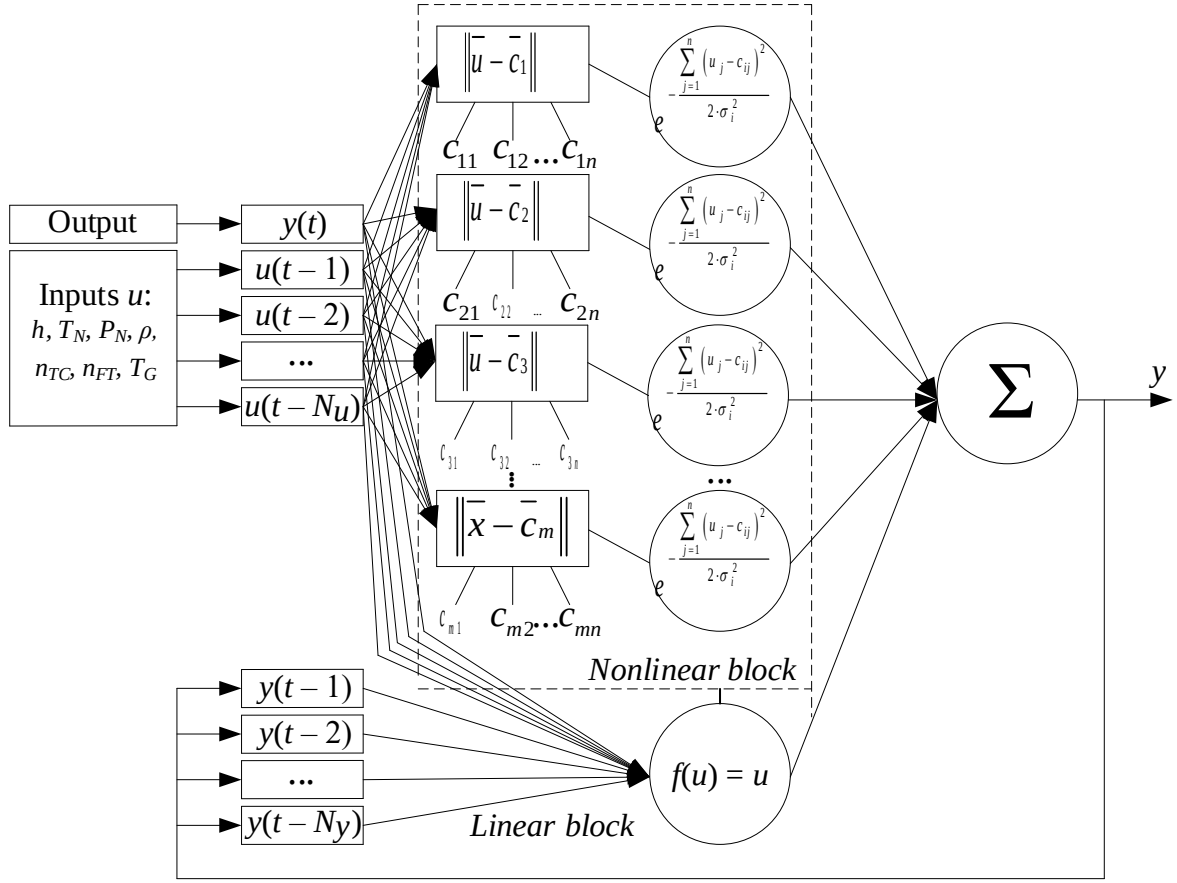


Figure 2: Gaussian NARX architecture using modified neural network (author's research [40])

To the aforementioned, the comprehensive schematic representation of configuring the neural network model parameters for complex dynamic objects (using the example of helicopters TE [40]) is presented in Fig. 3, where Δw_{ij} is the neural network synaptic connections increase; $\mathbf{Y} = (y_1, y_2, \dots, y_m)^T$ is the object output parameters vector; $\mathbf{U} = (u_1, u_2, \dots, u_m)^T$ is the input influences vector; $\mathbf{Y}^{NN} = (y_1^{NN}, y_2^{NN}, \dots, y_n^{NN})^T$ is the neural network outputs vector [40].

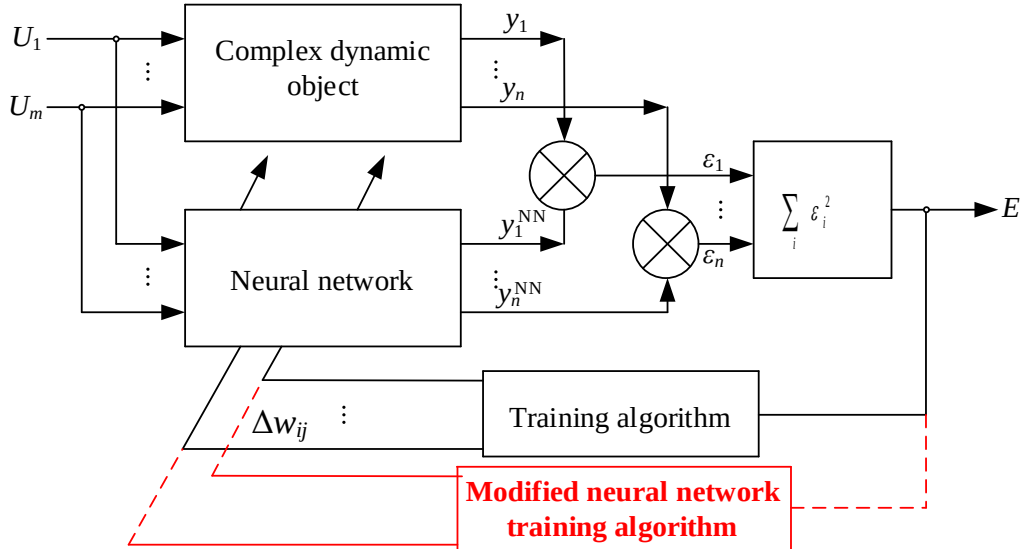


Figure 3: Complex dynamic objects neural network model scheme (author's research [40])

Modifications of training of neural networks are conducted to improve performance or adjust the model for the specific requirements of the task (Fig. 3). Modifying a neural network training includes changing its structure based on adding or removing layers and neurons, changing activation functions, and adjusting hyperparameters. Also, this includes training rate or regularization parameters, and introducing additional training methods, such as data

augmentation or based on pre-trained models for transferring training. Modification may also be necessary if the input data characteristics change, the performance requirements change, or the problem that the neural network needs to solve changes. Control influences vector conversion into initial parameters vector is elucidated by operator $\mathbf{F}$ [40]:

$$\mathbf{Y} = \mathbf{F}(\mathbf{U}). \quad (3)$$

The identifying helicopter EY task using a neural network can be formulated: using the results of the proposed training process for a neural network. It forms vectors $(\mathbf{U}_i; \mathbf{Y}_i)$ *training set* obtained experimentally for a separate engine instance. Aim is to find operator $\mathbf{F}^{\text{NN}}$ within neural network architectures class. The $\mathbf{F}$ operator approximation by $\mathbf{F}^{\text{NN}}$ operator is deemed optimal if a specified functional from the difference $(\mathbf{Y} - \mathbf{Y}^{\text{NN}})$ does not surpass a given small value ε_{add} , defining the $\mathbf{F}$ operator approximation accuracy [40]:

$$E = \|\mathbf{Y} - \mathbf{Y}^{\text{NN}}\| = \sum_i^n \varepsilon_i^2 \leq \varepsilon_{\text{add}}. \quad (4)$$

The condition (4) satisfaction is guaranteed by neural network training. It involves finetuning parameters using the training sample $\{(\mathbf{U}, \mathbf{Y})\}$ and is verified on a meticulously organized *test sample* [40]. A scientific concept of direct neural network model construction based on complex dynamic objects is proposed, which is shown in Table 3.

Table 3

The scientific concept of the step-by-step creation of a neural network model of complex dynamic objects (author's research)

Step	Description
1	Development of unique criteria and metrics for evaluating the effectiveness of identification of complex dynamic objects. Justification of the need to define goals to ensure the accuracy and objectivity of the assessment.
2	The rationale for selecting a particular neural network architecture and identifying its integration point within the complex dynamic object identification system.
3	Analysis and justification of the network training algorithm choice considering the specifics of the task to achieve optimal adaptive training.
4	Description of conducting experiments on a digital model using additional resulting results to create a training sample and taking into account new criteria and metrics to improve model accuracy.
5	Network training process description using formed training sample and training algorithm.
6	Justification for the neural network simplifying and reducing to achieve optimal information storage and efficient operation with a minimum number of parameters.
7	Justification of measures aimed at increasing the robustness of the functioning of the trained neural network model, taking into account possible challenges and unexpected situations.
8	Modelling and testing algorithms description for monitoring the operational status and operation for complex dynamic objects management, including ACS based on a neural network.
9	Justification of the choice of software or hardware development of a neural network for implementation in a real system of identification of complex dynamic objects.

The proposed scientific concept of the step-by-step creation of a neural network model of complex dynamic objects defines clear stages of intelligent systems development and implementation. The results of the work of each stage (the development of evaluation criteria, the selection of a network structure, the analysis of training algorithms, and experimental research on a digital model) form a reliable basis for the creation of an effective intelligent monitoring system. In particular, the justified reduction of the neural network and measures to increase the robustness of the model indicate a desire for optimal efficiency. The final stages (modelling and testing algorithms, as well as the choice of software or hardware implementation) emphasize the practical suitability of the concept for implementation in a real intelligent system for the identification of complex dynamic objects.

For the practical implementation of the proposed scientific concept of the step-by-step creation of a neural network model of complex dynamic objects (Table 3), attention should be drawn to the indicator of model robustness (*generalization ability*). This is the stability of modelling results to input data disturbances. Evaluation of the resilience or generalization capability of a neural network model is conducted using an algorithm grounded on incremental multi-criteria training [44].

Complex dynamic object model training using a neural network (Fig. 2) is performed by sequentially presenting pre-prepared delay vectors and corresponding output values while simultaneously adjusting the weights of hidden layers by a certain procedure [45, 46]. The neural network training process includes the application of the Levenberg-Marquardt method. The method combines the steepest descent methods (i.e., minimizing the training error along a gradient) and Newton's method (i.e., using a quadratic model to accelerate the quest for the minimum of the error function [47]). The Levenberg-Marquardt method is intended for optimizing nonlinear regression models of the form $F(\mathbf{u}) = \sum_{i=1}^N (f_i(\mathbf{u}) - y_i)^2$. As an optimization criterion, it uses the model mean square error (MSE) on the set training, which it minimizes [40].

The Levenberg-Marquardt method combines the ideas of the Gauss-Newton method and gradient descent. At each iteration, this method updates the parameters $\mathbf{u}$ in this way:

$$\mathbf{u}_{k+1} = \mathbf{u}_k - (J^T J + \lambda_k I)^{-1} J^T \Delta \mathbf{y}, \quad (5)$$

where J – is the Jacobian matrix of $J_{ik} = \frac{\partial f_i}{\partial u_k}$, describing partial derivatives of a model concerning parameters; $\Delta \mathbf{y}$ – is the vector representing the disparity between the present model values and the real data; I – is the identity matrix; λ_k – is the regularization parameter that controls the step size at each iteration.

According to the research of Serhii Parkhomenko [47, 48], most often, to search for the Jacobian matrix, various variations of difference formulas for calculating derivatives are used, including the central difference derivative [49]. According to [47, 48] for the hybrid NARX network y_i values can be written as:

$$y(u, w) = f_1 \left(\sum_{i=1}^{N_i} \tilde{w}_i \cdot f_2 \left(\sum_{j=1}^{N_u} w_{ij} \cdot u_j + \sigma_i \right) \cdot f_3 \left(\sum_{l=1}^{N_l} w_{il} \cdot u_l + \sigma_i \right) + \tilde{\sigma} \right), \quad (6)$$

where $y(u, w)$ shows the dependence of an output value y_i on input parameters vector values u and corresponding weighting coefficients w , u_j – is the value received by the j -th input neuron, w_{ij} – is the weight coefficient connecting the j -th input neuron with i -th nonlinear hidden layer neuron, σ_i – is the bias coefficient for i -th nonlinear hidden layer neuron, $\tilde{w}_i$ – is the weight coefficient connecting i -th neuron of the nonlinear hidden nonlinear layer with the output neuron, $\tilde{\sigma}$ – is the bias coefficient for the output neuron, $f_1(\bullet)$, $f_2(\bullet)$, $f_3(\bullet)$ – are the activation functions for the output neuron and neurons of hidden nonlinear and linear layers, respectively, N_u – is the number of the input, N_i and N_l – are the hidden nonlinear and linear layers neurons number, respectively. The radial basis function is chosen as the activation function for neurons of the output and nonlinear hidden layers according to [40] and for neurons of the linear hidden layer – a linear function.

According to [47, 48] when calculating the Hessian matrix, it is convenient to use the formula $\mathbf{H} = J^T J$, which is derived from the premises:

- the function $y(u, w)$ has a low order of nonlinearity: the second derivatives $\frac{\partial^2 y(u, w)}{\partial w_i \partial w_j}$ do not take very large values;
- matrix $\mathbf{H}$ is considered in a small neighbourhood of the minimizing vector w , for which the $y(u, w)$ values are close to the desired $f_i(\mathbf{u})$, i.e. $|f_i(\mathbf{u}) - y(u, w)| \approx 0$.

With an efficient operation of scalar matrix multiplication, the search for $\mathbf{H}$ is fast, while the time for calculating the vector of weight changes depends on the variable number w_{ij} . Experiments show [47, 48] that it is rarely necessary to solve more than three systems per iteration, which does not greatly affect the execution time of the algorithm. Calculating the Jacobian matrix takes up the bulk of the work of the Levenberg-Marquardt algorithm. So reducing the cost of searching for it speeds up the neural network training. One of these methods is to abandon the calculation of the completely accurate matrix J in favour of its approximate version.

For example, the Broyden method calculates J_{n+1} using the matrix J_n calculated at step n according to the formula [47, 48, 50]:

$$J_{n+1} = J_n + h^T \cdot \frac{y(u, w_{n+1}) - y(u, w_n) - J_n \cdot h}{h^T \cdot h}, h = w_{n+1} - w_n, \quad (7)$$

From a theoretical point of view, using this approach at each step of the Levenberg-Marquardt algorithm makes sense. However, in practice, the approximation becomes coarser over time. This affects the $J^T E$ gradient vector and requires re-calculation of the Jacobian matrix using more accurate methods after an unsuccessful selection of a vector of weight changes δ . Analytical calculation of partial derivatives improves the accuracy of calculations. This allows you to shorten the process by reusing intermediate data and reducing the number of calls to complex functions. Similar to [47, 48], the following substitution was made in this work:

$$s_i = \sum_{j=1}^{N_u} w_{ij} \cdot u_j + \sigma_i, \quad S = \sum_{i=1}^{N_i} \tilde{w}_i \cdot f_2(s_i) \cdot f_3(s_i) + \tilde{\sigma}; \quad 1 \leq j \leq N_u. \quad (8)$$

Thus, we get:

$$\begin{aligned} \frac{\partial y(u, w)}{\partial w_{ij}} &= u_j \cdot [E \cdot C \cdot s_i], \quad \frac{\partial y(u, w)}{\partial \sigma_i} = [E \cdot C \cdot s_i], \quad \frac{\partial y(u, w)}{\partial \tilde{w}_i} = \frac{[E \cdot C'] \cdot [E \cdot s_i]}{1 + [E \cdot s_i]}, \\ \frac{\partial y(u, w)}{\partial \tilde{\sigma}} &= [E \cdot C'], \quad C = \frac{\partial^2 C}{\partial S^2} \cdot [E \cdot S], \quad \frac{\partial^2 C}{\partial S^2} = \alpha \cdot [E \cdot C'], \quad \frac{\partial C}{\partial s_i} = \frac{\tilde{w}_i \cdot [E \cdot C'^2] \cdot [E \cdot s_i]}{(1 + [E \cdot s_i])^2}. \end{aligned} \quad (9)$$

The analytical method of calculating the Jacobian matrix is not applicable in all cases, since for each new neural network model it is necessary to revise the formulas. However, it requires less computation than using central difference derivatives while maintaining accuracy. Distributing calculations of the rows of matrix J between threads allows it to be filled in parallel since the elements are calculated independently. This corresponds to the ribbon pattern. According to [48] τ – is the maximum number of threads running simultaneously. For $\tau \ll N$, each thread on average processes about $\frac{N}{\tau}$ rows of matrix J . Row J represents the minimum unit of processing within a parallel block, Ξ – set of numbers of rows J processed by thread t . Since $H^* = J^T J$, then, according to [48], to save memory, you don't have to store the J^T and J matrices separately. For any $n \in [1, N]$ and $m \in [1, M]$, the equality $J_{nm} = J_{nm}^T$ is true, which allows you to get elements J^T and J by swapping the indices. To calculate the Hessian matrix element according to [48]

$$H_{pq}^* = \sum_{i=1}^{N_u} J_{qi}^T \cdot J_{ip} = \sum_{i=1}^{N_u} J_{iq} \cdot J_{ip}; \quad 1 \leq p \leq M; \quad 1 \leq q \leq M. \quad (10)$$

you need to know all the elements of the p -th and q -th columns of J . The simplest way is to wait for an entire matrix J to be calculated, but this causes a synchronization point where all processes must wait for the calculation to complete before continuing. Decomposing the matrix H^* into a sum allows us to avoid this [48]:

$$H^* = \sum_{t=1}^{\tau} ([^t]H) = [^1]H + [^2]H + \dots + [^{\tau}]H = \left\{ [^t]H_{pq} \right\}_{M \times M}, \quad (11)$$

where $[^t]H$ – is the matrix of the cumulative sum of flow t , $[^t]H_{pq}$ – is the element of this matrix.

For each calculated row of matrix J in [48], all its elements are multiplied by each other, which leads to obtaining a matrix term:

$$[^t]H_k^+ \Big|_{ij} = J_{ik}^T \cdot J_{kj} = J_{ki} \cdot J_{kj}; \quad k \in \Xi, \quad 1 \leq i \leq M; \quad 1 \leq j \leq M, \quad (12)$$

where $[^t]H_k^+ \Big|_{ij}$ – is the element of the matrix $[^t]H_k^+$.

By combining the matrices of one stream, a cumulative sum matrix is formed in the form:

$$[^t]H = \sum_{k \in \Xi_i} [^t]H_k^+, \quad (13)$$

which, when added with matrices from other streams outside the parallel region, results in a finite Hessian matrix H^* .

Thus, combining the calculations of row J and the matrix H within one parallel block was achieved by distributing the calculations of a scalar matrix product over ${}^{[t]}H$. Calculation $g = J^T \tilde{\varepsilon}(w)$ also requires all elements of the p -th column of matrix J [48]:

$$g_p = \sum_{i=1}^{N_u} J_{pi}^T \tilde{\varepsilon}(w) = \sum_{i=1}^{N_u} J_{ip} \tilde{\varepsilon}(w), 1 \leq p \leq M. \quad (14)$$

By decomposing the vector g into cumulative vectors ${}^{[t]}g$ for each thread, and then decomposing them into term vectors ${}^{[t]}g_k^+$ for all rows J processed within a single thread, we obtain a similar calculation method used to calculate H^* :

$${}^{[t]}g_k^+ \Big|_i = J_{ik}^T \tilde{\varepsilon}(w) = J_{ki} \tilde{\varepsilon}(w), g = \sum_{k \in \Xi_i} {}^{[t]}g_k^+, g = \sum_{t=1}^{\tau} {}^{[t]}g, k \in \Xi_i, 1 \leq i \leq M. \quad (15)$$

After processing of the string J and all related calculations (10–15) is completed, it loses its relevance, and, therefore, storing it in RAM, as well as the entire matrix J as a whole, becomes redundant. It is enough to carry out a line-by-line search in several threads to obtain an array of partial derivatives for weighting coefficients, using pairs of input and expected values, which provides significant memory savings.

In the dynamic approach, to select the regularization parameter λ_k at each iteration of the Levenberg-Marquardt method, it is proposed to use an algorithm that adaptively adjusts its value depending on the change in error at the current and previous iterations. Let ΔE_k be the change in error between the current and previous iterations, that is, $\Delta E_k = E_k - E_{k-1}$, where E_{k-1} is error function value at previous iteration $k - 1$ and E_k is the error function value at current iteration k . Then we can determine the new value of Λ_k as follows:

$$\Lambda_k = \begin{cases} \alpha \cdot \lambda_k, & \text{if } \Delta E_k > 0, \\ \beta \cdot \lambda_k, & \text{if } \Delta E_k < 0, \\ \lambda_k, & \text{if } \Delta E_k = 0, \end{cases} \quad (16)$$

where α and β – are the coefficients that control the increase or decrease in the value of Λ_k in the event of a change in error. For example, $\alpha > 1$, $0 < \beta < 1$. This approach allows the regularization parameter Λ_k to be adaptively changed depending on the direction of error change at each iteration. This can help speed up the convergence of the method and improve its efficiency.

An efficient choice of the initial value of the regularization parameter λ_k in the Levenberg-Marquardt method can be essential to ensure fast convergence and avoid potential problems such as overfitting or underconvergence. Mathematically, it can be done this way:

1. The analytical method is based on analytically estimating the initial value of the regularization parameter. Also, it takes into account the characteristics of the error function and gradient. For example, if the error function has large values, the initial value of λ_k can be chosen relatively large to ensure the stability of the algorithm. If the error function has small values, the initial value of λ_k can be chosen relatively small to allow the algorithm to converge quickly.
2. Heuristic method – heuristics or rules of thumb can be used to select the initial value λ_k . For example, the initial value of λ_k may be chosen to be a small fixed value based on an assumption of typical parameter scales or error function values.
3. Optimization method – optimization methods can also be used to select the optimal initial value of λ_k that minimizes the error function at the initial stage. For example, one can use grid search or optimization methods such as gradient descent to select the initial value of λ_k in such a way as to minimize the error function.

Choosing an effective initial value for the regularization parameter λ_k can significantly impact the Levenberg-Marquardt method performance. So, it is important to pay attention to this and apply appropriate mathematical techniques to ensure the optimal choice.

Since an error function sometimes contains several local minima or has different scales of change, it is advisable to use the multiscale optimization method to adaptively adjust the step in different parameter directions. At each iteration of the optimization algorithm, a base step size η is selected and used to update the parameters based on an error function gradient. For each parameter u_i , its characteristic scale of change Δu_i is calculated, for example, as the standard deviation or scale of a change in the parameter at previous iterations. The step size η is adaptively

adjusted for each parameter u_i by its characteristic scale of change, allowing us to take into account scale differences between parameters and adapt the step size for each parameter:

$$\eta_i = \frac{\eta}{\Delta u_i}. \quad (17)$$

These parameters are updated using the adapted step size η_i :

$$u_i^{(k+1)} = u_i^{(k)} - \eta_i \frac{\partial E}{\partial u_i}. \quad (18)$$

This process allows the step size to be adaptively varied in different parameter directions, which can improve the convergence of the optimization algorithm and help avoid getting stuck in local minima or jumps in the error function due to large-scale parameter differences. Taking into account the above, (5) is rewritten as:

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \left([{}^t]H + \Lambda_k I \right)^{-1} J^T \Delta \mathbf{y} + \eta_i \frac{\partial E}{\partial u_i}. \quad (19)$$

Expression (19) is the modified Levenberg-Marquardt method. The Levenberg-Marquardt algorithm adjusts the neural network weights using a quadratic approximation of the error surface. This approximation ensures that the minimum is quickly found, but the risk of hitting a local extremum on the training surface increases.

4. Experiment

To conduct a computational experiment, the TV3-117 TE was selected as the research object. The helicopter TE model parameters include atmospheric parameters (ρ – is the air density, P_N – is the pressure, T_N – is the temperature, and h – is the flight altitude). The helicopter onboard parameters (T_G – is the gas temperature in compressor turbine front, n_{FT} – is free turbine rotor speed, n_{TC} – is the gas generator rotor r.p.m) are normalized to absolute values (Table 4) based on V. Avgustinovich theory [51, 52].

Table 4
The training set part (author's research [51, 52])

Number	n_{FT}	n_{TC}	T_G
1	0.943	0.929	0.932
2	0.982	0.933	0.964
3	0.962	0.952	0.917
4	0.987	0.988	0.908
5	0.972	0.991	0.899
...	...	...	...
256	0.981	0.973	0.953

In complex dynamic object identification problems, error surfaces with numerous plateaus and valleys are often encountered, which makes the local minima task one of the main difficulties in achieving maximum efficiency. To overcome this problem, two heuristic approaches were developed and tested to prevent the search process from becoming trapped in local minima [53].

According to [53], the first heuristic requires the algorithm to take *risky* steps in a random direction along the error surface with increasing step length, to local minimum *jump out*. After several unsuccessful attempts, in this case, 4, the minimum found is considered the smallest, and the algorithm completes its work by the rules of the original algorithm. This heuristic was tested on 4 data sets, including 64 input numerical variables and 1 output variable. For each data set, the optimal neural network architecture is determined, including the number of neurons in the hidden layer, while minimizing the error. This architecture was identified by an exhaustive search of options, varying from 1 to 20 neurons in the nonlinear layer and from 1 to 10 neurons in the linear layer. After this, similarly [53], 50 test runs were carried out. Each of these included 10 neural network training with different initialization of weights for each data set, without using heuristics. At the final stage, another 50 test runs were performed, within each of which 10 networks were trained with different initialization of weights for each data set, using the heuristics proposed in [53]. The results of testing the neural network are shown in Table 5.

Table 5**Neural network testing results (author's research, based on [53])**

Data set number	Standard deviation of traditional Levenberg-Marquardt method			The standard deviation of the modified Levenberg-Marquardt method		
	Minimum	Maximum	Average	Minimum	Maximum	Average
Data set 1	6.95	18.38	12.67	2.84	16.12	9.48
Data set 2	7.30	22.42	14.86	5.98	19.76	12.87
Data set 3	0.22	0.64	0.43	0.14	0.20	0.17
Data set 4	4.54	7.12	5.83	3.68	5.22	4.45

Thus, the use of heuristics increases the likelihood of finding successful solutions. However, it is worth noting that in some cases this can lead to a loss of a previously found optimal solution, followed by *getting stuck* in a local minimum discovered later. This disadvantage can be surmounted by augmenting the random step number, which, in turn, requires additional computing resources [53]. The recommended 4 attempts provide the optimal balance for our subject area. According to [53], the second heuristic is to change the neural network weights when a local minimum is reached, calculating the weighting coefficients using the formula:

$$w_{ij} = w_{ij} + \theta \cdot w_{ij}, \quad (20)$$

where $\theta_1 \leq \theta \leq \theta_2$ – random number. According to [53], the θ_1 and θ_2 values are chosen through empirical means as the outcome of an experiment. Various values were tested in the range $[-0.5; 0.5]$ with steps of 0.05 [54]. As a result, the following optimal values were obtained: $\theta_1 = -0.035$, $\theta_2 = 0.035$.

Thus, when a local minimum is reached, the weights of the neural network undergo random changes by an amount not exceeding θ in both directions. Large values of θ often reset the previous training phase, requiring a neural network to recover and start training again. This, in essence, leads to the fact that the heuristic goes into the mode of a series of tests with different initial weights. Small values of θ may prevent the network from exiting the local minimum. And it will cycle back to an original minimum without benefiting from the heuristic. After *shaking* the scales, training continues according to the rules of the original algorithm. If the new minimum is not reached, a random change is made again. Through experimentation, it was revealed that the optimal number of random changes should not exceed three. To evaluate the effectiveness of the 2nd heuristic, the same data sets were used as the first. The optimal neural network architectures found at the previous stage were also used. Similarly, 50 test runs were carried out: 10 trainings of neural networks with different initialization of weights for each data set, using the second heuristic. The results of testing the neural network are shown in Table 6.

Table 6**Neural network testing results (author's research, based on [53])**

Data set number	Standard deviation of traditional Levenberg-Marquardt method			The standard deviation of the modified Levenberg-Marquardt method		
	Minimum	Maximum	Average	Minimum	Maximum	Average
Data set 1	6.95	18.38	12.67	2.84	16.12	9.48
Data set 2	7.30	22.42	14.86	5.98	19.76	12.87
Data set 3	0.22	0.64	0.43	0.14	0.20	0.17
Data set 4	4.54	7.12	5.83	3.68	5.22	4.45

Thus, the second heuristic, like the first, significantly increases the probability of detecting a global minimum. But it has the same drawback as the first – in some instances, it may result in the loss of a previously found optimal solution. In addition, its use requires an empirical selection of boundaries for changing the parameter θ .

5. Results

Labview created specialized software that uses a modified Levenberg-Marquardt algorithm and automates the recording of experimental results (Fig. 4, 5). The program implements the following single-threaded and parallel options: the traditional Levenberg-Marquardt method with the calculation of J using the central difference derivative formula; the modified Levenberg-Marquardt method with the calculation of J by the Broyden method once every two epochs with the two heuristic approaches described above; the traditional Levenberg-Marquardt method with the calculation of J using analytically derived formulas.

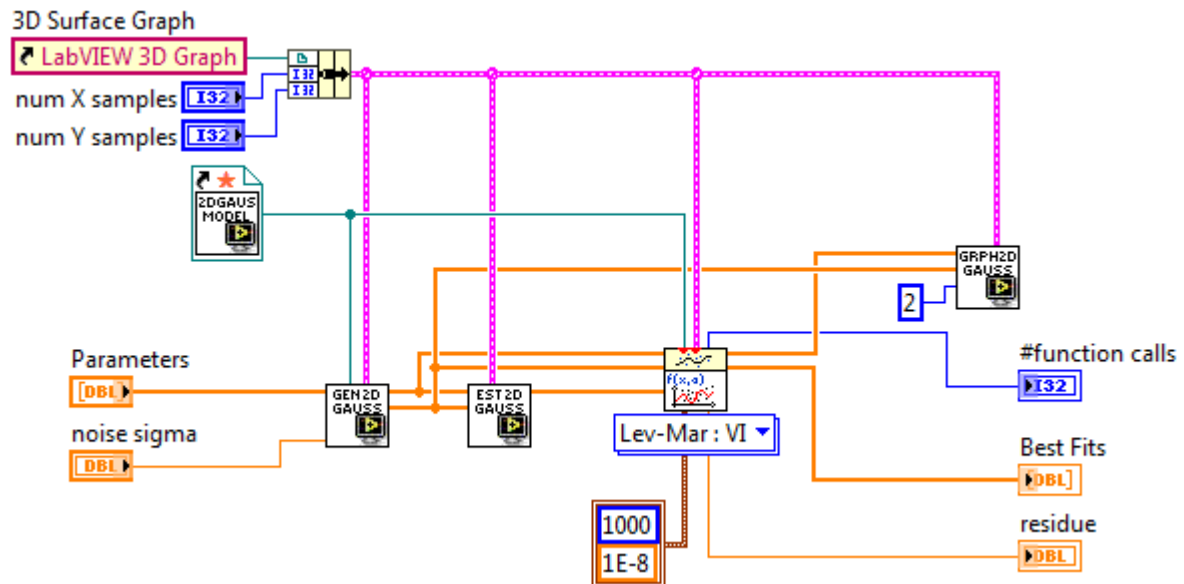


Figure 4: Diagram of complex dynamic objects neural network model (author's research)

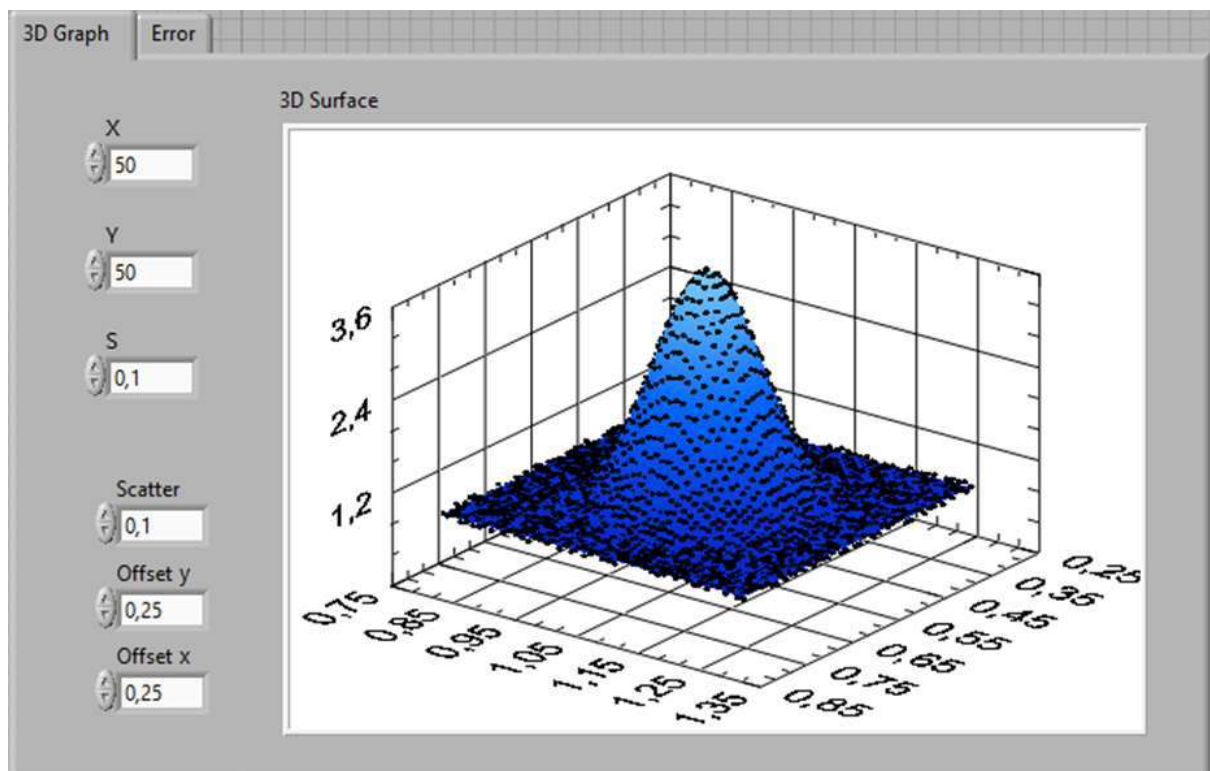


Figure 5: Developed software user interface (author's research)

For testing, we used a personal computer running the GNU/Linux operating system with an AMD Ryzen 5 5600 processor. It has 6 cores with 12 threads operating at a frequency of 3.3 GHz and 32 GB RAM for DDR-4. The computational experiment aims to assess the execution time of each method, considering the proposed methods for calculating the Jacobian matrix. For training, a NARX hybrid network (Fig. 2) with 7 inputs, in linear layer 5 neurons, in nonlinear layer 20 neurons, and in output layer 1 neuron is used [40]. The instantaneous functional $\mathbf{u}_{k+1}$ (19) is used to assess training according quality to [48]. The calculation results are presented in Table 7.

Table 7

Neural network training results (author's research, based on [42])

Data set number	Central difference		Broyden method		Analytically derived	
	t , seconds	$\mathbf{u}_{k+1}$	t , seconds	$\mathbf{u}_{k+1}$	t , seconds	$\mathbf{u}_{k+1}$
1 data stream, $N = 256$ (total training sample size [45, 46])						
Data set 1	121.382	0.988	11.785	0.985	58.639	0.983
Data set 2	121.371	0.976	11.559	0.973	58.072	0.971
Data set 3	120.989	0.995	10.082	0.992	56.537	0.990
Data set 4	121.295	0.969	11.776	0.966	57.759	0.964
6 data streams, $N = 256$ (total training sample size [45, 46])						
Data set 1	40.193	0.988	3.902	0.985	19.417	0.983
Data set 2	40.323	0.976	3.840	0.973	19.250	0.971
Data set 3	39.669	0.995	3.306	0.992	18.281	0.990
Data set 4	40.298	0.969	3.912	0.966	19.190	0.964
12 data streams, $N = 256$ (total training sample size [45, 46])						
Data set 1	20.298	0.988	1.971	0.985	9.806	0.983
Data set 2	20.262	0.976	1.930	0.973	9.695	0.971
Data set 3	20.098	0.995	1.675	0.992	9.262	0.990
Data set 4	20.216	0.969	1.963	0.966	9.627	0.964

Based on the results of the computational experiment, we can state the following:

1. In hidden layer neurons number increases and the number of steps is limited, the value of $E(w)$ increases and more steps are required to correct the parameters.
2. Using the Broyden method, it was possible to reduce the computation time by approximately 10...12 times compared to the central difference derivative, but $\mathbf{u}_{k+1}$ increased.
3. Direct calculations made it possible to reduce the calculation time by approximately 2.07...2.14 times compared to the central difference derivative. With a small training sample size, direct calculations usually yield the minimum $\mathbf{u}_{k+1}$.
4. Parallel versions of methods work on average about 3...6 times faster than the sequential implementation.
5. $\mathbf{u}_{k+1}$ almost does not change when the number of threads changes, which indicates the correct implementation of parallel processing. The maximum acceleration is approximately 61.38...120.71 times.

The next stage of the computational experiment is devoted to obtaining and analyzing the error in the operation of the trained neural network in the created software product on the identified parameters. Using the training sample (Table 4), identification errors were obtained for the following parameters of the TV3-117 TE: increase degree at compressor pressure (Fig. 6, top left), compressor turbine shaft power (Fig. 6, top right), compressor turbine operation (Fig. 6, bottom left), in combustion chamber fuel consumption (Fig. 6, bottom right), where *yellow line* is error obtained by the NARX hybrid neural network with the classical Levenberg-Marquardt method [40], *red line* is error obtained by the NARX hybrid neural network with the modified Levenberg-Marquardt method, *green line* is approximation line. Analysis of the results of processing the average error of training the hybrid neural network NARX using the modified Levenberg-Marquardt method showed a decrease in the average error value by 33 %, to a level of 0.025.

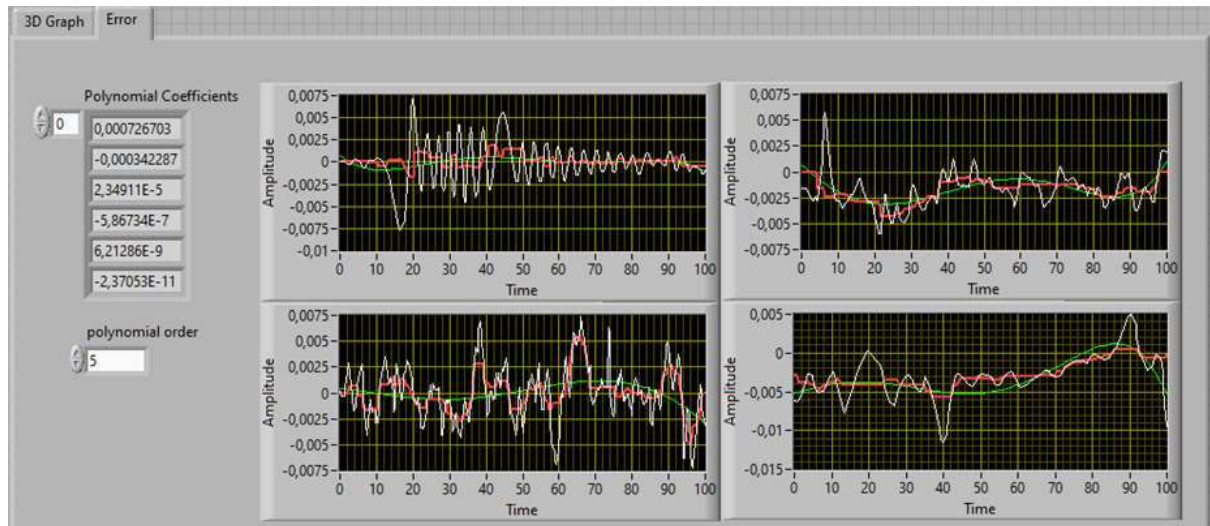


Figure 6: Neural network training error calculating results (author's research)

At the final stage of the computational experiment, the loss was calculated and analyzed in the hybrid neural network NARX (Fig. 7), which serves as an indicator of the variance between the model's predicted values and target variable during training. Loss reflects the degree of error in the model of the research object and is used in the training process to adjust the neural network parameters for minimize this error. The expression for calculating the loss L in a neural network usually depends on the type of problem (such as regression or classification) and the loss function used. For regression problems, the squared error is often used for classification tasks, cross-entropy, or other loss functions. The general formula can be represented as the loss sum over all examples of the training set. In this work, used [54, 55]:

$$L = \frac{1}{n} \cdot \sum_{i=1}^n w_i \cdot (y_i - \tilde{y}_i)^2, \quad (21)$$

where n is examples number, y_i is actual value of the target variable, $\tilde{y}_i$ is predicted value by the model, w_i is the weight assigned to each error, allowing their significance to be taken into account in the final loss function.

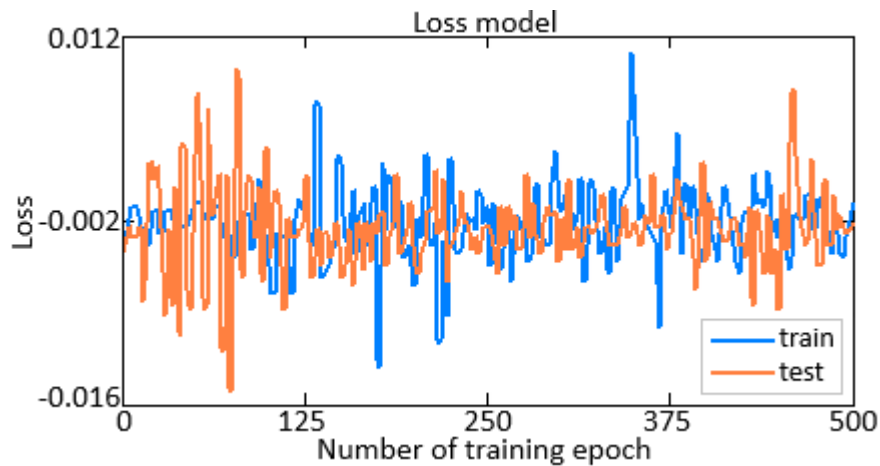


Figure 7: Loss rates during model training and testing on the input dataset (according to Table 4): blue curve is train; orange curve is validation (author's research)

6. Discussions

Fig. 7 shows that the loss function of the hybrid neural network NARX over 500 training epochs is generally stable and does not go beyond the limit of 0.025 (2.5 %), which indicates acceptable losses in problems of identifying complex dynamic objects [56, 57].

Table 8 contains comparative analysis results for thermogas-dynamic parameters identification accuracy in the engine operating process using a neural network and classical methods for each parameter in TV3-117 TE model [40].

Table 8

Absolute error calculation results (author's research, comparisons with [40])

Model	Absolute error, %			
	Fuel consumption in combustion chamber	Compressor turbine operation	Compressor turbine shaft power	Increase degree dependence on compressor pressure
Classical	1.95	1.95	1.96	1.95
Neural network:				
three-layer perceptron	0.65	0.68	0.64	0.66
[49, 50]	0.41	0.43	0.41	0.42
Gaussian NARX-model				
[34]	0.26	0.28	0.26	0.27
Gaussian NARX-model with modified Levenberg-Marquardt method				

We introduced supplementary noise to the dataset to assess neural networks resilience to variations in input information (Table 4). This noise was incorporated into each parameter by integrating white noise with a standard deviation of $\sigma_i = 0.025$ and a mean of zero. For each parameter this corresponds to maximum value 2.5 %. Table 9 illustrates the outcomes of a comparative assessment of the precision in implementing the technique for discerning thermogas-dynamic parameters in TV3-117 TE operating process using both neural networks and traditional approaches.

Table 9

Absolute error (with white noise) calculation results (author's research, comparisons with [40])

Model	Absolute error, %			
	Increase degree dependence on compressor pressure	Compressor turbine shaft power	Compressor turbine operation	Fuel consumption in combustion chamber
Classical	3.11	3.15	3.14	3.15
Neural network:				
three-layer perceptron [55, 56]	1.13	1.09	1.17	1.11
Gaussian NARX-model [40]	0.74	0.72	0.73	0.71
Gaussian NARX-model with modified Levenberg-Marquardt method	0.43	0.41	0.42	0.40

An examination of Table 9 demonstrates that under the stated noise conditions, the error in identification remains within specific limits: for Gaussian NARX model with modified Levenberg-Marquardt method is 0.43 %, for Gaussian NARX model is 0.71 %, for three-layer perceptron structured as 7-53-36 is 1.09 % [40, 58, 59], and for thermogas-dynamic TV3-117 TE model is 3.15 %. Due to the maximum absolute error and white noise presence in applying the identification technique for the thermogas-dynamic parameters using the least squares method rose from 1.96 % to 3.15 %. This error increased from 0.64 % to 1.09 % [40, 58, 59] for the three-layer perceptron structured as 7-53-36, , from 0.28 % to 0.43 % for the gaussian NARX model with modified Levenberg-Marquardt method, and it went up from 0.43 % to 0.74 % for Gaussian NARX model. To evaluate the dependability of the neural network approach in discerning the thermogas-dynamic parameters of TV3-117 TE operating process [40, 58, 59], the following formulations can be employed [60, 61]:

$$K_{error} = \frac{T_{error}}{T_0} \cdot 100\%,$$

$$K_{quality} = \left(1 - \frac{T_{error}}{T_0}\right) \cdot 100\%,$$
(22)

where K_{error} and $K_{quality}$ represent coefficients for erroneous and quality identification, respectively [62]; T_{error} indicates the cumulative time of segments associated with misclassification, while T_0 denotes test sample duration (in this context, $T_0 = 5$ s) [63].

Table 10 presents coefficients computing outcomes for parameters quality identification and both erroneous [40, 60–64], including the relations between the increase degree dependence in compressor turbine operation, compressor turbine shaft power, the total compressor pressure, and fuel consumption in the combustion chamber.

Table 10

Erroneous and qualitative coefficients calculating results (author's research, comparisons with [40])

Parameter	Gaussian NARX-model [40]		Gaussian NARX-model with modified Levenberg-Marquardt method	
	K_{error}	$K_{quality}$	K_{error}	$K_{quality}$
Compressor turbine operation	0.528	99.873	0.393	99.923
Compressor turbine shaft power	0.523	99.871	0.389	99.921
Increase degree dependence on compressor pressure	0.521	99.872	0.386	99.925
Fuel consumption in combustion chamber	0.526	99.872	0.390	99.922

As depicted in Table 10, the rates of erroneous identification coefficients remain below 0.393 %, while the minimum coefficients for accurate identification rate reach 99.925 %.

The main area of practical application of the developed method is the helicopter TE monitoring and operation controlling neural network on-board expert system [65]. The developed method can be included as a neural network module for helicopter TE parameters identification, which provides continuous and engine operation accurate monitoring in real-time, and also increases the level of safety and flight efficiency.

7. Conclusions

For the first time, a concept has been created for constructing neural network models of complex dynamic objects, in which, by increasing the robustness of the functioning of a trained neural network, it becomes possible at its output to increase the reliability of solving problems of identifying complex dynamic objects.

The universal neural network model of complex dynamic objects was further developed in the form of a hybrid neural network NARX with a radial-basis layer, in which, through the use of a modified Levenberg-Marquardt training method, a reduction in the maximum absolute identification error by almost 2 times is achieved – from 0.74 to 0.43 %.

The transition from linear neural networks (multilayer perceptron) to nonlinear ones (hybrid neural network NARX with a radial basis layer) in identification tasks of complex dynamic objects is scientifically substantiated, providing more accurate and flexible identification of parameters of complex dynamic objects in real-time.

The method of calculating the elements of the Hessian matrix, a component of the analytical expression of the Levenberg-Marquardt method, was further developed, which, by taking into account the weight connections of neurons of both nonlinear and linear layers, allowed for to reduction of the calculation time by approximately 10...12 times compared to the central difference derivative. The use of direct calculations made it possible to reduce the calculation time by approximately 2.07...2.14 times compared to the central difference derivative.

For the first time, an analytical description of the regularization parameter has been proposed, which is based on control coefficients of increasing or decreasing its value in the event of a change

in error, in the mathematical expression of the Levenberg-Marquardt method, which significantly affects its performance.

The proposed complex modification of the Levenberg-Marquardt method made it possible to experimentally select the optimal structure of the neural network, reduce the average value of the training error of a NARX hybrid neural network with a radial basis layer by 33 % for the level of 0.025, and also ensure the stability of the loss function of the neural network throughout 500 training epochs, which does not exceed 2.5 %.

References

- [1] R. Voliansky, A. Pranolo, Parallel mathematical models of dynamic objects, *International Journal of Advances in Intelligent Informatics* 4:2 (2018) 120–131. doi: 10.26555/ijain.v4i2.229.
- [2] V. Sherstjuk, M. Zharikova, I. Didmanidze, I. Dorovskaja, S. Vyshemyrska, Risk modeling during complex dynamic system evolution using abstract event network model, *CEUR Workshop Proceedings* 3101 (2022) 93–110.
- [3] A. Sharma, E. Kosasih, J. Zhang, A. Brintrup, A. Calinescu, Digital Twins: State of the art theory and practice, challenges, and open research questions, *Journal of Industrial Information Integration* 30 (2022) 100383. doi: 10.1016/j.jii.2022.100383.
- [4] M. Fore, M. O. Alver, J. A. Alfredsen, A. Rasheed, T. Hukkelas, H. V. Bjelland, B. Su, S. J. Ohrem, E. Kelasidi, T. Norton, N. Papandroulakis, Digital Twins in intensive aquaculture – Challenges, opportunities and future prospects, *Computers and Electronics in Agriculture* 218 (2024) 108676. doi: 10.1016/j.compag.2024.108676.
- [5] A. Becue, E. Maia, L. Feeken, P. Borchers, I. Praca, A New Concept of Digital Twin Supporting Optimization and Resilience of Factories of the Future, *Applied Sciences* 10 (13) (2020) 4482. doi: 10.3390/app10134482.
- [6] D. Galar, U. Kumar, Digital Twins: Definition, Implementation and Applications, *Advances in Risk-Informed Technologies* (2024) 79–106.
- [7] A. Nikiforov, Automatic Control of the Structure of Dynamic Objects in High-Voltage Power Smart-Grid, *Automation and Control* (2020). doi: 10.5772/intechopen.91664. URL: <https://www.intechopen.com/chapters/71513>
- [8] O. Maksymov, E. Malakhov, V. Mezhyuev, Model and method for representing complex dynamic information objects based on LMS-trees in NoSQL databases, *Herald of Advanced Information Technology* 4:3 (2021) 211–224. doi: 10.15276/hait.03.2021.1.
- [9] D. Kahl, M. Kschischo, Searching for Errors in Models of Complex Dynamic Systems, *Frontiers in Physiology* 11 (2020). doi: 10.3389/fphys.2020.612590. URL: <https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2020.612590/full>
- [10] Y. Li, Application Analysis of Artificial Intelligent Neural Network Based on Intelligent Diagnosis, *Procedia Computer Science* 208 (2022) 31–35. doi: 10.1016/j.procs.2022.10.006.
- [11] A. Kupina, D. Zubov, Y. Osadchuka, R. Ivchenkoa, V. Saiapin, Intelligent Neural Networks Models for the Technological Separation Processes, *CEUR Workshop Proceedings* 3373 (2023) 76–86.
- [12] S. S. Talebi, A. Madadi, A. M. Tousi, M. Kiaee, Micro Gas Turbine fault detection and isolation with a combination of Artificial Neural Network and off-design performance analysis, *Engineering Applications of Artificial Intelligence*, vol. 113 (2022) 104900. doi: 10.1016/j.engappai.2022.104900.
- [13] S. Kim, J. H. Im, M. Kim, J. Kim, Y. I. Kim, Diagnostics using a physics-based engine model in aero gas turbine engine verification tests, *Aerospace Science and Technology*, vol. 133 (2023) 108102. doi: 10.1016/j.ast.2022.108102.
- [14] J. Zeng, Y. Cheng, An Ensemble Learning-Based Remaining Useful Life Prediction Method for Aircraft Turbine Engine, *IFAC-PapersOnLine*, vol. 53, issue 3 (2020) 48–53. doi: 10.1016/j.ifacol.2020.11.009.
- [15] B. Li, Y.-P. Zhao, Y.-B. Chen, Unilateral alignment transfer neural network for fault diagnosis of aircraft engine, *Aerospace Science and Technology*, vol. 118 (2021) 107031. doi: 10.1016/j.ast.2021.107031.

- [16] Y. Shen, K. Khorasani, Hybrid multi-mode machine learning-based fault diagnosis strategies with application to aircraft gas turbine engines, *Neural Networks*, vol. 130 (2020) 126–142. doi: 10.1016/j.neunet.2020.07.001.
- [17] R. Chen, X. Jin, S. Laima, Y. Huang, H. Li, Intelligent modeling of nonlinear dynamical systems by machine learning, *International Journal of Non-Linear Mechanics*, vol. 142 (2022) 103984. doi: 10.1016/j.ijnonlinmec.2022.103984.
- [18] M. Soleimani, F. Campean, D. Neagu, Diagnostics and prognostics for complex systems: A review of methods and challenges, *Quality and Reliability Engineering*, vol. 37, issue 8 (2021) 3746–3778 doi: 10.1002/qre.2947.
- [19] Z. Huang, Automatic Intelligent Control System Based on Intelligent Control Algorithm, *Journal of Electrical and Computer Engineering*, vol. 7 (2022) 1–10. doi: 10.1155/2022/3594256
- [20] S. Tian, J. Zhang, X. Shu, L. Chen, X. Niu, Y. Wang, A Novel Evaluation Strategy to Artificial Neural Network Model Based on Bionics, *Journal of Bionic Engineering*, 19 (1) (2022) 224–239. doi: 10.1007/s42235-021-00136-2.
- [21] L. Qian, C. Liu, J. Yi, S. Liu, Application of hybrid algorithm of bionic heuristic and machine learning in nonlinear sequence, *Journal of Physics: Conference Series* 1682 (2020) 012009. doi: 10.1088/1742-6596/1682/1/012009.
- [22] H. Lin, C. Wang, J. Sun, X. Zhang, Y. Sun, H. H. C. Iu, Memristor-coupled asymmetric neural networks: Bionic modeling, chaotic dynamics analysis and encryption application, *Chaos, Solitons & Fractals* 166 (2023) 112905. doi: 10.1016/j.chaos.2022.112905
- [23] J. Sun, S. Sathasivam, M. Khan, Analysis and Optimization of Network Properties for Bionic Topology Hopfield Neural Network Using Gaussian-Distributed Small-World Rewiring Method, *IEEE Access* 10 (2022) 95369–95389. doi: 10.1109/ACCESS.2022.3204821.
- [24] S. Vladov, Y. Shmelov, R. Yakovliev, Optimization of Helicopters Aircraft Engine Working Process Using Neural Networks Technologies, *CEUR Workshop Proceedings* 3171 (2022) 1639–1656.
- [25] R. Abdulkadirov, P. Lyakhov, N. Nagornov, Survey of Optimization Algorithms in Modern Neural Networks, *Mathematics* 11 (11) (2023) 2466. doi: 10.3390/math11112466
- [26] J. Chen, Y. Liu, Neural optimization machine: a neural network approach for optimization and its application in additive manufacturing with physics-guided learning, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 381: 2260 (2023). doi: 10.1098/rsta.2022.0405. URL: <https://royalsocietypublishing.org/doi/10.1098/rsta.2022.0405>
- [27] F. Mehmood, S. Ahmad, T. K. Whangbo, An Efficient Optimization Technique for Training Deep Neural Networks, *Mathematics* 11 (6) (2023) 1360. doi: doi.org/10.3390/math11061360
- [28] T. Asrav, E. Aydin, Physics-informed recurrent neural networks and hyper-parameter optimization for dynamic process systems, *Computers & Chemical Engineering* 173 (2023), 108195. doi: 10.1016/j.compchemeng.2023.108195
- [29] A. Merabet, S. Kanukollu, A. Al-Durra, E. F. El-Saadany, Adaptive recurrent neural network for uncertainties estimation in feedback control system, *Journal of Automation and Intelligence* 2:3 (2023), 119–129. doi: 10.1016/j.jai.2023.07.001
- [30] M. F. Ab Aziz, S. A Mostafa, C. F. M. Foozy, M. A. Mohammed, M. Elhoseny, A. Z. Abualkishik, Integrating Elman recurrent neural network with particle swarm optimization algorithms for an improved hybrid training of multidisciplinary datasets, *Expert Systems with Applications* 183 (2021), 115441. doi: 10.1016/j.eswa.2021.115441
- [31] Q. Ali M. N. Mahdi, M. Ali, M. N. Atta, A. Khan, S. A. Lashari, D. A. Ramli, Training Learning Weights of Elman Neural Network Using Salp Swarm Optimization Algorithm, *Procedia Computer Science* 225 (2023), 1974–1986. doi: 10.1016/j.procs.2023.10.188
- [32] Y. Li, Z. Wang, R. Han, S. Shi, J. Li, R. Shang, H. Zheng, G. Zhong, Y. Gu, Quantum recurrent neural networks for sequential learning, *Neural Networks* 166 (2023), 148–161. doi: 10.1016/j.neunet.2023.07.003
- [33] Y. Wang, W. Zhou, H. Feng, L. Li, H. Li, Progressive Recurrent Network for shadow removal, *Computer Vision and Image Understanding* 238 (2024), 103861. doi: 10.1016/j.cviu.2023.103861
- [34] H. Wang, W. Jiang, X. Deng, J. Geng, A new method for fault detection of aero-engine based on isolation forest, *Measurement* 185 (2021) 110064. doi: 10.1016/j.measurement.2021.110064

- [35] S. Zhernakov, A. Gilmanshin, New onboard gas turbine engine diagnostic algorithms based on neural-fuzzy networks, *Aviation and rocket and space technology* 19: 2 (68) (2015) 63–68.
- [36] S. Zhernakov, A. Gilmanshin, Realization of hybrid gas turbine engine control and diagnostics algorithms using modern on-board computing devices, in: *Proceedings of the VII International conference “Actual problems of mechanical engineering”*, March 25–27, 2015, pp. 765–769.
- [37] B. Mokin, V. Mokin, O. Mokin, O. Mamyrbayev, S. Smailova, The synthesis of mathematical models of nonlinear dynamic systems using Volterra integral equation, *Informatyka, Automatyka, Pomiar w Gospodarce i Ochronie Środowiska* 12:2 (2022) 15–19. doi: 10.35784/iapgos.2947.
- [38] A. Slipchuk, P. Pukach, M. Vovk, O. Slyusarchuk, Study of the dynamic process in a nonlinear mathematical model of the transverse oscillations of a moving beam under perturbed boundary conditions, *Mathematical Modeling and Computing* 11:1 (2024) 37–49. doi: 10.23939/mmc2024.01.037
- [39] A. Abdunagimov, G. Ageev, Neural network technologies in hardware-in-the-loop simulation: principles of gas turbine digital twin development, *Informatics, Computer Science and Management* 23: 4 (86) (2019) 115–121.
- [40] S. Vladov, R. Yakovliev, O. Hubachov, J. Rud, Y. Stushchanskyi, Neural Network Modeling of Helicopters Turboshift Engines at Flight Modes Using an Approach Based on “Black Box” Models, *CEUR Workshop Proceedings* 3624 (2024) 116–135.
- [41] S. Vladov, I. Dieriabina, O. Husarova, L. Pylypenko, A. Ponomarenko, Multi-mode model identification of helicopters aircraft engines in flight modes using a modified gradient algorithms for training radial-basis neural networks, *Visnyk of Kherson National Technical University* 4 (79) (2021) 52–63. doi: 10.35546/kntu2078-4481.2021.4.7
- [42] G. Alcan, M. Unel, V. Aran, M. Yilmaz, C. Gurel, K. Koprubasi, Diesel Engine NOx Emission Modeling Using a New Experiment Design and Reduced Set of Regressors, *IFAC-PapersOnLine* 51:15 (2018) 168–173. doi: 10.1016/j.ifacol.2018.09.114
- [43] S. Vladov, Y. Shmelov, R. Yakovliev, Modified Searchless Method for Identification of Helicopters Turboshift Engines at Flight Modes Using Neural Networks, in: *Proceedings of the 2022 IEEE 3rd KhPI Week on Advanced Technology*, Kharkiv, Ukraine, October 03–07, 2022, pp. 257–262. doi: 10.1109/KhPIWeek57572.2022.9916422
- [44] S. Novikova, E. Kremleva, Increasing the robustness of the neural network model for monitoring gas turbine engines based on reduction, *Aircraft, aircraft engines and methods of their operation* 3 (2019) 17–26.
- [45] J. Bill, B. A. Cox, L. Champagne, A comparison of quaternion neural network backpropagation algorithms, *Expert Systems with Applications* 232 (2023) 120448. doi: 10.1016/j.eswa.2023.120448
- [46] G. Xing, J. Gu, X. Xiao, Convergence analysis of a subsampled Levenberg-Marquardt algorithm, *Operations Research Letters* 51:4 (2023) 379–384. doi: 10.1016/j.orl.2023.05.005
- [47] S. Parkhomenko, A Levenberg-Marquardt algorithm execution time reducing in case of large amount of the data, *International Research Journal* 1 (20) part 1 (2014) 80–83.
- [48] S. Parkhomenko, T. Ledeneva, Training neural networks using the Levenberg-Marquardt method in conditions of a large amount of data, *System analysis and information technology* 2 (2014) 98–106.
- [49] N. Marumo, T. Okuno, A. Takeda, Majorization-minimization-based Levenberg–Marquardt method for constrained nonlinear least squares, *Computational Optimization and Applications* 84 (2023) 833–874. doi: 10.1007/s10589-022-00447-y.
- [50] A. O. Umar, I. M. Sulaiman, M. Mamat, M. Y. Waziri, N. Zamri, On damping parameters of Levenberg-Marquardt algorithm for nonlinear least square problems, *Journal of Physics: Conference Series* 1734 (2021) 012018. doi: 10.1088/1742-6596/1734/1/012018
- [51] S. Vladov, Y. Shmelov, R. Yakovliev, Modified Helicopters Turboshift Engines Neural Network On-board Automatic Control System Using the Adaptive Control Method, *CEUR Workshop Proceedings* 3309 (2022) 205–224.
- [52] S. Vladov, Y. Shmelov, R. Yakovliev, M. Petchenko, Modified Neural Network Fault-Tolerant Closed Onboard Helicopters Turboshift Engines Automatic Control System, *CEUR Workshop Proceedings*, vol. 3387 (2023) 160–179.

- [53] K. Makhotilo, D. Voronenko, Modification of the Levenberg-Marquardt algorithm to improve the accuracy of predictive models of connected energy consumption in everyday life, *Bulletin of the National Technical University "KhPI" A series of "Information and Modeling"* 56 (2005) 83–90.
- [54] W. Zonghui, H. Jian, S. Xiaodan, Study on Robust Loss Function for Artificial Neural Networks Models in Reliability Analysis, *Procedia Structural Integrity* 52 (2024), 203–213. doi: 10.1016/j.prostr.2023.12.021
- [55] S. Zhang, L. Xie, Leader learning loss function in neural network classification, *Neurocomputing* 557 (2023), 126735. doi: 10.1016/j.neucom.2023.126735
- [56] S. Vladov, Y. Shmelov, R. Yakovliev, Helicopters Aircraft Engines Self-Organizing Neural Network Automatic Control System, *CEUR Workshop Proceedings* 3137 (2022) 28–47. doi: 10.32782/cmris/3137-3
- [57] Y. Wang, H. Li, Y. Zheng, J. Peng, A fractional-order visual neural network for collision sensing in noisy and dynamic scenes, *Applied Soft Computing* 148 (2023), 110897. doi: 10.1016/j.asoc.2023.110897
- [58] S. Vladov, Y. Shmelov, R. Yakovliev, M. Petchenko, S. Drozdova, Neural Network Method for Helicopters Turboshift Engines Working Process Parameters Identification at Flight Modes, in: *Proceedings of the 2022 IEEE 4th International Conference on Modern Electrical and Energy System (MEES)*, Kremenchuk, Ukraine, 2022, pp. 604–609. doi: 10.1109/MEES58014.2022.10005670
- [59] S. Vladov, Y. Shmelov, R. Yakovliev, M. Petchenko, S. Drozdova, Helicopters Turboshift Engines Parameters Identification at Flight Modes Using Neural Networks, in: *Proceedings of the IEEE 17th International Conference on Computer Science and Information Technologies (CSIT)*, Lviv, Ukraine, 2022, pp. 5–8. doi: 10.1109/CSIT56902.2022.10000444
- [60] S. Marton, Stefan Ludtke, C. Bartelt, Explanations for Neural Networks by Neural Networks, *Applied Sciences* 12(3) (2022), 980. doi: 10.3390/app12030980
- [61] C. Yuan, J. Y. Wang, C. E. Lee, K.-N. Chiang, Equation Informed Neural Networks with Bayesian Inference Improvement for the Coefficient Extraction of the Empirical Formulas, in: *Proceedings of the 2023 24th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*, Graz, Austria, 2023. doi: 10.1109/EuroSimE56861.2023.10100752
- [62] F. Munoz, J. M. Valdovinos, J. S. Cervantes-Rojas, S. S. Cruz, A. M. Santana, Leader-follower consensus control for a class of nonlinear multi-agent systems using dynamical neural networks, *Neurocomputing* 561 (2023), 126888. doi: 10.1016/j.neucom.2023.126888
- [63] V. Makarov, The neural network to identify an object by a sequential training mode, *Procedia Computer Science* 190 (2021), 532–539. doi: 10.1016/j.procs.2021.06.062
- [64] H. Taherdoost, Deep Learning and Neural Networks: Decision-Making Implications, *Symmetry* 15(9) (2023), 1723. doi: 10.3390/sym15091723
- [65] Y. Shmelov, S. Vladov, Y. Klimova, M. Kirukhina, 60. Expert system for identification of the technical state of the aircraft engine TV3-117 in flight modes, in: *Proceedings of the System Analysis & Intelligent Computing : IEEE First International Conference on System Analysis & Intelligent Computing (SAIC)*, Kyiv, Ukraine, 2018, pp. 77–82. doi: 10.1109/SAIC.2018.8516864

The Robustness of the Edge Detection on Noisy Natural Images with HED and PiDiNet Networks

Marina Polyakova¹, Serhii Khyrenko¹

¹ Odesa Polytechnic National University, 1, Shevchenko Ave., Odesa, 65044, Ukraine

Abstract

The edge detection performance of the HED and PiDiNet networks is compared on noised natural images. To research the robustness of edge detection with HED and PiDiNet networks, the technique of detecting the influence of the noise level on the CNN edge detector efficiency is proposed. At first, as a result of literature review, HED and PiDiNet are selected to detect edges on noisy images. Then types of image noise of interest to the researcher and the noise parameters which will be controlled by the researcher are determined. We considered white Gaussian noise, impulse noise, and speckle noise. Mathematical modeling of noisy images is performed. Next, the training and test sets from image datasets for edge detection is selected. After that, the researched networks are learned to detect edges on training set of images or weights of pre-trained networks are loaded. Further the measures of edge detection performance are selected, specifically, Precision, Recall, F1-score, and Pratt's Figure of Merit. Then images of test set are corrupted with controlled level of noise. The trained networks are applied to noisy images and the edge detection performance is evaluated depending on the noise level in the images. The obtained results are analyzed to generate the recommendations allowing to determine which CNN is better for edge detection when natural images affected by different level of white Gaussian noise, or impulse noise, or speckle noise. In particular, the HED network is generally preferred at high noise levels. The PiDiNet network is best used when the noise level is low.

Keywords

Edge detection, noisy image, HED network, PiDiNet network

1. Introduction

The natural image edge detection is implemented in intelligent systems and computer vision to search images in databases and the Internet, to monitor the transport, infrastructure, and environment. The continuous growth of the computing power allows implementing the image edge detection even in mobile devices. Natural images are distinguished by smooth color change areas, a low level of noise, and texture regions. Then to detect object edges on natural images it is necessary to ignore the noisy pixels and background texture to establish a correspondence between boundaries of objects and the color differences [1, 2].

The object of research is the natural image edge detection in computer graphics and intelligent systems.

Much better results compared to conventional edge detection methods can be obtained by using convolutional neural networks (CNNs). CNNs automatically learn to identify features of images at different levels of abstraction. For example, the initial layers of the network can detect an edges and boundaries of texture regions, while the deeper layers concentrate on combining the obtained edges into the boundaries of more complex structures, such as objects. An important aspect of the use of deep learning artificial intelligence is their ability to fine-tune for solving similar tasks after preliminary training. This makes it possible to process images of different complexity and content by the same CNN [3, 4].

The key factors determining the edge detection performance are the level of detailing, that is, how many different objects are visible in the image, and the level of image noise. The first factor is regulated by CNN architecture and computing power of research devices. The second factor, the level of noise in the image, can still be a challenge and affect the result. Different CNNs have different edge detection performance on the same image, but with different noise levels.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ marinapolyakova943@gmail.com (M. Polyakova); sergeyhyrenko@gmail.com (S. Khyrenko)

ORCID 0000-0001-7229-7657 (M. Polyakova); 0009-0003-2353-6617 (S. Khyrenko)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

It should be noted that image noise can significantly complicate the edge detection and affect the quality of the obtained results. Usually, noise can occur due to various factors, such as electromagnetic interference, equipment defects, accidental impacts during photography or filming. Noise appears in random fluctuations of the intensity of pixels, which can lead to a loss of edge quality and the appearance of false details. Images affected by Gaussian white noise, impulse or speckle noise are most often considered.

The subject of the research is the robustness of edge detection with the CNNs, namely, the dependence of edge detection performance on the level of image noise.

The Holistically-nested edge detection (HED) network was chosen for the robustness research because of its multi-layered architecture with side replies to process images with varying levels of details. The next advantage of this network is the comparison of interpolated side replies with ground truth image to understand the overall context of the original noised image. Then HED architecture is used in autopilot systems of vehicles. The robustness of edge detection by Pixel difference network (PiDiNet), which also uses context analysis around the processed pixel, is also researched. But unlike the HED network, it applies a gradient estimation operator compatible with conventional convolution.

The aim of research is to compare the robustness of HED and PiDiNet networks to detect edges on noised natural images. The comparison will help determine which network is better for edge detection when natural images affected by white Gaussian noise, impulse noise, or speckle noise.

2. Problem statement

In this paper, the natural images are selected for research. The color natural image is represented as $I(x,y)=(I_R(x,y), I_G(x,y), I_B(x,y))$, where $x=1, \dots, n$; $y=1, \dots, l$. Here n is the number of rows of the image; l is the number of columns of the image; (x,y) are coordinates of the image pixel; $I(x,y)$ is a vector function representing an image by color channels; $I_R(x,y)$, $I_G(x,y)$, $I_B(x,y)$ are the functions of intensity of the red, green, or blue color channels respectively. To detect edges, each pixel of the original image must be associated with the value of the target feature. There is a label of one of two classes, specifically, 1 for boundary pixels, 0 for pixels inside homogeneous areas. The values of the target feature for the natural image should be represented as a binary image which is the result of edge detection [2].

There are determined the types of noise that are of interest to the researcher and inherent in natural images. The models of noise of these types are selected as $M = \{M_1, M_2, \dots, M_L\}$, where L is a number of researched noise models. One or more parameters determining the noise level in the image are selected for each model.

Let an NN is a set of CNNs of deep learning $NN = \{NN_1, NN_2, \dots, NN_K\}$, where K is a number of researched CNNs. These CNNs are designed to detect edges on natural images and trained on test database.

The problem of researching of the robustness of the edge detection with the CNNs is as follows. It is necessary to determine the dependences of the edge detection performance on the value of the noise parameter for each researched network and each considered type of noise on a selected class of images. Using the obtained dependencies, it is necessary to elaborate recommendations for the use of a specific neural network to detect edges on an image with a certain type and level of noise.

3. Review of the literature

Recent surveys of CNNs designed to detect edges considered their architectures and the performance of edge detection with these networks [1, 5]. In this paper the robustness of edge detection on noised images using CNNs is researched. In the literature, in view of the robustness the conventional edge detectors not using CNNs are divided into two groups. These are differential and correlation for prototype matching methods [2].

The differential methods apply gradient operator to each color channel of the image. Then the gradient magnitude is calculated and thresholded for each color channel. Next, the results of edge detection of the color channels are combined into edge map of original image. Otherwise, the gradient vectors of color channels are transformed using the norm operator into gray-scale image. Thresholding is performed only for intensities of this image and thus the color edges are obtained [6].

The advantages of differential methods are the high-speed performance and the low error of edge positioning. However, the robustness to noise is low because of the application of the gradient

operators. Contour breaks and false edges occur. A wide variety of differential edge detection methods is due to attempts to apply gradient operators with the greatest possible robustness to noise using linear or nonlinear processing [2, 7].

To detect edges with correlation for prototype matching methods, W prototypes of edge are compiled. Based on these prototypes, the coefficients of the W filters are selected. The result of image processing by each of these filters is determined. Let the filtering result reach a maximum for some image area. This means that the edge intensity prototype matches the image area. The matching is defined by comparison with a threshold based on statistical decision theory. If edges are detected with correlation by prototype matching then several edge models are used, since the real edge can be ramped. These methods are robust to noise, but the positioning accuracy is low. The latter is due to poor discrimination of prototypes of different shape with the same energy. In addition, the use of W filters reduces high-speed performance of correlation by prototype matching [2, 7].

The authors propose to classify edge detectors with CNN into differential and correlation by prototype matching. We classify CNN as differential method, if some gradient operators are embedded into the convolution layers. If the architecture of the CNN includes conventional convolution layers, then we define edge detector with such CNN as correlation by prototype matching method. This is because the more similar the image area is to the convolution kernel, the larger the convolution result value.

CNNs that implement correlation by prototype matching methods for edge detection appeared earlier, since the convolutional layer is based on template matching. Examples of such networks are HED [8, 9], Richer convolutional features (RCF) network [10], Learning to predict crisp boundaries (LPCB) network [11] et al.

To resolve ambiguity in edge and object boundary detection, the HED is proposed in [8]. This fully convolutional network differs from other CNNs by side replies. When training the HED network, side replies are deeply-supervised and interpolated to initial image size. The results are fused to obtain multi-scale nested features. Thus rich hierarchical representation is automatically directed by deep supervision on HED side replies [12].

To solve the thick contour problem in edge detection the HED network is enhanced in the paper [11]. The obtained fully convolutional network of bottom-up/top-down architecture is named LPCB network. It is based on VGG16 network [13] and uses a new loss function, which evaluates image similarity and very effective for classifying unbalanced data. LPCB uses fewer parameters compared to the HED network. But the LPCB network shows better edge detection performance, and produces accurate results without post-processing.

Also, in [10] the RCF network is designed for accurate edge detection. It designed by removing the fully connected layer and the fifth pooling layer from the VGG16 network and estimates multi-scale features of the image. The RCF network uses convolutional layers with different perceptual fields following by pooling layers. Then the layer level features are fused, the automatic learning of all weight parameters is done. Thus, RCF combines the underlying feature maps to detect edges based on the pyramid architecture [3].

CNNs that implement differential edge detectors were designed to solve the problems of thick image contour and inaccurate positioning. These are PiDiNet [14], PiDiNeXt [15], Xception based network with fusion difference convolutions [16] et al.

A novel pixel difference convolution is integrated in [14] into network convolutional layer. The resulting PiDiNet network can easily retain the powerful learning ability of deep CNN to extract image features with semantic significance while capturing gradient information conducive to edge detection. Then the better edge detection accuracy with fewer parameters is achieved through the integration of the gradient estimation into the convolution operation.

Although PiDiNet achieves competitive results combining traditional difference operators with deep learning, the problem of the rationale behind the selection of gradient estimation operators remained unresolved. Therefore, in [15] the PiDiNeXt was proposed, which combines gradient estimation operators with deep learning-based model in parallel to solve the problem of operator selection and further feature improving. As a result, the experiments on test datasets demonstrate that PiDiNeXt outperforms PiDiNet in terms of edge detection accuracy.

In [16] the edge detection via fusion difference convolution is proposed. This new structure integrates vanilla convolutions with pixel difference convolutions to extract more semantic information from the image, thereby reducing noise and texture impact on the resulting edges.

Besides, fusion difference convolution strengthens the diagonal edges in image features. The last results in more accurate diagonal edges compared to the RCF and PiDiNet.

As a result of the literature review, the HED network implementing the correlation by prototype matching to detect edges and the PiDiNet network implementing the differential method were selected for robustness research. The combination of multi-scale convolutional kernels in the HED network architecture improves the robustness of edge detection by processing image details of different scales. The PiDiNet network architecture uses a combination of convolutional and separable convolutional layers with image gradient estimation to reduce edge positioning error. This research compares the edge detection performance of these two CNNs on noised natural images.

4. Materials and methods

To research the robustness of edge detection with HED and PiDiNet network, the technique of determining the influence of the noise level on the CNN edge detector efficiency is proposed. The stages of this technique are as follows.

1. As a result of literature review, CNNs are selected to detect edges on noisy images.
2. The types of image noise of interest to the researcher and the noise parameters whose values will be tuned by the researcher are determined. Mathematical modeling of noise images is performed.
3. The training and test sets of images from datasets for edge detection are selected.
4. The intervals of noise parameters for the research are determined. The images of test set are corrupted with controlled level noise.
5. The selected networks are learned to detect edges on training set of images or parameters of pre-trained networks are loaded.
6. The measures of edge detection performance are selected.
7. The trained networks are applied to noisy images and the edge detection performance is evaluated depending on the noise level in the images.
8. The obtained results are analyzed to generate the recommendations allowing to determine which CNN is better for edge detection when natural images affected by different level of white Gaussian noise, impulse noise, or multiplicative noise.

Then the literature is reviewed the proposed technique was not found by authors. If necessary, the elaborated technique of detecting the influence of the amount of noise on the CNN edge detector efficiency can be configured to research other CNNs with another known noise models. In the reviewed articles the processing of the noised images, ways to remove noise and improve image quality are considered, for example, using Gaussian filter, low-pass filters or wavelets [2]. The technique presented in this paper allows scientists to reasonably select the CNN architecture for edge detection, depending on the level of image noise, as well as to study the CNN in order to identify disadvantages and validate the obtained results.

Further the implementation of the technique of detecting the influence of the amount of noise on the CNN edge detector efficiency is considered. The first and second stages are discussed in this section. The following sections examine the remaining stages.

At the first stage the HED and PiDiNet networks are selected to detect edges on noisy images. The HED neural network is designed from several blocks of layers with side replies (Figure 1). Each layer block includes a convolutional layer, followed by a ReLU activation function, and then a pooling layer. As a result, image features at different scales are obtained with the side replies. After interpolation and concatenation, the side replies are combined into a single edge map [12].

The PiDiNet separable depth-wise convolutional structure has 4 stages with a shortcut for easy training. The max pooling layers are used among stages for down sampling. Each stage has 4 residual blocks including a depth-wise convolutional layer, a ReLU activation function, and a point-wise convolutional layer sequentially [14]. The exception is the first stage that has an initial convolutional layer and 3 residual blocks. The vanilla convolution in the depth-wise convolutional layers is replaced with pixel difference convolution in residual blocks (Figure 2). Any normalization layers are not included since the resolutions of the training images are not uniform.

At the second stage of determining the influence of the noise level on the CNN edge detector efficiency mathematical modeling of noisy images is performed. There are many types of noise affected on image processing [2, 17]. Gaussian noise, impulse noise, and speckle noise models are most often used when the image edge detection robustness is researched. These models allow

reproducing the scenarios of real image corruption by controlled variation of the noise parameter. The noise values are considered as random variables determined by the probability density function. Then noise is a mask of pixels of random color and brightness superimposed on the image. Considering a certain class of noise as a function allows focusing on determining its parameters and the effect of these parameters on the image quality. The result of noise generation is a pixel mask with specified values of parameters, specifically, noise intensity and distribution. The most common are additive and multiplicative noise [18].

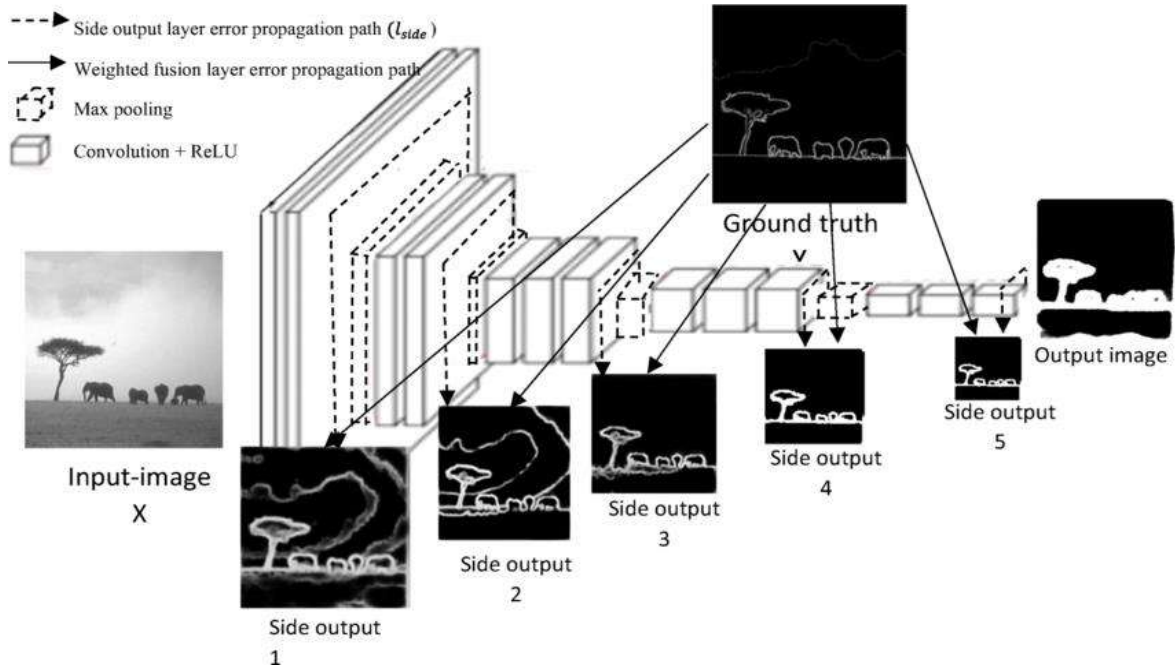


Figure 1: HED network architecture [9]

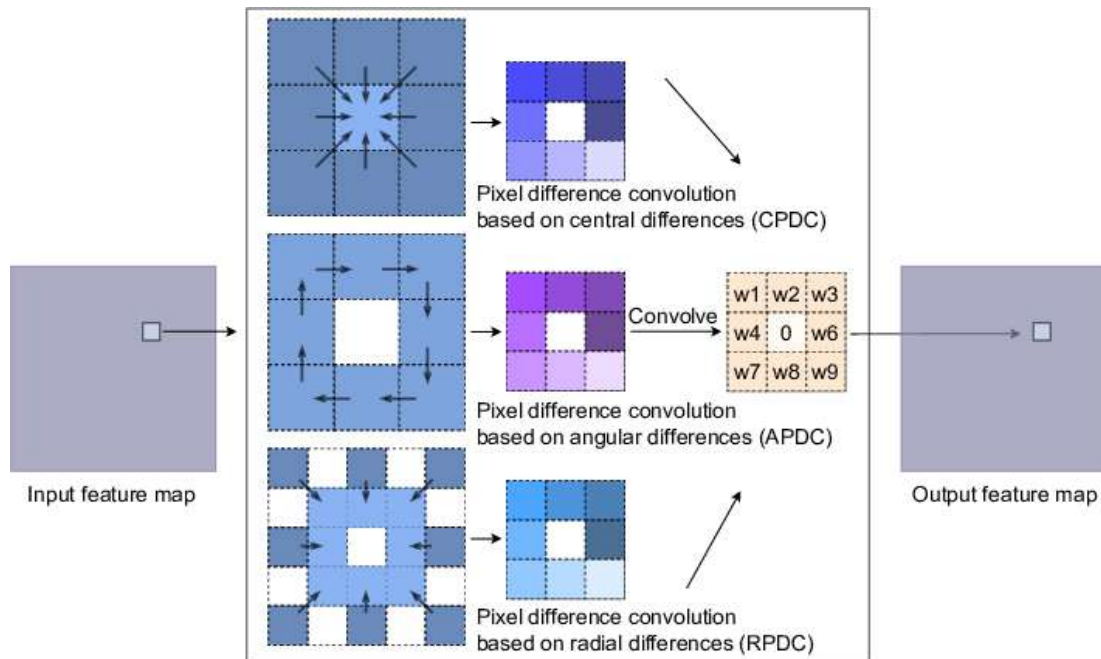


Figure 2: Pixel difference convolutions applied by PiDiNet network [14]

Additive noise predominantly determines the quality of natural images. It is assumed independent of the state of the system generating images. In most practical situations it is assumed to be Gaussian and have zero mean and standard deviation that is approximately constant over different color channels of the image. In addition, it is usually supposed that the values of additive noise are spatially uncorrelated, that is independent for neighboring pixels or pixels distant from each other [18].

Multiplicative noise model assumes dependence the noise on the state of the system generating images. Statistical characteristics of such noise depend on the content of the image. Often, the multiplicative noise variation is constant for image channels, since it is defined as the ratio of the local image variation to the square of the local mean [18].

At the same time, there are noises for which neither additive nor multiplicative models can be applied, for example, impulse noise. An image corrupted by impulse or salt and pepper noise will have dark pixels in light regions and light pixels in dark regions. This noise is typically caused by analog-to-digital conversion errors and bit errors while transmission [18].

The main reason for the appearance of Gaussian noise in the image is a lack of lighting while photographing. Another reasons for the Gaussian noise appearance are rather high temperature of the environment or the matrix of the device when long-term shooting. Impulse noise occurs because of image corruption while data transmission, or due to slight defects on the device's matrix. The cause of speckle noise can be the humidity of the environment, for example fog, as well as the use of wave lighting, which can be observed while laser scanning [2].

In statistical modeling of an image corrupted by white Gaussian noise, the latter is presented as (1) [17, 19]:

$$I_{K,gauss}(x,y) = I_K(x,y) + v_K(x,y), \quad (1)$$

where K means image channel which is red, green or blue; $I_{K,gauss}(x,y)$ is a noisy image channel; $I_K(x,y)$ is an original image channel; $v_K(x,y)$ is random additive noise in channel K . Here $v_K(x,y)$ is Gaussian distributed with m, s parameters. Here m is the mean; s is standard deviation of the noise.

When modeling impulse noise, the intensity of each point of the image channel is replaced with the probability Pa by the value a , and with the probability Pb is replaced by the value b . Additionally, the intensity of each point of the image channel is remained unchanged with the probability $1 - (Pa + Pb)$, where $Pa + Pb \leq 1$ [2]. Thus, the image corrupted with impulse noise is represented by the formulas (2) – (4):

$$I_{K,imp}(x,y) = a \text{ with probability } Pa, \quad (2)$$

$$I_{K,imp}(x,y) = b \text{ with probability } Pb, \quad (3)$$

$$I_{K,imp}(x,y) = I_K(x,y) \text{ with probability } 1 - (Pa + Pb), \quad (4)$$

where $I_{K,imp}(x,y)$ is an image channel corrupted with impulse noise [2].

Noise impulses can be positive or negative. Image digitization usually involves scaling and intensity limiting. After digitization, noise impulse becomes extreme. This corresponds to the appearance of completely black and white points in the image, since the noise impulse intensity is usually greater than the value of the useful signal. Thus, the values of a and b are usually considered as "intense" in the sense that they are equal to the minimum and maximum values that can be present in a digitized image. Therefore, negative impulses appear as black dots on the image, and positive impulses appear as white dots on the image [2]. If one of the probability values is zero, then impulse noise is called unipolar.

Speckle noise is a multiplicative noise occurring much less frequently in natural images than additive white Gaussian noise. Speckle noise mostly affects ultrasound and laser imaging, and is also common in coherent imaging systems such as synthetic aperture radars and sonar. On natural images it appears as a result of unfavorable shooting conditions (humidity, poor lighting) [17, 18]. Speckle noise is non-uniform noise that makes images grainier. Its values often have Rayleigh distribution or gamma distribution. In the paper Gaussian distribution is used, and each pixel of the original image is multiplied by the noise value by the formula (5) [17, 19]:

$$I_{K,sp}(x,y) = I_K(x,y)(1 + hS_K(x,y)), \quad (5)$$

where $I_{K,sp}(x,y)$ is an image corrupted with speckle noise; $S_K(x,y)$ is a random noise with Gaussian distribution, $m = 0, s = 1$; h is a multiplicative noise intensity.

The spatial characteristics of noise, as well as the question of whether there is a relationship between noise and image, are crucial for further research. We will assume that the noise distribution is not related to the image and does not depend on the spatial coordinates. But such assumptions may be at least unfounded in some cases. Examples are images obtained in situations with few quanta, such as X-ray and nuclear imaging, which are not considered in this paper.

5. Experiments

In this section the implementation of the proposed technique from the third to the sixth stage of detecting the impact of noise on the CNN edge detector efficiency is considered.

At the third stage of research of the robustness of the edge detection with the CNNs the natural images from the BSDS300 dataset are considered [20]. This dataset contains a total of 300 images with different complexity of scenes and textures, including 200 training images and 100 test images. The ground-truth images are also presented which are binary images with edges selected by 5-8 experts (edge maps) [20]. The performance of edge detection was evaluated by comparing edge maps obtained using the researched CNNs, with ground-truth images.

The fourth stage of the proposed technique is implemented as follows. To provide real-world cases of the noised images, the original images of the researched dataset are corrupted by one of various types of noise, such as Gaussian, impulse and speckle. The level of noise is controlled to obtain a set of images with noise parameters from minimum to maximum. This allows for a more detailed analysis of the noise effect on edge detection.

At the fifth stage of the proposed technique parameters of the pre-trained HED and PiDiNet networks are loaded to detect edges on the controlled noised images [21, 22]. The HED network was implemented using the publicly available Caffe Library and pre-trained on BSDS500 dataset [23] with the following hyper-parameters. These are mini-batch size (10), learning rate (1e-6), loss-weight for each side-output layer (1), momentum (0.9), initialization of the fusion layer weights (0.2), number of training iterations (10,000; divide learning rate by 10 after 5,000) et al. [21].

Implementation of pre-trained PiDiNet is based on the PyTorch library [22]. In detail, this network is randomly initialized and trained for 14 epochs with Adam optimizer. The initial learning rate was 0.005, which was decayed in a multi-step way (at epoch 8 and 12 with decaying rate 0.1). The values of other hyper-parameters are indicated in [14].

At the sixth stage of the proposed technique to evaluate the edge detection results the Precision Pr , Recall Rc , F1-score F_1 were used [1, 3]. They are represented by the formulas (6), (7):

$$F_1 = 2 Pr Rc / (Pr + Rc), \quad (6)$$

$$Pr = TP / (TP + FP), Rc = TP / (TP + FN), \quad (7)$$

where TP is a number of correctly detected edge pixels; FP is a number of background pixels that are incorrectly detected as edge pixels; FN is a number of edge pixels that are incorrectly detected as background pixels.

Precision is the fraction of relevant edge pixels detected on original image among the edge pixels on related ground-truth image. Recall is the fraction of relevant edge pixels among the all edge pixels detected on original image. F1-score is calculated as the harmonic mean of precision and recall. This is particularly useful in situations where false negatives and false positives are equally significant. In addition, the FOM value was used to evaluate the edge detection results [6]. There are three common errors associated with edge detectors: (1) missing valid edges, (2) inaccurate edge positioning, (3) classification of noise fluctuations as edges. Pratt has introduced a FOM that balances these three types of error, but mainly characterizes the error in the edge positioning [6]. The Pr , Rc , F_1 , FOM are varied from 0 to 1 and equal one for a well detected edge.

6. Results

In this section the implementation of the proposed technique from the seventh to the eighth stage of detecting the impact of noise on the CNN edge detector efficiency is considered. For images corrupted by white Gaussian noise, impulse noise, and speckle noise graphs of the edge detection performance measures on the noise level are shown in Figures 3, 4, 5 respectively. Orange line corresponds to HED network, gray line corresponds to PiDiNet network.

The results of edge detection by the HED and PiDiNet networks were researched for values of the Gaussian noise parameter σ from 0 to 45 in increments of 5. The probability of pixels corrupted by the impulse noise changed from 0.01 to 0.18 in increments of 0.02. The intensity of the speckle noise changed from 0.01 to 0.27 in increments of 0.03. The parameter ranges were obtained experimentally. It should be noted that images with a higher noise parameter are also allowed, but in the realities of natural images, such images are considered too corrupted and are not subject to further research. Figure 6 shows the noised BSDS300 image with low and high levels of noise. The image corrupted by Gaussian noise with standard deviation 20 and 40 is presented on Figure 6, a, d. The image corrupted

by impulse noise with probability of corrupted pixels 0.08 and 0.016 is shown on Figure 6, b, e. The image corrupted by speckle noise with intensity 0.25 and 0.5 is presented on Figure 6, c, f.

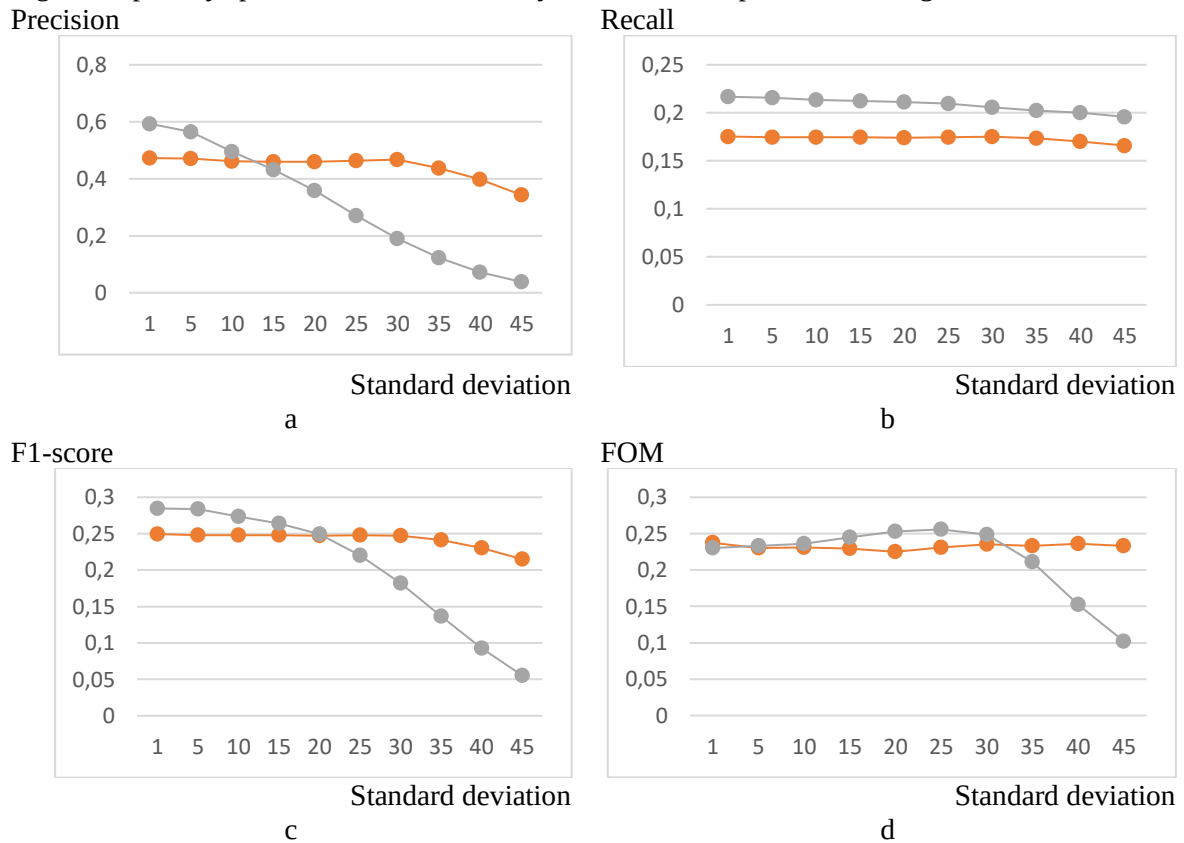


Figure 3: Graphs of dependence from the Gaussian noise parameter: a – Precision; b – Recall; c – F1; d – FOM; orange line corresponds to HED, gray line corresponds to PiDiNet network

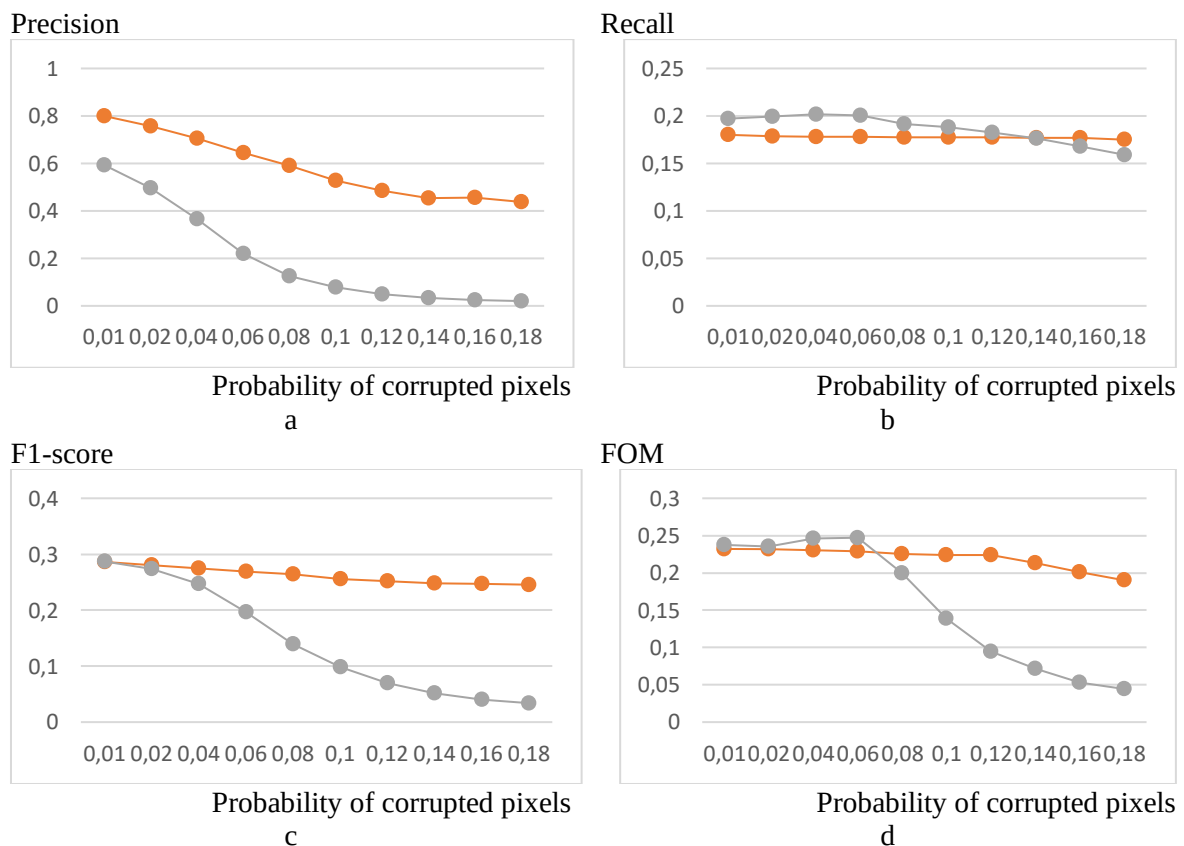


Figure 4: Graphs of dependence from the impulse noise parameter: a – Precision; b – Recall; c – F1; d – FOM; orange line corresponds to HED, gray line corresponds to PiDiNet network

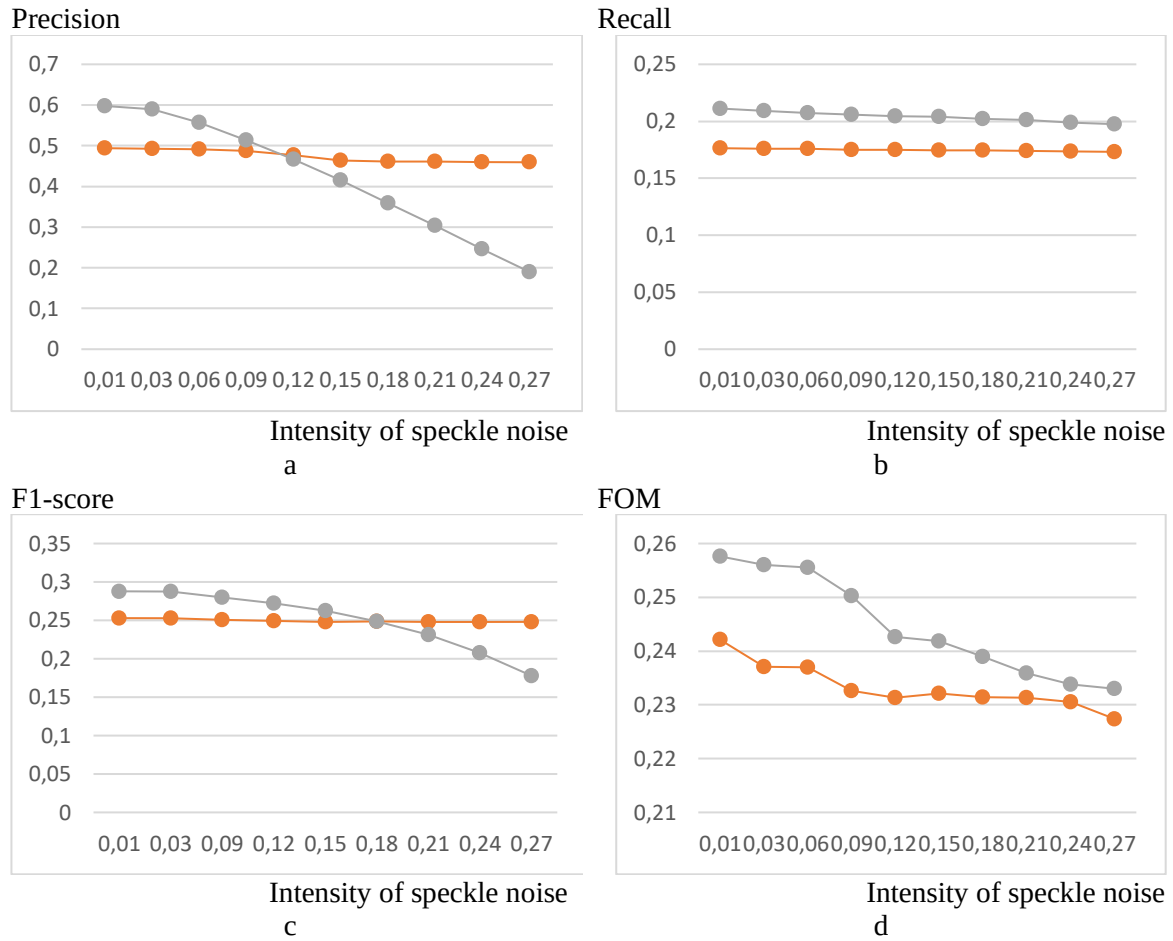


Figure 5: Graphs of dependence from the speckle noise parameter: a – Precision; b – Recall; c – F1-score; d – FOM; orange line corresponds to HED, gray line corresponds to PiDiNet

Also, the edge maps obtained by HED and PiDiNet networks are shown. The key feature of CNNs is altering the scale of abstraction while image processing. This can have a negative effect on edge detection, because the use of abstraction levels always leads to the loss of fine details and the accumulation of the strong edges. This is typical for edge detection by HED network (Figure 6, g–l).

The edge detection results obtained with PiDiNet network, on the contrary, includes weak and false edges because image gradient estimation applied with pixel difference convolution (Figure 6, m–r). Then the image noise level is increased the edge detection quality is reduced. The high level of impulse or speckle noise specially impacts on edge detection results (Figure 6, k, l, q, r), for which missing valid edges is observed. The detected edges are blurred.

7. Discussions

Analyzing the obtained results, we notice that the Precision and F1-score decreases almost proportionally when the Gaussian noise standard deviation increases (Figure 3, a, c). At first Precision and F1-score were taken into account. Then the edge detection performance of the PiDiNet network is up to 1.3 times higher than of the HED network if the standard deviation of the noise does not exceed 15. Otherwise, on the contrary, the edge detection performance of the HED network is up to 7 times higher than of the PiDiNet network.

Analyzing the obtained values of the Recall, notice that the PiDiNet network is up to 1.3 times more robust to Gaussian noise in images than the HED (Figure 3, b). This is due to the fact that ground-truth edges are quite thin, while the CNNs process images at several levels of abstraction. At high levels of abstraction there is always a loss of fine details, which generalizes and thickens the edges. The Pratt's FOM estimates the distance between ground truth edge and detected edge. Then for images corrupted by white Gaussian noise, the results of edge detection by the HED and PiDiNet networks showed almost similar FOM values for s less than 35 (Figure 3, d). For s greater than 35, the FOM of the HED network is up to 2 times higher than the FOM of the PiDiNet.

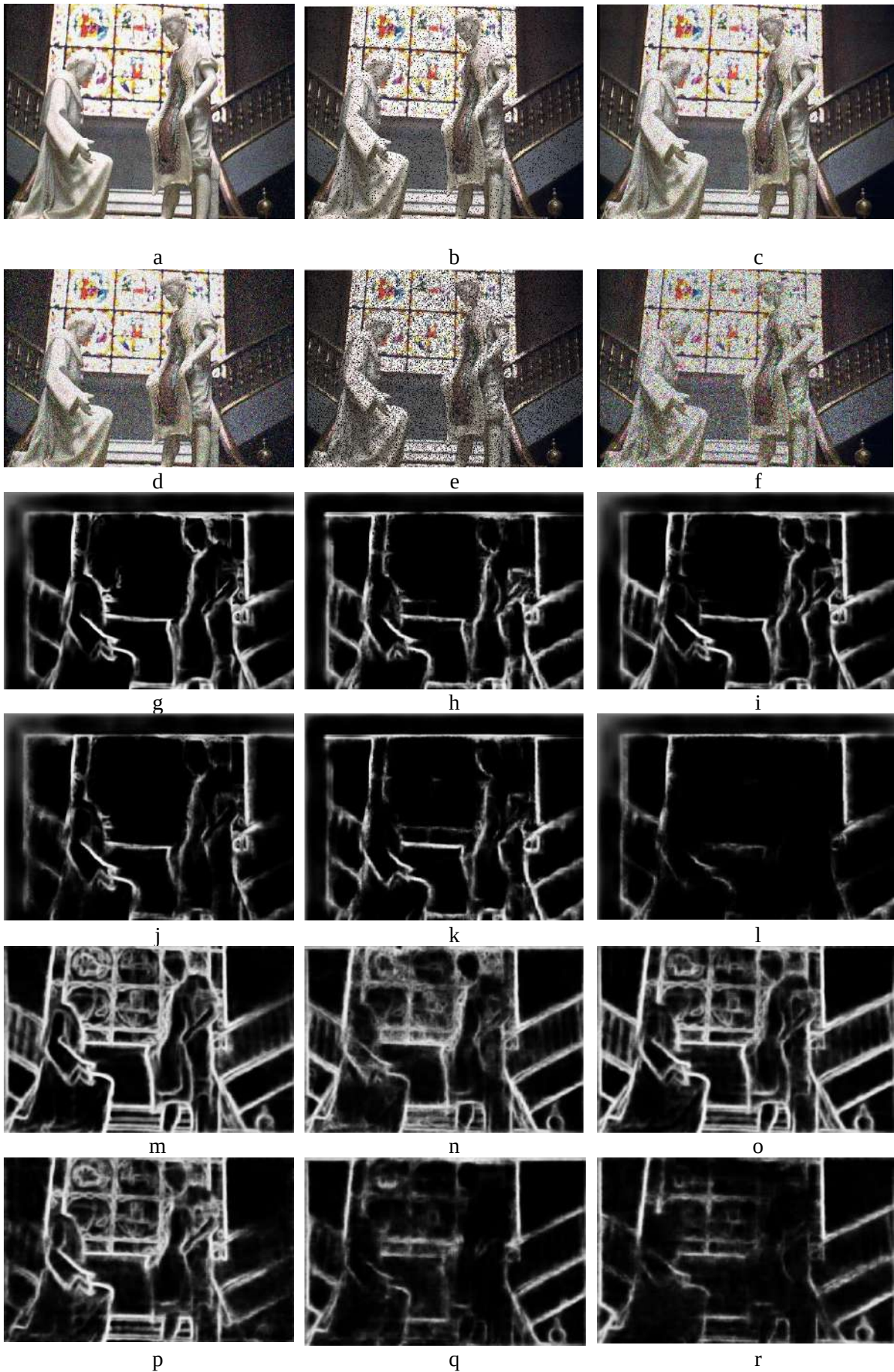


Figure 6: The edge detection results of noised images: a, d – the image corrupted by Gaussian noise with standard deviation 20 and 40; b, e – the image with probability of corrupted pixels 0.08 and 0.016; c, f – the image corrupted by speckle noise with intensity 0.25 and 0.5; g–l – edge maps

obtained by HED; m-r – edge maps obtained by PiDiNet

For images corrupted by impulse noise, the results of edge detection by HED and PiDiNet networks showed almost similar FOM values for probability of corrupted pixels less than 0.08 (Figure 4, d). For probability of corrupted pixels of 0.08 or more, the HED outperforms the PiDiNet in terms of FOM by up to a factor 4. For images corrupted by speckle noise, the PiDiNet network outperforms the HED in terms of FOM up to 1.1 times over the entire considered range of speckle noise intensity (Figure 5, d). This may be because the speckle noise can reduce the image contrast. Since the HED uses a multi-scale approach when detecting edges, an altering of pixel contrast negatively affects the network when edges are detected.

For images corrupted by speckle noise the edge detection performance on the noise level (Figure 5) are similar to the corresponding dependences for Gaussian noise (Figure 3). The Precision and F1-score decreases almost proportionally with increasing speckle noise intensity. In view of the Precision and F1-score, the quality of edge detection using the PiDiNet is up to 1.2 times higher than that of the HED if the speckle noise intensity does not exceed 0.15. Otherwise, the quality of edge detection using the HED is up to 2.3 times higher than that of the PiDiNet network. With the Recall, the PiDiNet is more robust to speckle noise than HED. Speckle noise can affect the contrast of the image, as well as create low-contrast edges, which complicates their detection. In the case the PiDiNet processed edges better than the HED. If for images corrupted by impulse noise the Precision and F1-score is estimated then the quality of edge detection by the HED is up to 5 times higher than that by the PiDiNet over almost the entire considered range of probability of corrupted pixels.

The Recall evaluates the ratio of correctly detected edges relative to all edges detected by the network. Low similar recall on the considered range of impulse noise parameter values is due to the architectures of the HED and PiDiNet. These networks can determine false edges when the level of abstraction is altered.

Since the F1-score evaluates the overall efficiency of the edge detector, its significance increases in the case of ambiguity of the Recall and Precision. In our case the F1-score confirms the advantage of HED over PiDiNet in edge detection on impulse noised images.

As a result of the analysis of the obtained results, recommendations were formulated to select the CNN for edge detection at a given or estimated noise level (Table 1). In particular, the HED is generally preferred at high noise levels. The PiDiNet is best used when the noise level is low. This indicates the weak robustness of the PiDiNet and the need for pre-processing of images when edges are detected by this network. The HED network has better values of Precision, F1 and FOM at significant noise levels.

Table 1

The recommendation on selection the CNN for edge detection at a given or estimated noise level

Type of noise	Parameter range	Recommended CNN
White Gaussian noise	[1, 20]	PiDiNet
	(20, 45]	HED
Impulse noise	[0.01, 0.06]	PiDiNet
	(0.06, 0.18]	HED
Speckle noise	[0.01, 0.15]	PiDiNet
	(0.15, 0.30]	HED

8. Conclusions

The actual scientific and applied problem of the robustness of the edge detection on noisy images by CNNs has been researched. The scientific novelty is the technique of determining the effect of the image noise level on the CNN edge detector efficiency which is proposed for the first time. When reviewing the literature devoted to the edge detection using neural networks, no similar techniques with the controlled noising of the image and subsequent evaluation of the edge detection performance were found.

The practical significance of obtained results is that to research the robustness of the edge detection by HED and PiDiNet networks the experiments have been conducted. The recommendations on robust CNN selection are formulated. The experimental results allow scientists to reasonably select the CNN architecture for edge detection depending on the level of image noise.

Prospects for further research are application of the elaborated technique to study the impact of mixed noise on the edge detection performance using CNNs. It is possible to complicate the representation of multiplicative noise by assuming gamma or Rayleigh distribution [24]. In a similar way, one can research the influence of the noise level on the effectiveness of solving other problems using CNNs, specifically, image segmentation, and identification or classification objects on images.

References

- [1] R. Sun, T. Lei, Q. Chen et. al., Survey of image edge detection, *Frontiers of Signal Processing* 3 (2022). doi:10.3389/frsip.2022.826967.
- [2] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 4th ed., Pearson, New York, NY, 2017.
- [3] M. Polyakova, RCF-ST: Richer Convolutional Features network with structural tuning for the edge detection on natural images, *Radio electronics, computer science, management* 4 (2023) 122–134. doi:10.15588/1607-3274-2023-4-12.
- [4] M. S. J. Rogers, M. Bithell, S. Brooks, T. Spencer, VEdge_Detector: automated coastal vegetation edge detection using a convolutional neural network, *International Journal of Remote Sensing* 42.13 (2021) 4809–4839. doi:10.1080/01431161.2021.1897185.
- [5] J. Jing, S. Liu, G. Wang, W. Zhang, C. Sun, Recent advances on image edge detection: A comprehensive review, *Neurocomputing* 503 (2022) 259–271. doi:10.1016/j.neucom.2022.06.083.
- [6] C. Akinlar, C. Topal, ColorED: color edge and segment detection by edge drawing, *Journal of Visual Communication and Image Representation* 44 (2017) 82–94. doi:10.1016/j.jvcir.2017.01.024.
- [7] M. Polyakova, V. Krylov, N. Volkova, The methods of image segmentation based on distributions and wavelet transform, in: *Proceedings of Data Stream Mining & Processing: IEEE First International Conference, DSMP, IEEE, Lviv, Ukraine, 2016*, pp. 243–247. doi:10.1109/DSMP.2016.7583550.
- [8] S. Xie, Z. Tu, Holistically-nested edge detection, *International Journal of Computer Vision* 125.5 (2017) 1–16. doi:10.1007/s11263-017-1004-z.
- [9] B. Ray, S. Mukhopadhyay, S. Hossain, S. Ghosal, R. Sarkar, Image steganography using deep learning based edge detection, *Multimedia Tools and Applications* 80.24 (2021) 33475–33503. doi:10.1007/s11042-021-11177-4.
- [10] Y. Liu, M. M. Cheng, X. Hu, K. Wang, X. Bai, Richer convolutional features for edge detection, in: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, Honolulu, HI, USA, 2017*, pp. 5872–5881. doi:10.1109/CVPR.2017.622.
- [11] Deng, C. Shen, S. Liu, H. Wang, X. Liu, Learning to predict crisp boundaries, in: *Proceedings of 15th European Conference on Computer Vision, ECCV, IEEE, Munich, 2018*, VI, pp. 562–578. doi:10.1007/978-3-030-01231-1_35.
- [12] R. Grompone von Gioi, G. Randall, A brief analysis of the holistically-nested edge detector, *Image Processing On Line* 12 (2022) 369–377. DOI:10.5201/ipol.2022.422.
- [13] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: *Proceedings of 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 2015*. URL: <https://arxiv.org/pdf/1409.1556>. doi:10.48550/arXiv.1409.1556.
- [14] Z. Su, W. Liu, Z. Yu et al., Pixel difference networks for efficient edge detection, in: *Proceedings of IEEE/CVF International Conference on Computer Vision, ICCV, IEEE, online, 2021*, pp. 5117–5127. doi:10.1109/ICCV48922.2021.00507.
- [15] Y. Li, X. S. Poma, G. Li et al., PiDiNeXt: an efficient edge detector based on parallel pixel difference networks, in: *Proceedings of 6th Chinese Conference on Pattern Recognition and Computer Vision, PRCV, Springer-Verlag, Xiamen, China, 2023*, pp. 261–272. doi:10.1007/978-981-99-8549-4_22.

- [16] Z. Yin, Z. Wang, C. Fan, X. Wang, T. Qiu, Edge detection via fusion difference convolution, *Sensors*, 23.15 (2023). doi:10.3390/s23156883.
- [17] P. Arulpandy, M. Trinita Pricilla, Speckle noise reduction and image segmentation based on a modified mean filter, *CAMES* 27.4 (2020) 221–239. doi: 10.24423/cames.290.
- [18] N. B. Shakhovska, O. I. Kosar, Analysis of common methods of noise overlaying on images, *Scientific Bulletin of UNFU* 28.1 (2018) 145–149. doi: 10.15421/40280129
- [19] M. Uss, B. Vozel, V. Lukin, K. Chehdi, Comparison of learning-based and maximum-likelihood estimators of image noise variance for real-life and synthetic anisotropic textures, in: *Proceedings of Image and Signal Processing for Remote Sensing XXVI*, SPIE, Edinburgh, United Kingdom, 2020, pp. 153303. doi: 10.1117/12.2573934.
- [20] Berkeley Segmentation Dataset 68. Kaggle. URL: <https://www.kaggle.com/code/mpwolke/berkeley-segmentation-data-set-68>.
- [21] Code for Holistically-Nested Edge Detection. URL: <https://github.com/s9xie/hed>.
- [22] Code for the ICCV 2021 paper "Pixel Difference Networks for Efficient Edge Detection" (Oral). URL: <https://github.com/hellozhuo/pidinet>.
- [23] The Berkeley Segmentation Dataset and Benchmark Web site, 2019. URL: <https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds>.
- [24] M. Uss, V. Lukin, B. Vozel, K. Chehdi, Accuracy of model-based and learning-based approaches for image noise variance estimation, in: *Proceedings of Ukrainian Microwave Week, UkrMW, IEEE, Kharkiv, Ukraine, 2020*, pp. 438–442. doi: 10.1109/UkrMW49653.2020.9252696.

Computer Modeling of Discrete Systems in the Case of Linear and non-Linear Restrictions on the Optimal Speed of the Aircraft Based on the Lagrange Method

Andriy Goncharenko and Serhii Teterin

National Aviation University, 1, Liubomyra Huzara Avenue, Kyiv, 03058, Ukraine

Abstract

The paper of the report represents the study dedicated to the main link of the aircraft flight speed to the minimal time of the air transportation process. The simplest problem of the aviation transportation technologies fundamental factors optimization models the considered process of the delivery and it makes an attempt to the process theoretical description. Two segment air traffic elementary chain of supply is implied at the presented research. The algorithm, which is used for calculating the objective parameters of the aircraft motion, is developed. Approaches to aircraft speed optimization are used. The speed of the delivery by the aircraft at each of the segments have been conditionally optimized. The objective value is the time of delivery. The aircraft speeds are subject to both linear and nonlinear constraints. The influence of the speeds' variations upon the conditionally minimal objective delivery time values are studied. Theoretical contemplations are conducted in the framework of the Lagrange uncertainty multipliers implementation. The hypothetical provisions of the derived mathematical models are illustrated with the help numerical simulation. The part of the computer modeling is conducted on the Mathcad platform at the educational and scientific laboratory "Modeling of transport systems and processes" of National Aviation University. The necessary diagrams are plotted. The obtained results of both theoretical study and computer simulation allows construction of optimal delivery chains with a better determination of the exchange point location.

Keywords

Computer modeling, optimization, simulation, aviation transport technologies, aircraft flight speed, speed optimization, delivery time minimization

1. Introduction

Computer and numerical simulation for searching optimal aircraft speed by the criterion of the minimal time of the delivery is an urgent task. That requires a proper maintenance of aircraft itself [1], as well of the airplane engines and powerplants [2], in order to support the aeronautical components' reliability [3] and risk [4] at the due level.

However, the unsolved part to the general problem there, at references [1 – 4], is the lack of the conditional optimization.

In such context, expected operational efficiency and utility [5] is combined with the choice problems [6].

This means that the transport technologies theoretical constraints, like in reference [7], strengthening learning for intelligent applications [8, 9] are important.

These variants of uncertainty can be estimated using entropy approaches [10].

In conjunction with the economic models [11], the entropy methods resulted in the subjective analysis theory [12] allows solving various types of the applicable problems, similar to the stated in the references of [13 – 16]. Some problems relevant to the air transport management and aviation transport technologies have already been posed in [17 – 20].

According to the presented concepts, a scientific gap that needs to be solved is the development of a reliable mathematical approach to the actual and important problems related to the formulation of the optimal combination of aviation resources, especially in relation to

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ andygoncharenko@yahoo.com (A. V. Goncharenko); sergiyteterin@gmail.com (S. O. Teterin)



0000-0002-6846-9660 (A. V. Goncharenko); 0009-0004-2662-6252 (S. O. Teterin)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

supporting the process of analytical decision-making based upon the advantages of the computer modeling.

Thus, in the case with the aircraft transportation speeds, it is necessary to formulate the scientific hypothesis of the conducted research as the speeds' variations, subject to both linear and nonlinear constraints, impact upon the conditional minimal value of the delivery time.

The problem statement of the presented research concerns with the theoretical studies focused on the calculations of the optimal speed of the aircraft as the key element of the aviation transport technologies.

Thus, the goals of the article are a general description of the possible optimization of the aircraft speed for the theoretical and mathematical obtaining of rational solutions.

2. Possibilities of optimization

The simplest problem of the aviation transportation technologies fundamental factors optimization stated here is based upon an elementary two segment supply chain consideration. The speed of the delivery by the aircraft at each of the segments could be conditionally optimized.

This necessitates the further development of the optimization methods of [18] and [19].

2.1. Basic concept

It is going to be considered the theoretical background for calculating aircraft movement parameters and approaches to their optimization.

The simulation of the aircraft motion was conducted with use of the software capabilities of the educational and scientific laboratory "Modeling of transport systems and processes".

2.1.1. A case of a linearly dependable constraint

Taking into account the speed of the aviation transportation delivery

$$T(v_1, v_2) = \frac{|AB|}{v_1 + v_2}, \quad (1)$$

where $T(v_1, v_2)$ is the time of the delivery; v_1 and v_2 are correspondingly the speeds of the first and the second aircraft that fly towards each other the same time $T(v_1, v_2)$; $|AB|$ is the distance covered by both aircraft in the time of $T(v_1, v_2)$ and at the speeds of v_1 and v_2 in respect.

Model (1) imply, for instance, a case when allows you to find an initial reference solution, and then, improving it, get the optimal solution.

Considering the condition of

$$P = f_1(v_1 - v_{1_{\min}}) + f_2(v_2 - v_{2_{\min}}) = \text{cost} = \text{idem}, \quad (2)$$

where P is some idempotent (independent upon the parameters of the considered problem model, stable, steady, unchanged, constant) value; f_1 and f_2 are the corresponding speeds coefficients; $v_{1_{\min}}$ and $v_{2_{\min}}$ are the minimal values in respect for the aircraft speeds of v_1 and v_2 possible range of variation. The idea of (2) is close to [18].

The linear dependence between the aircraft speeds of v_1 and v_2 coefficients values of f_1 and f_2 could be derived supposing as from (2).

$$P - f_1(v_1 - v_{1_{\min}}) = f_2(v_2 - v_{2_{\min}}). \quad (3)$$

And from (3)

$$v_2(v_1) = \frac{P}{f_2} - \frac{f_1}{f_2}(v_1 - v_{1_{\min}}) + v_{2_{\min}}. \quad (4)$$

Also assuming

$$v_2(v_1) = v_{2_{\max}} - \frac{v_{2_{\max}} - v_{2_{\min}}}{v_{1_{\max}} - v_{1_{\min}}}(v_1 - v_{1_{\min}}), \quad (5)$$

where $v_{1_{\max}}$ and $v_{2_{\max}}$ are the maximal values in respect for the aircraft speeds of v_1 and v_2 possible range of variation.

Comparing the corresponding members of (4) and (5)

$$\left. \begin{aligned} \frac{P}{f_2} + v_{2_{\min}} &= v_{2_{\max}}; \\ \frac{f_1}{f_2} &= \frac{v_{2_{\max}} - v_{2_{\min}}}{v_{1_{\max}} - v_{1_{\min}}}. \end{aligned} \right\} \quad (6)$$

System (6) yields

$$\left. \begin{aligned} f_2 &= \frac{P}{v_{2_{\max}} - v_{2_{\min}}}; \\ f_1 &= \frac{P}{v_{1_{\max}} - v_{1_{\min}}}. \end{aligned} \right\} \quad (7)$$

Having determined the coefficients of f_1 and f_2 from (2) – (7), it is possible to consider now the condition of (2) as a constraint to the objective function of (1):

$$\Phi(v_1, v_2) = \frac{v_1 - v_{1_{\min}}}{v_{1_{\max}} - v_{1_{\min}}} + \frac{v_2 - v_{2_{\min}}}{v_{2_{\max}} - v_{2_{\min}}} - 1 \equiv 0. \quad (8)$$

Therefore, the problem is becoming a problem of a conditional optimization.

Namely, find the optimal aircraft speeds: v_1 and v_2 , (1) – (8), extremizing the time of the delivery by the aviation transportation: $T(v_1, v_2)$, (1), subject to the only constraint of (2) as (8).

Thus, the extended Lagrange function is

$$L(v_1, v_2) = T(v_1, v_2) + \lambda \Phi(v_1, v_2) = \frac{|AB|}{v_1 + v_2} + \lambda \left(\frac{v_1 - v_{1_{\min}}}{v_{1_{\max}} - v_{1_{\min}}} + \frac{v_2 - v_{2_{\min}}}{v_{2_{\max}} - v_{2_{\min}}} - 1 \right), \quad (9)$$

where λ is the Lagrange uncertain multiplier.

The necessary conditions for a possible extremum of (9) existence are

$$\left. \begin{aligned} \frac{\partial L(v_1, v_2)}{\partial v_1} &= 0; \\ \frac{\partial L(v_1, v_2)}{\partial v_2} &= 0; \\ \frac{\partial L(v_1, v_2)}{\partial \lambda} &= 0. \end{aligned} \right\} \quad (10)$$

Then

$$\left. \begin{aligned} \frac{\partial L(v_1, v_2)}{\partial v_1} &= -\frac{|AB|}{(v_1 + v_2)^2} + \frac{\lambda}{v_{1_{\max}} - v_{1_{\min}}} = 0; \\ \frac{\partial L(v_1, v_2)}{\partial v_2} &= -\frac{|AB|}{(v_1 + v_2)^2} + \frac{\lambda}{v_{2_{\max}} - v_{2_{\min}}} = 0; \\ \frac{\partial L(v_1, v_2)}{\partial \lambda} &= \frac{v_1 - v_{1_{\min}}}{v_{1_{\max}} - v_{1_{\min}}} + \frac{v_2 - v_{2_{\min}}}{v_{2_{\max}} - v_{2_{\min}}} - 1 = 0. \end{aligned} \right\} \quad (11)$$

The systems of (10) and (11) yield

$$\left. \begin{aligned} \frac{\lambda}{v_{1\max} - v_{1\min}} &= \frac{|AB|}{(v_1 + v_2)^2}; \\ \frac{\lambda}{v_{2\max} - v_{2\min}} &= \frac{|AB|}{(v_1 + v_2)^2}; \\ \frac{v_1 - v_{1\min}}{v_{1\max} - v_{1\min}} + \frac{v_2 - v_{2\min}}{v_{2\max} - v_{2\min}} &= 1. \end{aligned} \right\} \quad (12)$$

From the first two equations of system (12)

$$v_{1\max} - v_{1\min} = v_{2\max} - v_{2\min} = C. \quad (13)$$

Then, using the third equation of system (12)

$$C = v_1 - v_{1\min} + v_2 - v_{2\min}. \quad (14)$$

Applying (13) and (14) to the first or second equation of system (12)

$$\frac{\lambda}{v_1 - v_{1\min} + v_2 - v_{2\min}} = \frac{|AB|}{(v_1 + v_2)^2}. \quad (15)$$

From (15)

$$\lambda = \frac{|AB|(v_1 - v_{1\min} + v_2 - v_{2\min})}{(v_1 + v_2)^2}. \quad (16)$$

But system (12) might have a solution. It is because its first two equations are the same. Indeed, instead of (12) there is a possibility to write

$$\left. \begin{aligned} \frac{\lambda}{C} &= \frac{|AB|}{(v_1 + v_2)^2}; \\ \frac{\lambda}{C} &= \frac{|AB|}{(v_1 + v_2)^2}; \\ \frac{v_1 - v_{1\min}}{C} + \frac{v_2 - v_{2\min}}{C} &= 1. \end{aligned} \right\} \quad (17)$$

The rewritten system (12) means

$$\left. \begin{aligned} \lambda &= \frac{|AB|C}{(v_1 + v_2)^2}; \\ v_1 - v_{1\min} + v_2 - v_{2\min} &= C. \end{aligned} \right\} \quad (18)$$

Then

$$\left. \begin{aligned} \lambda &= \frac{|AB|C}{(C + v_{1\min} + v_{2\min})^2}; \\ v_1 + v_2 &= C + v_{1\min} + v_{2\min}. \end{aligned} \right\} \quad (19)$$

So, $v_1 + v_2$ has a constant (idempotent) value. One of the speeds can be determined through the other one. It can be resolved with the help of the expressions of (2) – (5), or conditions of (8), (9), the third equations of (11), (12), as well as from (14), the third equation of (17), and the second equations of (18) or (19) too.

Moreover, therefore, the duration of flight has a idempotent (stable, steady, unchanged, constant) value as well. That follows the model expression (1).

Thus, the system of two equations (18) obtained from/of (12) has happened to be a system of two equations with three unknowns. And (19) is actually the one equation with the two unknowns.

The following sections dedicated to simulation and discussion will visualize and dispute upon the case set as (1) – (19).

2.1.2. A variation upon the constraint

This subsection deals with the time: $T(v_1, v_2)$, (1), but subject a nonlinear constraint in the type of the variation to the equation of (2) or to the equation (4).

The necessary conditions for a possible extremum of (9) existence are

Now, it is going to be

$$v_2(v_1) = v_{2_{\max}} - \frac{v_{2_{\max}} - v_{2_{\min}}}{v_{1_{\max}} - v_{1_{\min}}} (v_1 - v_{1_{\min}}) + \Delta(v_1), \quad (20)$$

where $\Delta(v_1)$ is the variation to the linear dependence of v_2 , (2), or the equation (4), both v_2 and $\Delta(v_1)$ being dependent upon v_1 .

Suppose a nonlinear variation of $\Delta(v_1)$, (20). The proposed model is

$$\Delta(v_1) = k_{\Delta_1} (v_1 - v_{1_{\min}}) (v_1 - v_{1_{\max}}), \quad (21)$$

where k_{Δ_1} is a coefficient.

On the other hand, it is possible to model the opposite side, of the equation of (2), or to the equation (4), dependence of v_2 , (2), upon v_1 variation. That is

$$v_2 = \frac{P}{f_2} - \frac{f_1}{f_2} (v_1 - v_{1_{\min}}) + v_{2_{\min}} + \delta(v_1), \quad (22)$$

where $\delta(v_1)$ is the variation to the linear dependence of v_2 , (2) or the equation (4), upon v_1 , however this time the variation provides the v_2 values on the contrary to the previous option of (20) and (21).

The variation itself can have formally the mathematically identical expression though:

$$\delta(v_1) = k_{\delta_1} (v_1 - v_{1_{\min}}) (v_1 - v_{1_{\max}}), \quad (23)$$

where k_{δ_1} is a coefficient.

Making allowance for the above option of (20) and (21), just for the certainty of the problem setting, the new constraint will have the view of

$$\Phi(v_1, v_2) = \frac{P}{f_2} - \frac{f_1}{f_2} (v_1 - v_{1_{\min}}) + v_{2_{\min}} + \Delta(v_1) - v_2 \equiv 0. \quad (24)$$

That means

$$L(v_1, v_2) = T(v_1, v_2) + \lambda \Phi(v_1, v_2) = \frac{|AB|}{v_1 + v_2} + \lambda \left[\frac{P}{f_2} - \frac{f_1}{f_2} (v_1 - v_{1_{\min}}) + v_{2_{\min}} + \Delta(v_1) - v_2 \right]. \quad (25)$$

Or in the view convenient for differentiating

$$L(v_1, v_2) = \frac{|AB|}{v_1 + v_2} + \lambda \left[\frac{P}{f_2} - \frac{f_1}{f_2} (v_1 - v_{1_{\min}}) + v_{2_{\min}} + k_{\Delta_1} (v_1 - v_{1_{\min}}) (v_1 - v_{1_{\max}}) - v_2 \right]. \quad (26)$$

After applying the conditions of (10) to (25)

$$\left. \begin{aligned} \frac{\partial L(v_1, v_2)}{\partial v_1} &= -\frac{|AB|}{(v_1 + v_2)^2} + \lambda \left\{ -\frac{f_1}{f_2} + k_{\Delta_1} [(v_1 - v_{1_{\min}}) + (v_1 - v_{1_{\max}})] \right\} = 0; \\ \frac{\partial L(v_1, v_2)}{\partial v_2} &= -\frac{|AB|}{(v_1 + v_2)^2} - \lambda = 0; \\ \frac{\partial L(v_1, v_2)}{\partial \lambda} &= \frac{P}{f_2} - \frac{f_1}{f_2} (v_1 - v_{1_{\min}}) + v_{2_{\min}} + k_{\Delta_1} (v_1 - v_{1_{\min}}) (v_1 - v_{1_{\max}}) - v_2 = 0. \end{aligned} \right\} \quad (27)$$

The second equation of (27) immediately meant that

$$\lambda = -\frac{|AB|}{(v_1 + v_2)^2}. \quad (28)$$

Then, substituting (28) for its value into the first equation of (27) it yields

$$-\frac{|AB|}{(v_1 + v_2)^2} - \frac{|AB|}{(v_1 + v_2)^2} \left\{ -\frac{f_1}{f_2} + k_{\Delta_1} [(v_1 - v_{1_{\min}}) + (v_1 - v_{1_{\max}})] \right\} = 0. \quad (29)$$

And

$$-\frac{|AB|}{(v_1 + v_2)^2} \left[1 - \frac{f_1}{f_2} + k_{\Delta_1} (2v_1 - v_{1_{\max}} - v_{1_{\min}}) \right] = 0. \quad (30)$$

Which means

$$1 - \frac{f_1}{f_2} + k_{\Delta_1} (2v_1 - v_{1_{\max}} - v_{1_{\min}}) = 0. \quad (31)$$

And

$$v_{1_{\text{opt}}} = \frac{1}{2} \left(\frac{f_1 - f_2}{k_{\Delta_1} f_2} + v_{1_{\max}} + v_{1_{\min}} \right). \quad (32)$$

The solution (32) ensures

$$v_{2_{\text{opt}}} = \frac{P}{f_2} - \frac{f_1}{f_2} (v_{1_{\text{opt}}} - v_{1_{\min}}) + v_{2_{\min}} + k_{\Delta_1} (v_{1_{\text{opt}}} - v_{1_{\min}}) (v_{1_{\text{opt}}} - v_{1_{\max}}). \quad (33)$$

2.2. Simulation

Let's consider the results obtained with the help of the theoretical considerations mentioned above using formulas (1) - (20) and calculation procedures. In the interests of achieving the goal of the study, computer modeling of the process of objectivity of the criteria for evaluating the optimization of transport work in the implementation of air transportation was carried out.

2.2.1. Computer modeling with the linearly dependable constraint

In case of (1) - (19), the accepted calculation data are as follows:

$$|AB| = 1 \cdot 10^4, \quad v_1 = 600 \dots 1 \cdot 10^3 \quad \text{and} \quad v_2 = 600 \dots 1 \cdot 10^3. \quad (34)$$

The results for $T(v_1, v_2)$ obtained by (1) with the use of data (28) are shown in the Figure 1

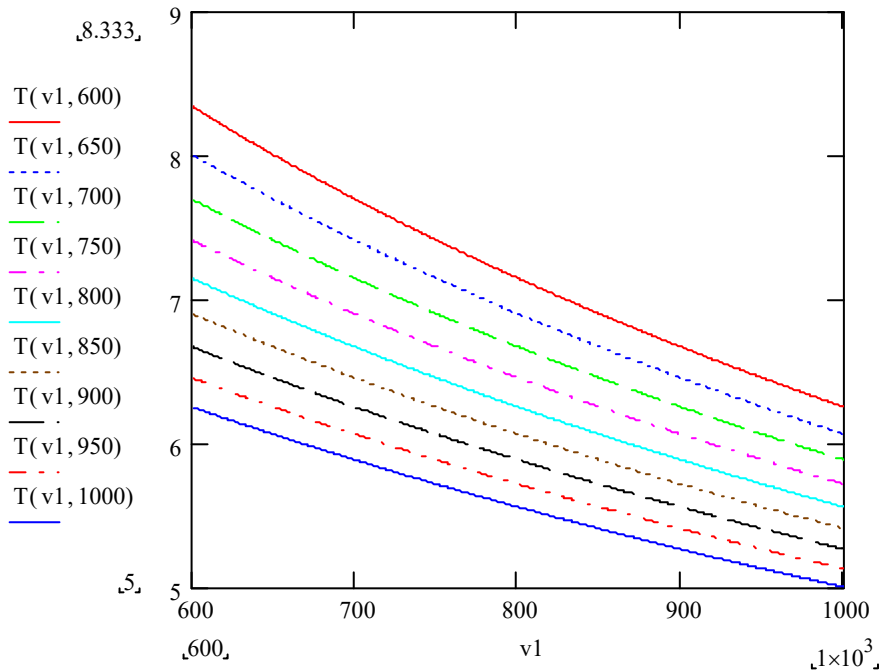


Figure 1: Time of the air transportation delivery

The three-dimensional plot of $T(v_1, v_2)$ by (1) is shown in the Figure 2.

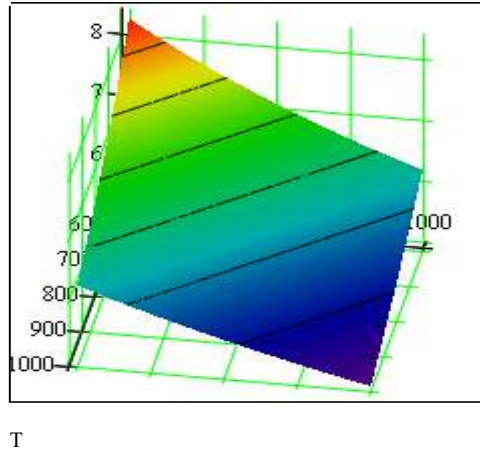


Figure 2: Time of the air transportation delivery

Applying the constraint in the view of (2), or $v_2(v_1)$: (4), or (5), to the duration of the aircraft transportation delivery (flight time), i.e.

$$T[v_1, v_2(v_1)] = \frac{|AB|}{v_1 + v_2(v_1)} = T(v_1). \quad (35)$$

In such case, $T(v_1, v_2(v_1))$: (1), modified to (35), it proves to have no extremum shown in the Figure 3.

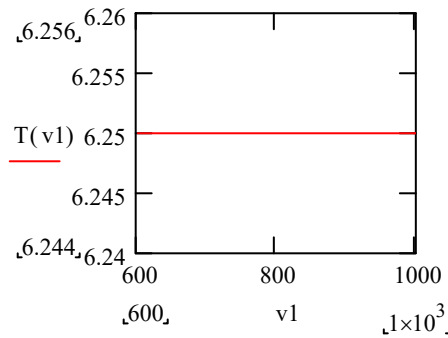


Figure 3: Absence of the extremum of the time of the delivery

The constant (idempotent) value of $T(v_1, v_2(v_1))$: (1), modified to (35), visible in the Figure 3 relates to the equations of (2) – (5), represented in the Figure 4.

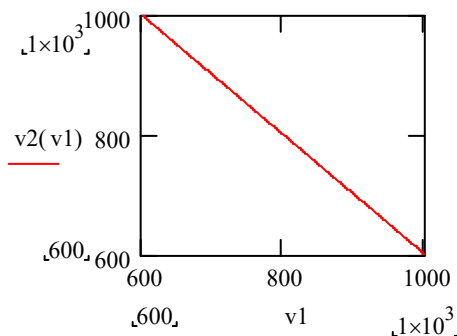
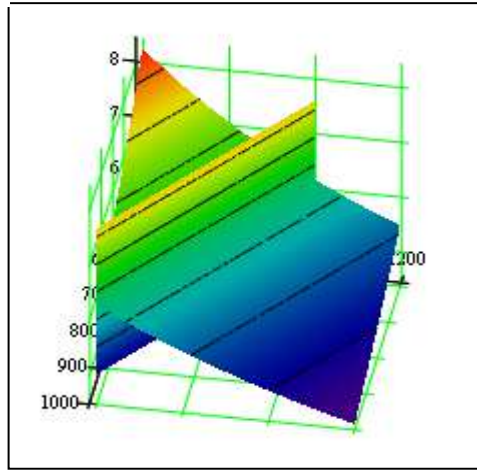


Figure 4: Linearly constrained speeds dependence

Additional data used for plotting diagrams in the Figures 3 and 4 may be relevant to the P value, (2) – (4) and (6), (7), however, it can be canceled or substituted, (8) – (19).

The absence of the extremum is noticeable in the three-dimensional plots of $T(v_1, v_2(v_1))$: (1), and the equations of $v_2(v_1)$: (2) – (5), illustrated for the perceptual ease in the Figure 5.



T, (X, Y, Z)

Figure 5: Absence of the extremum of the time of the air transport delivery

X,Y,Z shown in the Figure 5 are the parametric equations of the plain symbolizing the linear constraints (2) – (5).

1. The results of additional experimental studies made it possible to obtain new data regarding the values of the weighting coefficients of the component indicators of the integral indicator and to reveal a significant deviation from the values obtained according to experts' assessments.
2. Local extrema are more inherent in the solution of optimization tasks of the parameters of specific technologies and devices, since, as a rule, they have technical limitations of independent variable objective functions.

2.2.2. Computer modeling with the nonlinearly dependable constraint

In the case with the aircraft transportation speeds variations of (20) – (33), in addition to the data of (34), there is a need to have data for the computer simulations of the variations: $\Delta(v_1)$ and $\delta(v_1)$.

In fact, it was necessary to accept the values for the coefficients of k_{Δ_1} and k_{δ_1} : entering the expressions (21) and (23), and used throughout the modeling (20) – (33). Those data were as the following:

$$k_{\Delta_1} = -5 \cdot 10^{-3} \quad \text{and} \quad k_{\delta_1} = 5 \cdot 10^{-3}. \quad (36)$$

The results are presented in the Figure 6.

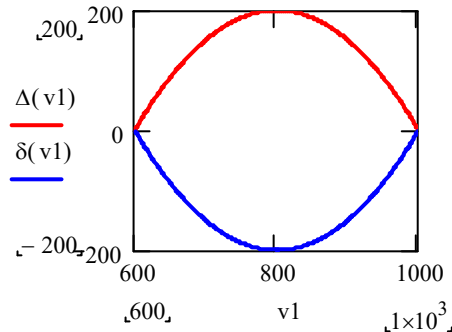


Figure 6: Aircraft speeds variations

The variated speeds with the basic one are shown in the Figure 7.

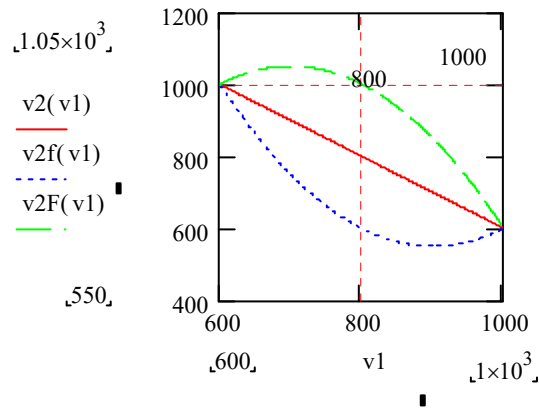


Figure 7: Variated aircraft speeds

Computer modeling for the time of the air transportation delivery is illustrated in the Figure 8.

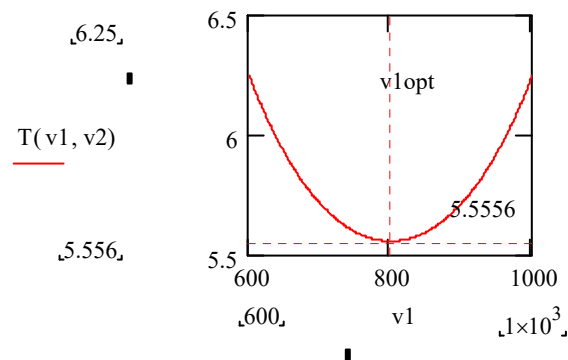
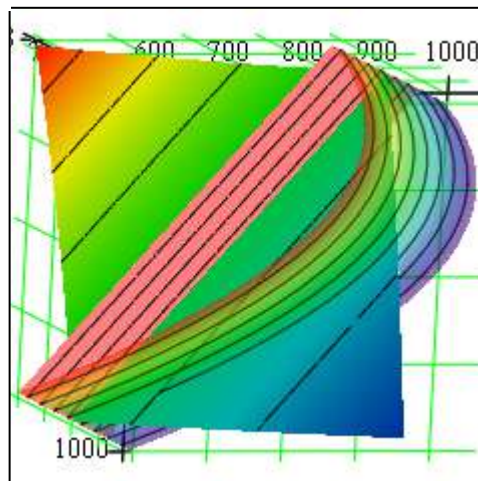


Figure 8: Time of the air transportation delivery constrained by the nonlinearly dependable aircraft speeds

The three-dimensional plots of $T(v_1, v_2)$ by (1) and $v_2(v_1)$ by (2) – (5), as well as $v_2(v_1)$ by (20) are shown in the Figure 9.



$T, (X, Y, Z), (X, Y_m, Z)$

Figure 9: Time of the air transportation delivery subject to constraints upon the aircraft speeds

The extreme values are shown in the Figures 10 and 11 as well.

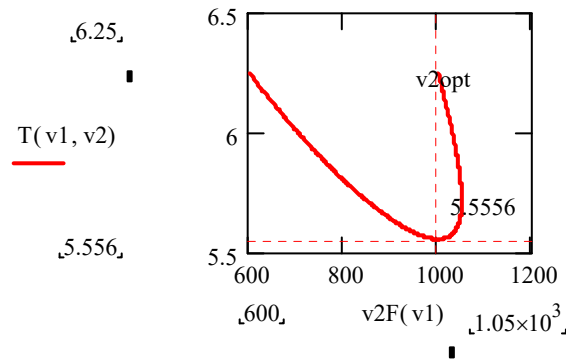


Figure 10: Phase diagram of the time of the air transportation delivery subject to constraints upon the aircraft speeds

The phase portraits in the Figures 10 and 11 demonstrate the minimal time and the optimal speeds combination.

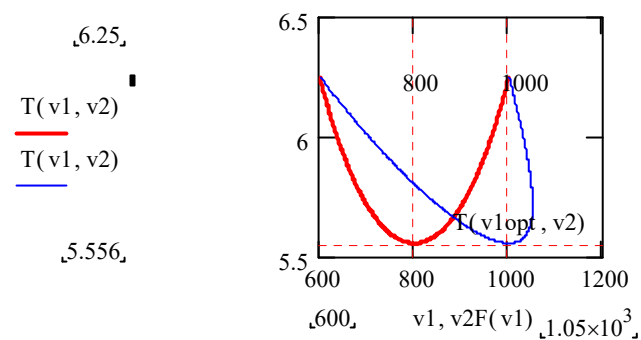


Figure 11: Combined phase portrait of the time of the air transportation delivery subject to constraints upon the aircraft speeds

3. Discussion

As it was shown, the creation of similar formalized models, that is, the relationship of target functions at different levels of the system hierarchy, will allow to maximize adequacy to optimal conditions of air transport operation.

The results of the experiment and the computational experiment based on the mathematical model by successive approximation by appropriate iterative methods are compared.

1. Optimization will always end with the search for local extrema of the objective functions, since the intervals of variation of the independent variables included in the objective functions are set a priori.

2. Phase portraits and trajectories of oscillation forms in the configuration space of the system were constructed and analyzed. The conditions for the localization of the forms of oscillations of the system have been obtained. The stability of the oscillation forms was studied. The presented system and mathematical model can be a source for new modeling approaches.

At the first stage of the research, a problem was identified that lay in the optimization of the aircraft's speed. To achieve this goal and systematize our understanding of the problem and potential ways to solve it, the following was defined:

- The problem that required our attention and what goal we want to achieve through the research.
- Possible ways of solving the problem, various alternatives were considered and the most effective and suitable variant of its mathematical solution was chosen.

The improvement of the criteria for evaluating the transport work in the performance of air transportation is carried out in the direction of expanding the list of factors that are taken into account when determining the relevant indicators, successively - the number (mass) of objects of transportation (passengers and cargo), range (distance between the points of departure and

destination) and speed (time) of their spatial movement (delivery) from the point of departure to the destination.

If we draw a parallel between the ratio of optimal speed and the theory of individual risk perception, then this may make it possible to conduct another study regarding the theoretical-mathematical model of the demand for insurance services based on the conditional optimization apparatus. When solving this problem by the method of undetermined Lagrange multipliers, the derivative must equal zero before the necessary extremum condition. Accordingly, in this case, at the optimal point, the budgetary limitation of the insurance cost should be beneficial for the policyholder.

4. Conclusion

The obtained values as the implementation of this study allowed to analytically and graphically determine the region of the optimal solution, taking into account the limitations of the objective function.

The conditionally optimized aircraft speeds ensure minimal time of the air transport delivery, which in turn leads to the improvement of the air transportation technologies.

For further research, it is proposed to investigate the dynamics, of the process of choosing the desired optimal technologies of air transport, and models based on the given calculation conditions that implement operational alternatives and the subjective entropy.

References

- [1] M. J. Kroes, W. A. Watkins, F. Delp, R. Sterkenburg, Aircraft Maintenance and Repair, 7th. ed., McGraw-Hill, Education, New York, NY, 2013.
- [2] T. W. Wild, M. J. Kroes, Aircraft Powerplants, 8th. ed., McGraw-Hill, Education, New York, NY, 2014.
- [3] B. S. Dhillon, Maintainability, Maintenance, and Reliability for Engineers, Taylor & Francis Group, New York, NY, 2006.
- [4] D. J. Smith, Reliability, Maintainability and Risk. Practical Methods for Engineers, Elsevier, London, 2005.
- [5] M. Amarante, Conditional expected utility, Theory and Decision 83. (2017) 1-19. DOI:10.1007/s11238-017-9597-9, https://www.researchgate.net/publication/315902126_Conditional_expected_utility.
- [6] R. D. Luce, Individual Choice Behavior: A theoretical analysis, Dover Publications, Mineola, NY, 2014.
- [7] D. Everett, et al., Multisystem Bayesian constraints on the transport coefficients of QCD matter. Physical Review, C 103.5 (2021): 054904. DOI:10.1103/PhysRevC.103.054904, <https://journals.aps.org/prc/pdf/10.1103/PhysRevC.103.054904>.
- [8] A. Coronato, M. Naeem, G. De Pietro, G. Paragliola, Reinforcement learning for intelligent healthcare applications: A survey, Artificial Intelligence in Medicine, Volume 109, 2020, 101964, ISSN 0933-3657, <https://doi.org/10.1016/j.artmed.2020.101964>. (<https://www.sciencedirect.com/science/article/pii/S093336572031229X>)
- [9] Mehta, Yash, et al., Recent trends in deep learning based personality detection, Artificial Intelligence Review, 53.4 (2020): 2313-2339. <https://doi.org/10.1007/s10462-019-09770-z>.
- [10] F. C. Ma, P. H. Lv, M. Ye, Study on global science and social science entropy research trend, in: Proceedings of the IEEE International Conference on Advanced Computational Intelligence, ICACI, Nanjing, Jiangsu, China, 2012, pp. 238-242. <https://ieeexplore.ieee.org/document/6463159>. doi:10.1109/ICACI.2012.6463159.
- [11] E. Silberberg, W. Suen, The Structure of Economics. A Mathematical Analysis, McGraw-Hill Higher Education, New York, NY, 2001.
- [12] V. Kasianov, Subjective Entropy of Preferences. Subjective Analysis, Institute of Aviation Scientific Publications, Warsaw, Poland, 2013.

- [13] O. Solomentsev, M. Zaliskyi, T. Herasymenko, O. Kozhokhina, Yu. Petrova, Data processing in case of radio equipment reliability parameters monitoring, in: Proceedings of the International Conference on Advances in Wireless and Optical Communications, RTUWO, Riga, Latvia, 2018, pp. 219–222. <https://ieeexplore.ieee.org/abstract/document/8587882>. doi:10.1109/RTUWO.2018.8587882.
- [14] A. M. B Pavani et al., "VISIR+ Project Follow-up after four years: Educational and research impact," 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA, 2023, pp. 1–8, doi: 10.1109/FIE58773.2023.10343298.
- [15] D. Shevchuk, O. Yakushenko, L. Pomytkina, D. Medynskyi, Y. Shevchenko, Neural network model for predicting the performance of a transport task, Lecture Notes in Civil Engineering. 130 LNCE (2021) 271–278. doi:10.1007/978-981-33-6208-6_27.
- [16] S. Subbotin, The neuro-fuzzy network synthesis and simplification on precedents in problems of diagnosis and pattern recognition, Opt. Mem. Neural Networks 22 (2013) 97–103. <https://doi.org/10.3103/S1060992X13020082>.
- [17] A. V. Goncharenko, Multi-optional hybridization for UAV maintenance purposes, in: Proceedings of the IEEE International Conference on Actual Problems of UAV Developments, APUAVD, Kyiv, Ukraine, 2019, pp. 48–51. <https://ieeexplore.ieee.org/abstract/document/8943902>. doi:10.1109/APUAVD47061.2019.8943902.
- [18] A. V. Goncharenko, Relative pseudo-entropy functions and variation model theoretically adjusted to an activity splitting, in: Proceedings of the IEEE International Conference on Advanced Computer Information Technologies, ACIT'2019, Ceske Budejovice, Czech Republic, 2019, pp. 52–55. <https://ieeexplore.ieee.org/abstract/document/8779876>. doi:10.1109/ACITT.2019.8779876.
- [19] A. Goncharenko, Development of a theoretical approach to the conditional optimization of aircraft maintenance preference uncertainty, Aviation 22 (2) (2018) 40–44. <https://journals.vilniustech.lt/index.php/Aviation/article/view/5929>. <https://doi.org/10.3846/aviation.2018.5929>.
- [20] A. V. Goncharenko, Airworthiness support measures analogy to the prospective roundabouts alternatives: theoretical aspects, Journal of Advanced Transportation Article ID 9370597 2018 (2018) 1–7. <https://www.hindawi.com/journals/jat/2018/9370597/>. <https://doi.org/10.1155/2018/9370597>.

Modeling and Programming of Elements of Integrated Systems in Industrial Controller Languages

Mykhailo Poliakov¹, Bohdan Zhurakovskiy², Oleksii Poliakov² and Ivan Shyshkin¹

¹ National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

² National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, Ukraine

Abstract

Context. Building integrated systems based on industrial controllers is a promising direction for creating adaptive control systems. The problems in this area are the limited functionality of operating and control automata models, the lack of a methodology for their software implementation as part of typical elements of an integrated system (TEIS).

Objective. The goal of the work is to expand the functionality of TEIS, to create a methodology for their programming in the languages of the IEC 61131-3 standard. The scientific novelty of the work lies in the fact that new types of controls for the structure and parameters of operational and control automata of an integrated system are proposed, such as control of the initial state, blocking of specified states, control of the output parameters of the control automaton and the parameters of operational automata.

Method. The possibilities of changing the parameters of elements of set-theoretic models of TEIS control and operational automata by external control from the overlying control automaton of the hierarchical system are analyzed. A methodology for detailing the control structure to the Program Organization Unit level in the languages of the IEC 61131-3 standard is described.

Experiments. An example of TEIS programming in languages according to the IEC 61131-3 standard in the OpenPLC application environment with subsequent loading and execution of the code on the Arduino microcontroller board is given.

Results. An experiment with a logical controller and a physical board confirmed the functionality of the structural and parametric adaptation of the control algorithm inherent in the TEIS.

Conclusions. The current problem of modeling adaptive behavior in an integrated hierarchical system and programming such behavior using the languages of the IEC 61131-3 standard has been solved. The proposed methods for controlling the parameters of operating and control automata as part of the TEIS expand the functionality of adaptive behavior in the system. The practical value of the proposed methodology for implementing adaptive behavior lies in the possibility of constructing TEIS in the programming languages of industrial controllers.

Keywords

Integrated control system, programming languages for industrial controllers, controlled operating and control machines.

1. Introduction

Programmable logic controllers (PLC) are the main elements of complex industrial automation systems [1, 2]. PLC programming is performed in languages according to the IEC 61131-3 standard [3, 4]. The ability to program in these languages is implemented in the applications OpenPLC [5], ISaGRAF, [6], CoDeSys [7] and others. The program that is loaded into the PLC is a software implementation of the system object control algorithm.

One of the classes of control systems are integrated systems. At least two subsystems in such a system have a common element that performs different functions in these systems [8]. For example, in one system – the function of a control unit, and in the second – a control object. This interaction of subsystems makes it possible to create systems with a control hierarchy, with the help of which the structure and parameters of the system are adapted to changes in the goals of their functioning and the parameters of external influences on the system [9].

2. Problem statement

An element of interaction between subsystems within the control hierarchy is the complex of controlled and control automata. The lack of models and methods for programming such a complex in programming languages according to the IEC 61131-3 standard is an urgent unsolved problem.

Object of research: the process of modeling the functional structure and programming of integrated PLC-based control systems in an application environment that supports languages according to the IEC 61131-3 standard.

Purpose of the work: to develop a methodology for designing standard elements of integrated systems with software implementation of control algorithms in the languages of the IEC 61131 - 3 standard.

3. Review of the literature

The starting point for writing a control program for a control system is the model of the system's control unit. When creating such a model for discrete and hybrid control systems, the formalism of control and operational automata is widely used [2], [10–13].

In the model of a classical control finite automaton there are three sets (inputs, outputs and states), an element of the set of states “initial state” and two functions (transitions and outputs). The elements of these sets are binary. Each element of these sets can take the values “active” or “passive”. At each moment of time, only one input, one output and one state are active. The activity of the output coincides in time with the activity of the state. The action in the active state is independent of the input whose activity caused the state to activate. Such a model is simple but modeling complex system control algorithms with its help leads to an increase in the dimension of this model. In addition, the classical model does not provide for structural adaptation of the machine during its operation.

The works [14], [19] describe a model of an automaton with an additional set of controls and a set of functions of the automaton in its states. An element of a set of controls determines the possibility of the automaton exiting the current state under various control options, which makes it possible to change the trajectory of state changes during the operation of the system and thereby implement the structural adaptation of the automaton. An element of the set of functions of an automaton in its state describes the activation function of the state, output functions and structure. However, such an automaton does not form an integrated system.

Models of operating machines are classified by location in the system in relation to the control machine (input and output) and by the type of operation performed [10]. The input operational automaton forms the input (element of the set of inputs) of the control automaton. The logic of the operation performed in this case determines the meaning of the input activity. For example, the activity of the “Temperature above normal” input requires different control actions (outputs) at different standards. Thus, by changing the “norm” parameter of the operating machine, we influence the control logic. The output operational automaton generates an impact on the control object based on the activity of the output (an element of the set of outputs) of the control automaton. The parameters of this influence can serve as an object of control by a higher-level control machine.

The work [8] describes a model of an element of an integrated system, consisting of two interconnected automata CA1 and CA2 in combination with input operational automata IOA1, IOA2 and output operational automata OOA.

The element represents a two-level integrated system. In this system, the automaton CA1 in the lower-level subsystem is the control automaton, and in the upper-level subsystem it is the object of control by the automaton CA2. The automaton CA1 receives events X_1 from the input operational automaton IOA1 and the control option C_1 from the superior automaton CA2. In turn, the automaton CA1 generates control signals Y_{01} for the output operational automaton OOA and information signals Y_{12} for the automaton CA2. Automaton CA2 receives events X_2 from automaton IOA2 and control option C_2 from a higher level of the hierarchical system. In addition, the machine CA2 generates control signals C_1 and information signals Y_{23} . At the same time, work [8] does not reflect the capabilities of controlling operational machines of an element of an integrated system and the issues of implementing this element of an integrated system in the environment of programming applications in languages according to the IEC 61131-3 standard. These languages are Ladder Diagram (LD), Function Block Diagram (FBD), Sequential Function Chart (SFC), Structured Text (ST) and Instruction List (IL) [3].

In this work, the OpenPLC application [5] was used to implement the system control algorithm in software. This is a freely distributed software product that allows programming in all languages according to the IEC 61131-3 standard. The application contains OpenPLC Editor and OpenPLC Runtime components. The OpenPLC Editor component is a programming environment and contains a logic controller with which you can simulate the process of executing a user program. The OpenPLC Runtime component allows you to transfer a program to a PLC, which can be either an industrial controller of the Controllino family or popular microcontroller boards of the Arduino, Raspberry, ESP32, STM32 and others [15] – [18]. To perform the transfer, the PLC must be connected to the computer via a USB port.

4. Materials and methods

By a controlled automaton we mean an automaton CA1, which, with the help of another automaton CA2, can change at least one element of its tuple. For example, block some inputs [8], change the initial state and others. It is possible to change several or even all elements of a tuple at the same time. Such changes, as a rule, will lead to changes in the results of the machine's operation. The automaton CA2, which changes the elements of the tuple of the automaton CA1, will be called the manager. In general, a controlled automaton can receive control from more than one control automaton. Conversely, a control automaton can control more than one automaton.

The control signal generated by the control automaton CA2 can initiate changes in the input operating automata. For example, changes in the thresholds of signals that form events (inputs) of the machine. The purpose of such changes may be the parametric adaptation of the system control device.

The set of machine inputs characterizes the observability of processes in the control object. The point of blocking individual inputs of an automaton in a certain state is to change the number of control cycles performed in the automaton (possible paths in the graph of the automaton), and therefore the control functionality. Changing the initial state of the machine can reduce the time the system reaches a rational control cycle.

The set of outputs of the machine characterizes the controllability of the object. For a lower-level automaton this is a system object, and for other automata it is the underlying operating and control automaton. In binary control automata, control of outputs comes down to blocking/unblocking them, in non-binary ones, also to changing the parameters of influences (level, nomenclature, time interval).

Blocking the activity of the state of the machine is equivalent to blocking all inputs that lead to a transition to this state and blocking actions in it.

Thus, controlling the elements of sets of an automaton tuple changes its functions within wide limits. The subject of such control is the CA2 automaton (Fig. 1). The outputs of this control automaton in its active state determine the control vector of the automaton CA1, including the initial state of CA1, activity in the active state and paths of transition from this state to others. Elements of the control vector can be discrete (allow/block an element of the tuple set of a controlled automaton) or analog types. In the latter case, the value of the control element can be considered, for example, as a threshold for the formation of an input event by an input operational automaton of a controlled automaton, or as a coefficient by which the amplitude of the output signal of the output operational automaton is multiplied, or as the duration of output activity, or as a binary code, in which single outputs serve resolution of individual outputs of a subset of outputs of the controlled automaton in a given state.

The structure of the control automaton CA2 specifies the trajectory for changing the controls of the automaton CA1 within a certain cycle of object control. These trajectories, in turn, can be the object of control at the next level of the control hierarchy in the system. Typical control change trajectories provide expansion/narrowing of functionality, decreasing/increasing the number of stages in control cycles, decreasing/increasing the number of unlocked inputs of the lower-level machine.

The block diagram of the modernized typical element of the integrated system is shown in Fig. 1.

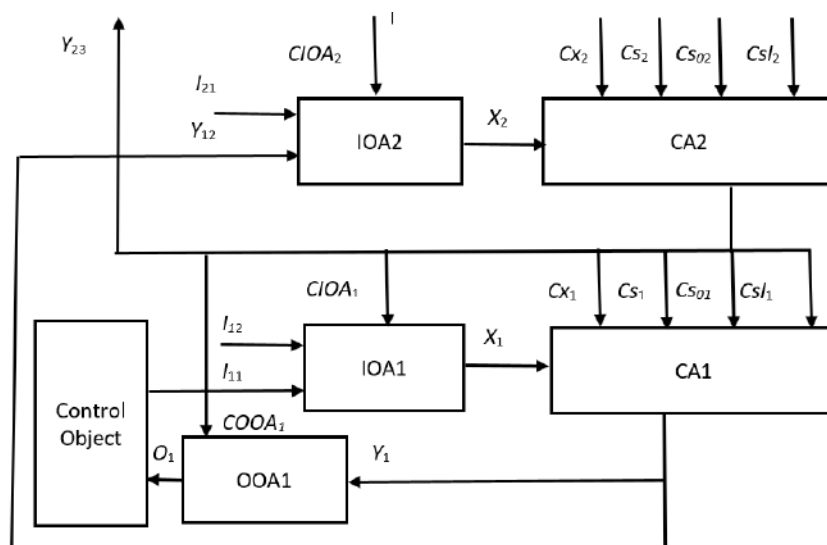


Figure 1. Block diagram of a typical element of an integrated system

The diagram shows control automata of the first (CA1) and second (CA2) levels, input operational automata of the first (IOA1) and second (IOA2) levels, output operational automaton of the first level (OOA1) and a system control object. The system control object responds to influences O_1 by changing parameters I_{11} . The IOA1 automaton uses the parameters of the object I_{11} and the parameters of the external environment I_{12} to form the X_1 inputs of the CA1 automaton. In turn, the automaton, under the influence of inputs X_1 , changes its state and generates impact outputs Y_1 for the automaton OOA1 and information outputs Y_{12} for informing the automaton CA2. The CIOA1 input controls the parameters of the IOA1 automaton, the COOA1 input controls the parameters of the OOA1 automaton. And the inputs C_{x1} , C_{s1} , C_{s01} , C_{s11} control the structure of the CA1 automaton. All of the listed control signals are generated as outputs of the CA2 machine, which also has similar control from the overlying machine.

The choice of the program organization unit (POU) for implementing TEIS elements in industrial controller programming languages was made based on the following considerations:

1. The IOA inputs receive the results of measurements of object parameters. The IOA machine, as a rule, performs the operation of data conversion, calculation of mathematical expressions, and checking the satisfiability of logical conditions. In this case, the IOA can be implemented as a function or function block in the ST or FBD programming languages. And the control signal CIOA will have the meaning of a scaling factor or a logical condition element. For example, for the IOA machine, which generates input x_1 "Temperature is normal", control consists of changing the parameters of this norm.

2. Control automata in the control program are implemented as functional blocks in the SFC language: the state of the automaton corresponds to the steps of the program; machine inputs – transition conditions; Several options for exiting the state correspond to alternative branches.

3. The input control logic C_{x1} is implemented at the "program" POU level. In this case, each input of the control device is supplied to the corresponding input of the machine only if there is permission from the C_{x1} logic, which is implemented using the POU of the functional block.

4. State blocking logic C_{s1} boils down to blocking inputs for all possible transitions to this state.

5. To implement the logic C_{s01} of changing the initial state, additional inputs are added to the control machine to transition from state s_0 to another initial state, which is specified by control C_{s01} .

6. The logic of actions in a state may depend on the input through which the state of the machine was activated. And this logic can be controlled by specifying control of C_{s11} from the higher-level machine. But this significantly complicates the management program and requires separate consideration.

7. The OOA automaton is implemented as a set of actions in each step of the corresponding functional block of the control automaton. These actions in program steps correspond to the outputs of the machine. For example, such as the result of assigning certain numerical values to output variables, which can be interpreted as: scaling factor of the output amplitude for a given state; N – bit binary code that controls the activation (one in the code) of N digital outputs of the control device; action qualifier parameter in this step.

Thus, TEIS elements can be mapped to a set of POUs of the type program, function and functional block and implemented in programming languages of the IEC 61131-3 standard.

5. Experiments

Let's consider the implementation of an element of the integrated system (Fig. 1) using the example of automata CA1 and CA2, specified by the graphs shown in Fig. 2. The automaton CA1 has sets of states' $S = (s_0, s_1, s_2, s_3, s_4)$, inputs $X = (x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18})$ and outputs $Y = (y_0, y_1, y_2, y_3, y_4)$. Moreover, output y_0 is active in state s_0 , output y_1 is active in state s_1 , and so on. As can be seen from Fig. 2a, the paths of four cycles of system object control pass through state s_0 in the automaton CA1. This is path 1: $s_0-s_1-s_3-s_4-s_0$; path 2: $s_0-s_1-s_4-s_0$; path 3: $s_0-s_2-s_3-s_4-s_0$ and path 4: $s_0-s_2-s_4-s_0$.

The control cycle changes when activity passes through state s_0 . The control strategy is to move the activity of the machine in the sequence: path 1 - path 2 - path 3 - path 4. Then the sequence is repeated again. The control strategy is formed by the automaton CA2, in which there is only one path $s_0-s_1-s_2-s_3-s_4-s_0$. The state change on this path occurs at the moment of completion of the next cycle of the automaton CA1. That is $x_{21} = x_{22} = x_{23} = x_{24} = x_{25} = \text{Change}$.

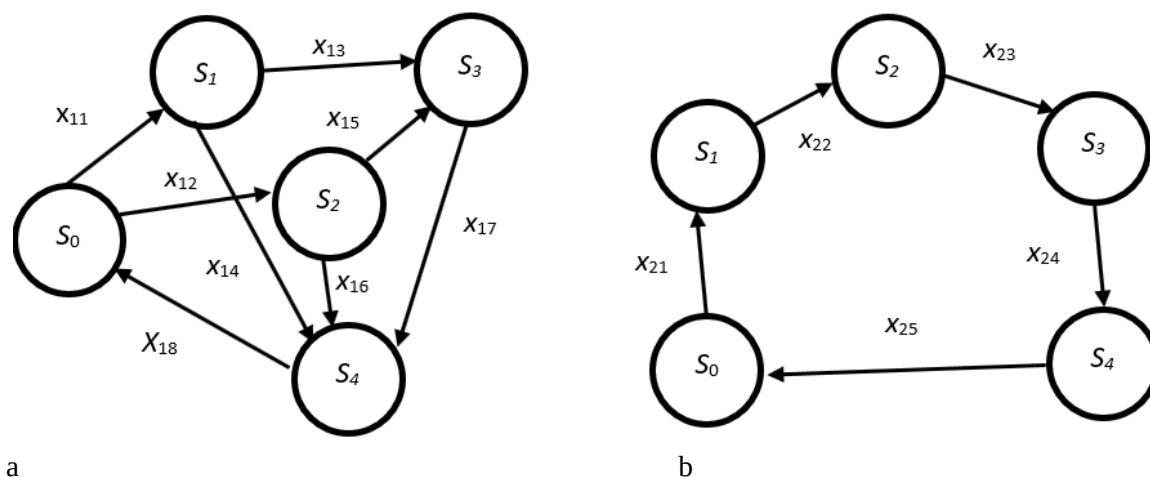


Figure 2 — Graph of controlled CA1 (a) and control CA2 (b) automaton

The task of controlling the automaton CA1 from the side of the automaton CA2 is to limit the inputs of the automaton CA1 so that it implements only one given path in the current control cycle. A graphical FBD representation of an integrated system element program in the OpenPLC application environment is shown in Fig. 3.

The program contains standard functional blocks EQ, R_TRIGO, AND, user functional blocks of types CA1, CA2 and contr. Graphical representations of which are shown in Fig. 4-6.

At the initial moment, both automata are set to state S_0 (variables $stateCA1=0$ and $stateCA2=0$), a pulse arrives at the Change input of automaton CA2, which sets automaton CA2 to state S_1 . As a result, the count block generates permissions/prohibitions at inputs $e_1 - e_8$, which correspond to strategy “1”. Allowed events from the set $x_1 - x_8$ arrive at the inputs of the automaton CA1 and cause transitions of this automaton to other states in the control cycle of the system object. At the moment of completion of this cycle, a new impulse is formed at the *Change* input and the described cycle is repeated for strategies “2”, “3”, “4”, “0”, “1”, “2”...

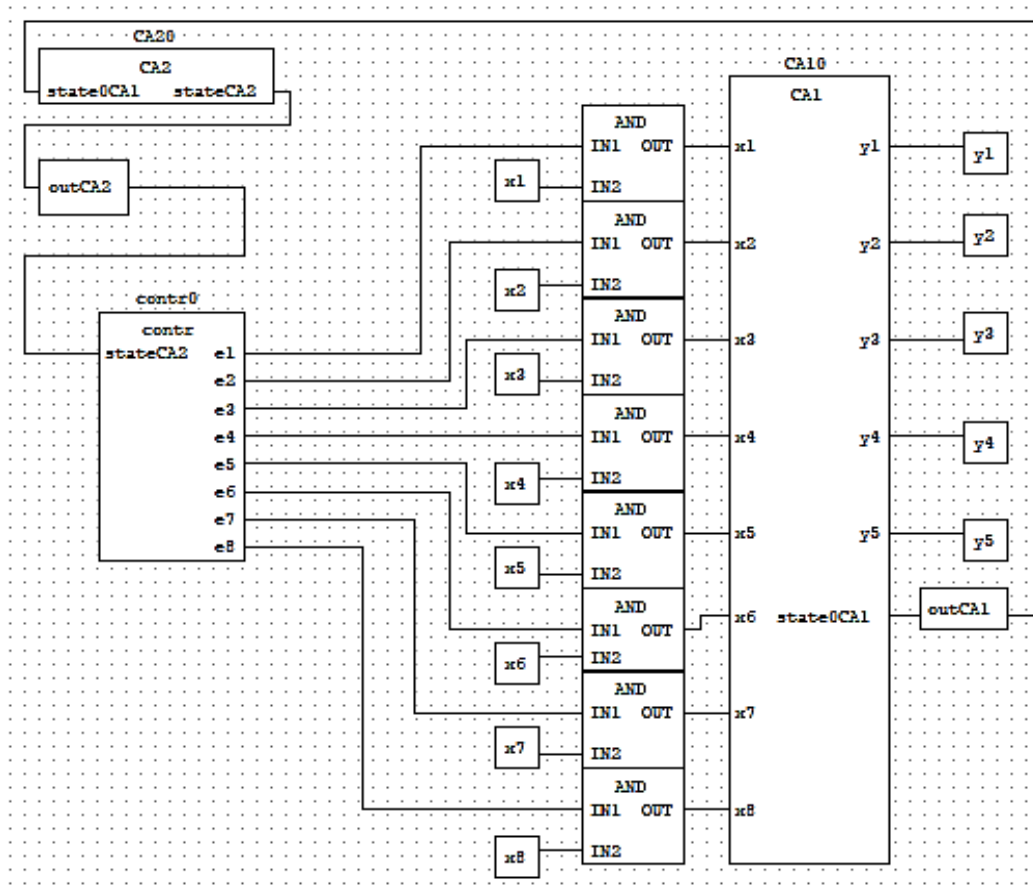


Figure 3 – Graphical FBD representation of the integrated system element program.

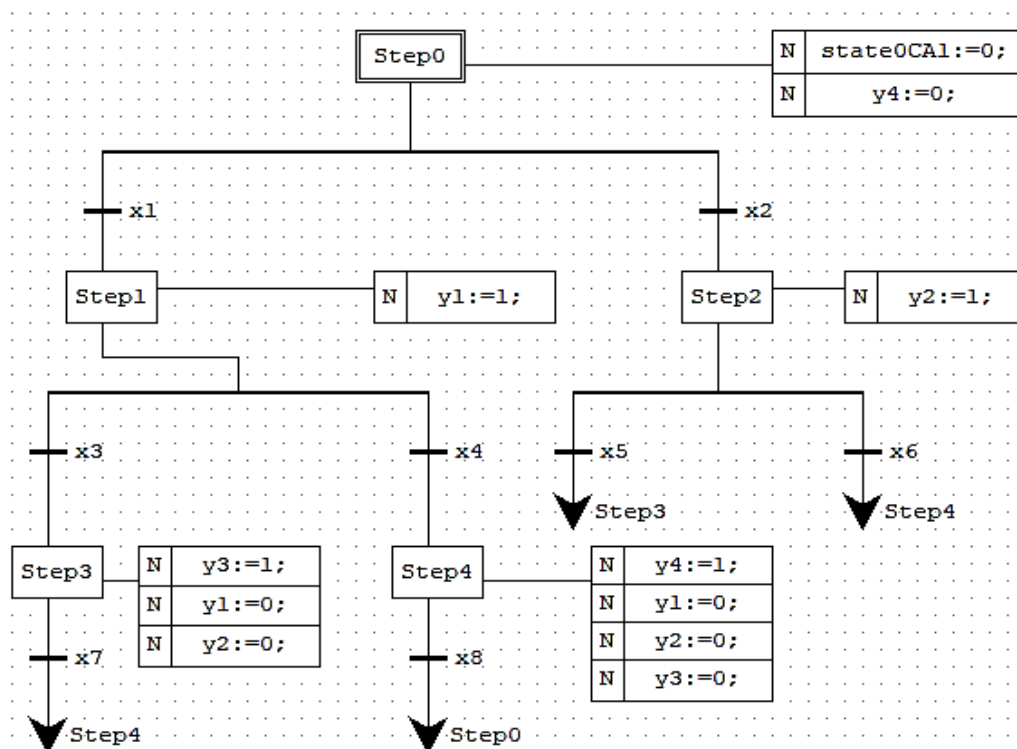


Figure 4 – SFC functional block of the CA1 machine

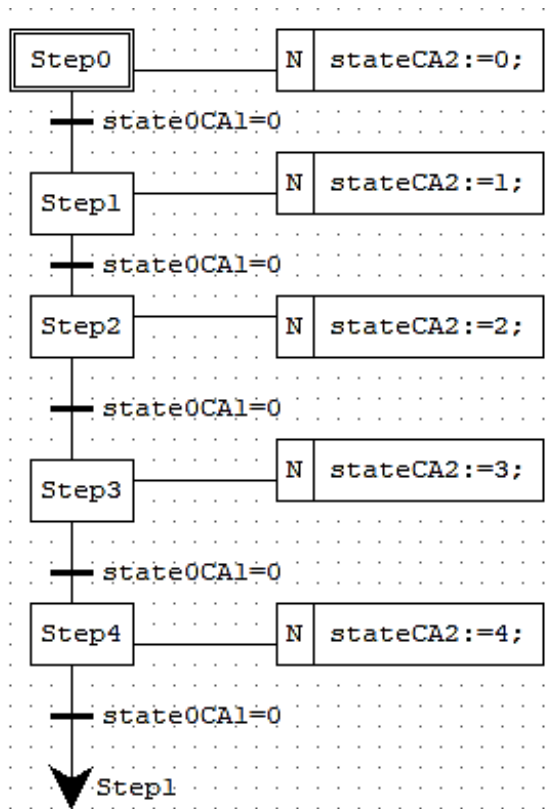


Figure 5 – SFC functional block of the CA2 machine

```

CASE stateCA2 OF
  0: e1:=0; e2:=0; e3:=0; e4:=0; e5:=0; e6:=0; e7:=0; e8:=0;
  1: e1:=1; e2:=0; e3:=1; e4:=0; e5:=0; e6:=0; e7:=1; e8:=1;
  2: e1:=1; e2:=0; e3:=0; e4:=1; e5:=0; e6:=0; e7:=0; e8:=1;
  3: e1:=0; e2:=1; e3:=0; e4:=0; e5:=1; e6:=0; e7:=1; e8:=1;
  4: e1:=0; e2:=1; e3:=0; e4:=0; e5:=0; e6:=1; e7:=0; e8:=1;
ELSE
  e1:=0; e2:=0; e3:=0; e4:=0; e5:=0; e6:=0; e7:=0; e8:=0;
END CASE;

```

Figure 6 – ST functional block contr for controlling the input permissions of the CA1 machine

Shown in Fig. 4 SFC functional block of the automaton CA1 describes the structure of transitions and actions in states according to the graph of the automaton in Fig. 2a.

Similar problems are solved by the SFC functional block of the CA2 machine shown in Fig. 5. In this example, the automaton CA2 is uncontrollable, but if another control level appears in the system, then this automaton can also be an object of control from the upper level.

ST functional block contr (Fig. 6) controls the permissions of the inputs of the CA1 machine in the control strategies described above. With the addition of a new strategy, a new element of the CASE statement must be added. For example, if all inputs are enabled, then control can be performed according to any of the cycles described above, depending on the sequence of occurrence of events at inputs $x_1 - x_8$.

In Fig. Figure 7 shows an example of SFC - a model of a controlled automaton CA2 with two control strategies of the CA2 level.

In the first strategy of level CA2, the change of strategies for controlling the automaton CA1 is carried out in the order: path 1 - path 2 - path 3 - path 4. And in the second strategy of level CA2 - in the reverse order. The first strategy uses the Change input permission, and the second one uses Change1. These signals are generated at the outputs of the CA3 machine of the third level in the system control hierarchy.

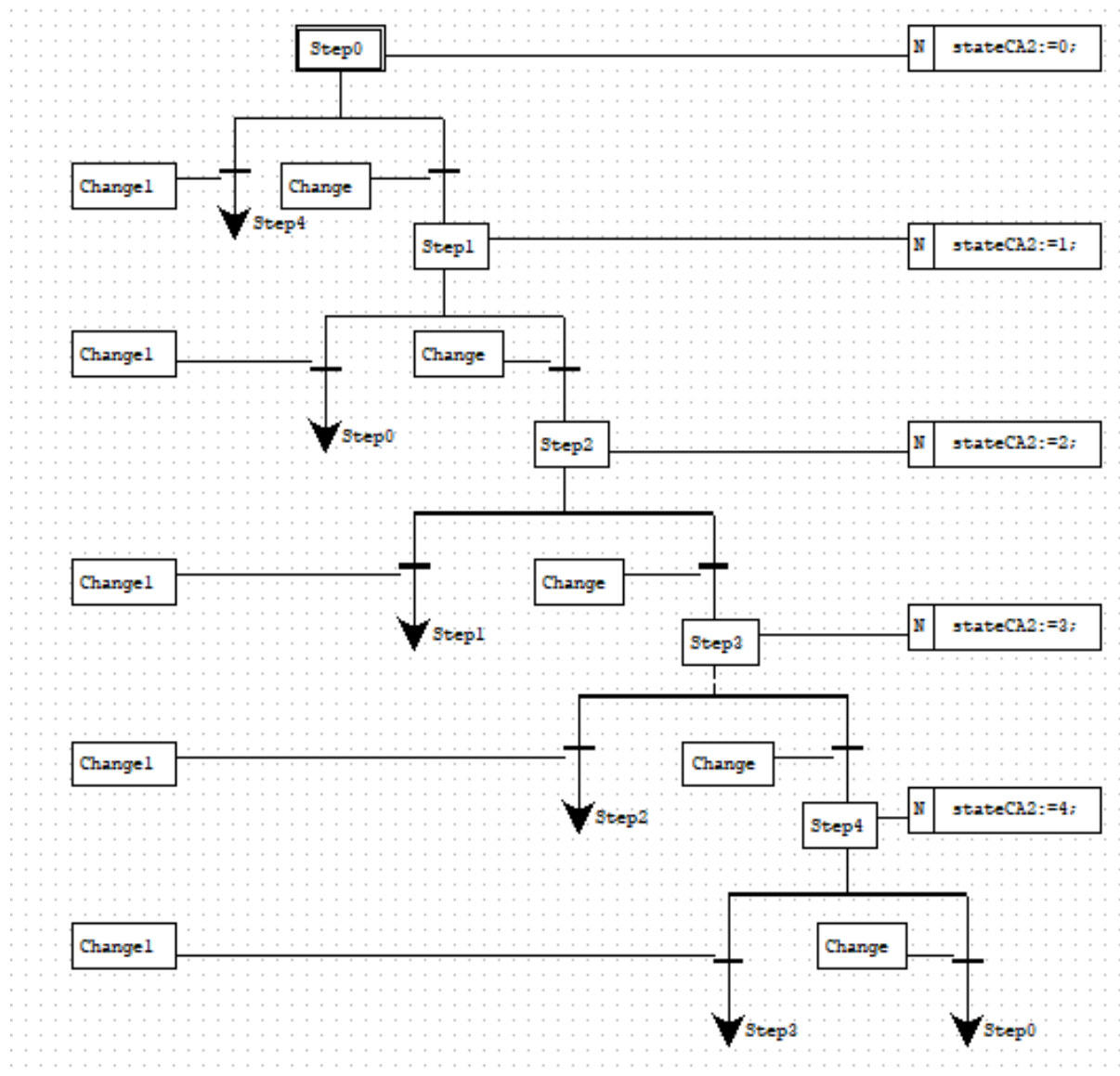


Figure 7 – SFC model of controlled automaton CA2 with two control strategies

6. Results

Testing of the developed project for a typical element of an integrated system was carried out both using the built-in debugger of the OpenPLC Editor application, and by transferring the project code to the Arduino board. In the latter case, the simulation of the inputs x1 - x8 of the CA1 machine was carried out using buttons, and the outputs - using light-emitting diodes. The test results confirmed the functionality of the integrated system declared in the project, such as managing the cycle of actions in the system and managing the typical sequence of changing control cycles.

Thus, the typical structure of an element of a hierarchical integrated system is implemented in programming languages according to the IEC 61131-3 standard in the OpenPLC application environment. This structure can be useful, for example, when forming a sequence of diagnostic procedures that is adapted to the results of previous measurements of the parameters of a system object.

7. Discussion

Modeling adaptive behavior reflects the relationship between two adjacent levels of control in a hierarchical system. The object of adaptation can be almost any element of the tuple of control automata and the parameter of the operation performed by the operating automata of the system.

The model uses expressive means of graphical programming languages for industrial controllers. The POU's of this model have two levels. TEIS operating and control machines are presented at the

lower level in the form of functions or functional blocks. The second level, the program level, displays the logic of the relationship between these elements. Example experiments showed the practical value of the proposed methodology for modeling adaptive control in an integrated hierarchical system.

The graphical representation of the model in the OpenPLC application describes the control logic in the system at the functional level, at the same time allowing translation into executable code and transfer of this code to a wide range of popular microcontroller boards.

Modeling of integrated systems in the OpenPLC application as part of the master's training process allows you to study the programming of industrial controllers using free software and inexpensive popular microcontroller boards.

8. Conclusions

The current problem of modeling adaptive behavior in an integrated hierarchical system and programming such behavior using the languages of the IEC 61131-3 standard has been solved.

The proposed methods for controlling the parameters of operating and control automata as part of the TEIS expand the functionality of adaptive behavior in the system.

The practical value of the proposed methodology for implementing adaptive behavior lies in the possibility of using TEIS implemented in the programming languages of industrial controllers.

The conducted research is limited to modeling the behavior of discrete systems based on binary finite state machines. Prospects for further research include studying the possibility of using the proposed models and methods to build integrated hybrid and cognitive systems using both binary and non-binary automata [20].

9. Acknowledgments

The work was supported by the state budget research work of the National University “Zaporozhzhia Polytechnic” “Intelligent information technologies for data processing” (state registration number 0118U100063).

10. References

- [1] Parr, E. A. Programmable Controllers. An engineer's guide / E. A. Parr. 3rd ed. – Oxford: Newnes, 2003. – 429 p.
- [2] Hooper, J. F. Introduction to PLCs. Second Edition. Published By: Carolina Academic Press. 2006. – 120 p.
- [3] IEC 61131-3, Revision 3.0, February 2013 - Programmable controllers – Part 3: Programming languages. Published By: International Electrotechnical Commission (IEC). – 468 p.
- [4] Karl-Heinz John –IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Aids to Decision & Making Tools. 1995, Springer Verlag.
- [5] OpenPLC Overview – Autonomy (autonomylogic.com) [Electronic resource]– Access mode: <https://autonomylogic.com/docs/openplc-overview/>
- [6] ISaGRAF – [Electronic resource]– Access mode: [http://www.isagraf.com](http://www.isagraf.com;);
- [7] CoDeSys – [Electronic resource]– Access mode: <http://www.3s-software.com>;
- [8] Poliakov, M. O. Interoperability of Integrated Hierarchical Systems / M. O. Poliakov, S. O. Subbotin, O. M. Poliakov // System technologies: regional interuniversity collection of scientific works. – Dnipro, 2021. – No. 2 (133). – pp. 68–78. DOI 10.34185/1562-9945-2-133-2021-08
- [9] Poliakov, M. Behavior classification of control unit of systems//Radio Electronics, Computer Science, Control. 2022. № 3, pp. 183-195 doi 10.15588/1607-3274-2022-3-17.
- [10] Glushkov, V. M. Sintez tsifrovyykh avtomatov. Moscow, Fiz-matizdat, 1962, 476 p.
- [11] Hopcroft, J. E. Introduction to automata theory, languages, and computation / by John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. -- 3rd ed. //Addison Wesley, 2006. – 550 p.
- [12] Solov'ev V. V., Klimowicz A. S., Structural models of finite-state machines for their implementation on programmable logic devices and systems on chip, Journal of Computer and Systems Sciences International, 2015, Vol. 54, № 2, pp. 230–242. doi.org/10.1134/s1064230715010074

- [13] Stéphane Lafortune. Discrete Event Systems: Modeling, Observation, and Control, Annual Review of Control, Robotics, and Autonomous Systems, 2019, No. 2:1, pp. 141– 159. doi.org/10.1146/annurev-control-053018-023659
- [14] Polyakov, M. A. Finite automata with non-binary elements of sets / M. A. Polyakov, I. A. Andrias // System technologies: regional inter-university collection of scientific works. – Dnipropetrovsk, 2019. – № 2 (121). – P. 85–94.
- [15] Arduino - Home [Electronic resource]– Access mode: <https://www.arduino.cc/>
- [16] Raspberry Pi for industry [Electronic resource]– Access mode: <https://www.raspberrypi.com/for-industry/>
- [17] The Internet of Things with ESP32 [Electronic resource]– Access mode <http://esp32.net/>
- [18] STM32 32-bit Arm Cortex MCUs. Overview. [Electronic resource]– Access mode: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>
- [19] Poliakov M., Subbotin S., Poliakov O. Performance indicators of models of non-binary control automates, Experience of Designing and Application of CAD Systems: IEEE 16th International Conference. Lviv, 22–26 Feb. 2021, proceedings, pp. 38–42. <http://doi.org/10.1109/CADSM52681.2021.9385220>
- [20] Poliakov, M., Subbotin S., Poliakov O. Set-theoretical FSM models activity subsystem for Cognitive Control Systems. // In Proceeding of the 15th International Conference "The Experience of Designing and Application of CAD Systems" (CADSM), (26 February - 2 March, 2019, Polyana-Svalyava (Zakarpattya), Ukraine). P. 1/1 – 1/4.

Combat drone swarm system (CDSS) based on Solana blockchain technology

Sviatoslav Vasylyshyn^{1*†}, Ivan Opirskyy^{1†},

¹ Lviv Polytechnic National University, 12 Stepana Bandery str, Lviv, 79000, Ukraine

Abstract

This paper examines the application of Solana blockchain technology in managing combat drone swarms, focusing on its ability to enhance security, reliability, and transparency in military and disaster relief operations. We present a model, UAV Swarm Net, which leverages the high throughput and low latency of Solana to ensure real-time decision-making and robust communication among UAVs. Our analysis includes a comparative study with traditional systems, emphasizing the advantages of blockchain in scalability and security. We conclude with real-world applicability, regulatory challenges, and future directions for integrating emerging technologies. Blockchain Swarm Management can help register each drone on the blockchain platform. Digitally signed transactions, data provenance, and a consensus mechanism can immediately identify and nullify data that has been corrupted between drones. Survivability and overall combat effectiveness enable it to complete difficult tasks such as swarm confrontation and forest firefighting.

Keywords

Blockchain, combat, Solana, drones, network, security

1. Introduction

In the context of cooperative operations of UAV clusters, information connection channels must be established between UAVs to form a mobile self-organizing (ad hoc) UAV network. Compared to traditional mobile ad hoc networks, the drone network features fast node movement, strong interference from the working environment, long working hours, and robust real-time performance. These characteristics introduce more complex security issues to the cooperative operations of the drone cluster. Data can easily become a target of hacker attacks. Once hackers intercept communication or hijack drones, it can impact the swarm combat environment and even result in serious consequences, such as leaking state secrets and disrupting social order [1]. For example, in January 2021 the incident of hundreds of drones falling in Chaotianmen the U.S. Department of Justice's update of the drone safety law indicates that drone safety plays an important role in future security risks, and the safety protection of drones is imperative [2].

The Seventh International Workshop on Computer Modeling and Intelligent Systems (CMIS-2024), May 3, 2024, Zaporizhzhia, Ukraine

✉ sviatoslav.i.vasylyshyn@lpnu.ua (S. Vasylyshyn); iopirsky@gmail.com (I. Opirskyy)

ORCID iD 0000-0003-1944-2979 (S. Vasylyshyn); 0000-0002-8461-8996 (I. Opirskyy)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

To enhance the robustness and security of UAV systems, in recent years, researchers have begun to try to use blockchain technology to solve the information security problem of UAV collaborative operations. Blockchain is an emerging technology that originated from Bitcoin. It connects data blocks in chronological order to form a distributed ledger that cannot be tampered with or forged. According to the degree of openness, blockchain can be divided into three types: public chain, alliance chain, and private chain. Among them, only the private chain limited to private members has both high transaction throughput performance and a high degree of decentralization if we compare it with UAVs in warfare. Compared with public and alliance chains, it is more suitable for integration with drones to create a safe and reliable communication network [3].

This article using the private chain Solana as the basic framework, explores solutions for applying the core technologies of blockchain (such as peer-to-peer networks, distributed ledgers, smart contracts, etc.) to drone cluster systems to help drones achieve collaborative operations. The data sharing and collaborative decision-making functions in the blockchain are used, and the security features of the blockchain itself are used to deal with the security issues of drone collaboration so that the drone cluster can operate normally in confrontational airspace and enhance the security of the drone system.

Section 2 of this article introduces related work, including the security issues faced by drone systems, the working principle of the private chain Solana, and existing solutions that combine blockchain and drones. Section 3 introduces the model UAV-Swarm Net proposed in this article, including network structure and transaction process. Section 4 explains how the UAV-Swarm Net model responds to the security challenges of UAV systems from three aspects: data confidentiality, integrity, and availability. The final summary discusses the limitations caused by the combination of the two technologies and the issues that need to be further studied in the future.

2. Related work

2.1. UAV system safety

A typical UAV swarm system (Unmanned Aircraft System, UAS) is mainly composed of a UAV swarm, a ground control base station (Ground Control Station, GCS), and a communication network used to exchange information between them. It is a complex system [4]. In combat, the increasing complexity of tasks and the improvement of the intelligence level of UAVs have led to the development of UAV system control methods from traditional radio remote control and automatic program control to autonomous collaborative control of UAV clusters. The difference between automatic control and autonomous control is that automatic control involves the system performing tasks according to instructions, whereas autonomous control requires the drone itself to make decisions at critical moments. Compared with simple UAS, UAV systems are facing major challenges in terms of safety, swarm intelligence, and cluster complexity. The complexity and practicality of UAV systems have also triggered a research boom in both corporate and academic circles. According to statistics, in 2020 alone, close to 20,000 papers were published with the keyword "UAV" [6]. Safety is one of the main concerns of UAS. Computers and GCS can communicate and share information in real time, and the information sent and received must be reliable to make further real-time and effective decisions and secure collaboration. The National Institute of Standards and Technology (NIST) has formulated the data security policy [7] it is specified as "CIA", which is the system's requirements for confidentiality (Confidentiality, C), integrity (Integrity, I), and availability

(Availability, A). Among them, confidentiality refers to the protection of data. Humans and machines must not disclose sensitive information to any unauthorized person, entity, or process; integrity refers to the reliability of data, and the drone system should have the ability to resist hackers from modifying or damaging data; availability refers to data and services Accessibility should ensure that the UAV system can access and use information timely and reliably [8]. A complete UAV security protection solution should combine active and passive protection methods. Active protection refers to the use of various information encryption Technology eliminates security threats and increases the difficulty of hacker intrusion. For example, literature [9] applies security encryption methods to the WiFi 802.11 access point in the drone telemetry box and the entire communication path to improve the confidentiality of data transmission. Passive protection refers to making appropriate responses after detecting security threats to avoid continued adverse effects. For example, the literature [10] uses Bayesian game theory to detect drone network intrusions and improve drone network performance. At the same time, security detection is implemented; literature [11] uses middleware to construct an encrypted channel to detect DoS intrusions by anonymous attackers to interrupt the hacker attack channel.

However, the above studies are all based on the underlying defense of the physical layer or link layer, only focus on the local security protection of drones, and lack consideration of the overall network architecture. To this end, this article will use the blockchain and drone networks to integrate multiple security technologies such as member authentication, encrypted channels, and asymmetric encryption algorithms, to establish a safe and reliable communication network for drone collaboration and effectively respond to the "CIA" security challenges.

2.2. Solana blockchain network

Solana is an open-source blockchain platform that focuses on scalability, speed, and security, while maintaining low transaction costs. Its architecture facilitates the creation of smart contracts and decentralized applications. According to the project's white paper, the network's throughput can reach up to 710,000 transactions per second (TPS). Due to its affordability, Solana has gained popularity among users in the DeFi sector. In December 2023, analysts estimated that the number of active addresses in the network exceeded 15.6 million [12]. As of January 4, 2024, the total value of funds locked in smart contracts (TVL) of the project exceeded \$1.3 billion, according to DeFiLlama. At the time of writing, Solana ranks among the top five blockchains by this indicator. The Solana project aims to address the blockchain trilemma, which involves balancing scalability, security, and decentralization. It is built on a hybrid consensus mechanism that theoretically allows the network to operate rapidly while maintaining a high level of decentralization and reliability.

Before we understand how Solana works, let's look at the technologies that support the operational operation of the network:

- the Proof-of-History blockchain synchronization concept;
- Tower BFT is an optimized version of the PBFT algorithm;
- Turbine is a mechanism for transferring data about a block to broadcast records to all nodes;
- Gulf Stream — protocol for transferring transactions without mempool;
- Sealevel — parallel execution of smart contracts;
- Pipelining — transaction processing to optimize the block confirmation process;
- Cloudbreak is a horizontally scalable database of accounts;
- Archivers is a distributed registry repository.

The main feature of Solana is the PoH technology, which functions as a cryptographic clock. Normally, to verify transactions, nodes need to communicate with each other to calculate timestamps and confirm the transaction. The speed of transaction processing depends on the speed of node synchronization.

With the implementation of PoH, the synchronization process is greatly simplified. Nodes do not need to wait for transaction confirmation from each node. Instead, they rely on a cryptographic clock to track events on the blockchain. Turbine breaks the block information into smaller data packets and distributes them randomly to different nodes. With this approach, you can reach up to 40,000 validators. The speed of the blockchain is also increased by Sealevel, a technology used to optimize resources. Transactions scale horizontally across GPUs and SSDs, which should meet the growing needs of the project.

The SHA-256 hashing algorithm is implemented in Solana for cryptographic data protection.

Since the Solana protocol runs on DPoS, validators block SOL to provide a consensus mechanism. There is no hard requirement for the number of coins to run a node, but the cost of confirming blocks can cost up to 1.1 SOL per day. At the same time, powerful equipment is required to participate in the project as a validator. However, DPoS allows you to delegate assets, i.e. to transfer them under the control of a validator. You can see the list of the latest ones here. At the time of preparation of the material, almost 2,000 nodes are working in the network. Profit from staking depends on many factors, including inflation and the total number of "frozen" coins in the network. Digital assets can be withdrawn at any time, but the process takes about three days.

As of January 8, 2024, about 378.5 million SOLs are locked in the protocol deposit contract. Solana's annual percentage rate of return (APY) is 7.2%. The Solana ecosystem also includes liquid staking services such as Lido Finance, Marinade, and Jito. They issue tokens instead of locked cryptocurrency. Assets can be used in DeFi applications.

Terms and APY vary depending on the project:

1. Lido Finance — Lido DAO members choose validators and decide how much cryptocurrency to delegate to them. As a rule, coins are transferred to validators who are long-standing partners of the project. Annual percentage return from staking — 7.5%, according to Solana Compass;
2. Marinade - the project team evaluates the decentralization and performance of validators on its scale and selects the 100 best. They are delegated 60%, and the remaining 40% are distributed according to the results of the community vote. APY is 6.6%;
3. Jito — to interact with the project, validators must apply through the Jito Foundation. The project promises its customers 7% APY.

This article will make full use of these functions to establish a peer-to-peer network UAV-Swarm Net based on Solana to meet the security requirements of UAV systems for data confidentiality, integrity, and availability.

2.3. Existing solutions

The report highlighted that as the autonomous capabilities of drones improve, trust will become a critical focus, and blockchain technology can enhance the integrity and trust of UAS through policies and protocols. The study incorporates the integration of blockchain and drones to provide a framework for commercial drone use. Blockchain technology ensures security, offers identity management, and supports UAS conflict management and flight authorization. Additionally, it aids in air traffic management, addresses data storage challenges, and resolves

the trust issues related to big data in drone cloud systems. Currently, many companies globally are engaged in research and development on the application of blockchain technology in drones. For instance, in August 2018, Boeing partnered with AI provider SparkCognition to develop a drone tracking system and an air traffic management solution based on blockchain technology. In September 2018, IBM applied for a patent that details how blockchain technology can be used to store data related to drone flights, with solutions aimed at facilitating the commercialization of drone applications.

Compared with the industry's exploration of blockchain in the commercial field of drones, the academic community is more focused on using blockchain to solve the security issues of drone systems. In terms of data confidentiality, the [13] proposed a blockchain-based system that protects drones from transmitting location data and prevents line-of-sight obstruction. [14] proposes an intelligent framework based on blockchain technology to defend against leak attacks by establishing an account and a unique identification code for each device. Data integrity In terms of data availability. [15] proposes a distributed solution that uses blockchain technology and cloud servers to ensure UAV communication and data integrity. In terms of data availability [16] uses certificates issued by a central authority to help UAVs resist DoS. Attack; [17] regards each drone in the drone cluster as a node of the blockchain, and each drone saves a copy of the data, effectively preventing blocking attacks.

Among the existing solutions, the applications in the corporate world are oriented to the commercial field and mainly use the traceability of blockchain to supervise drones. However, the academic research on the Solana blockchain in drone collaboration is only at the theoretical level and lacks an understanding of the situation. Exploration of specific technical routes. Therefore, this article will focus on the specific implementation of the integration of the Solana blockchain and drones, defining the network structure designing the transaction process, and providing a complete solution to solve the safety and security issues of drone collaboration.

3. Model establishment

3.1. UAV-Swarm Net network structure

UAV-Swarm Net consists of three roles: Ground Control Base Station (GCS), Master UAV (MasterUAV), and FollowerUAV. GCS is the combat command center in the traditional sense of UAS, which is responsible for controlling UAVs, flight process and trajectory, distributing combat tasks, maintaining [18] the normal operation of communication links, etc. The master drone and its submissive drones form a small drone cluster, and multiple small drone clusters are formed on demand Large UAV clusters. With the improvement of UAV autonomy, more and more control tasks are delegated from the GCS to the master UAV. The master UAV is responsible for controlling the submissive UAVs in its cluster and interacts with the GCS communicates in real-time with the master drones of other clusters and participates in collaborative decision-making while the submissive drones can only perform tasks under the command of the GCS [19] and the master drone, collect intelligence information and report to superiors.

UAV-Swarm Net is built on the blockchain using the network Solana. Its basic network structure is shown in Figure 1. Organization R1 is composed of multiple GCS from different theaters and assumes the responsibility of the network administrator. The initial structure of the network is determined by the network configuration NC1 pre-established by organization R1,

and it is started and managed by the administrator of R1. The small drone cluster participates in the network as an organization (R2, R3, and R4 in the figure represent different clusters respectively) [20]. To identify the identity of the organization members, the certificate is assigned to organization administrators and other network nodes and matches the certificate with the organization of the Solana blockchain network, effectively preventing illegal access to the UAV-Swarm Net network from the outside world.

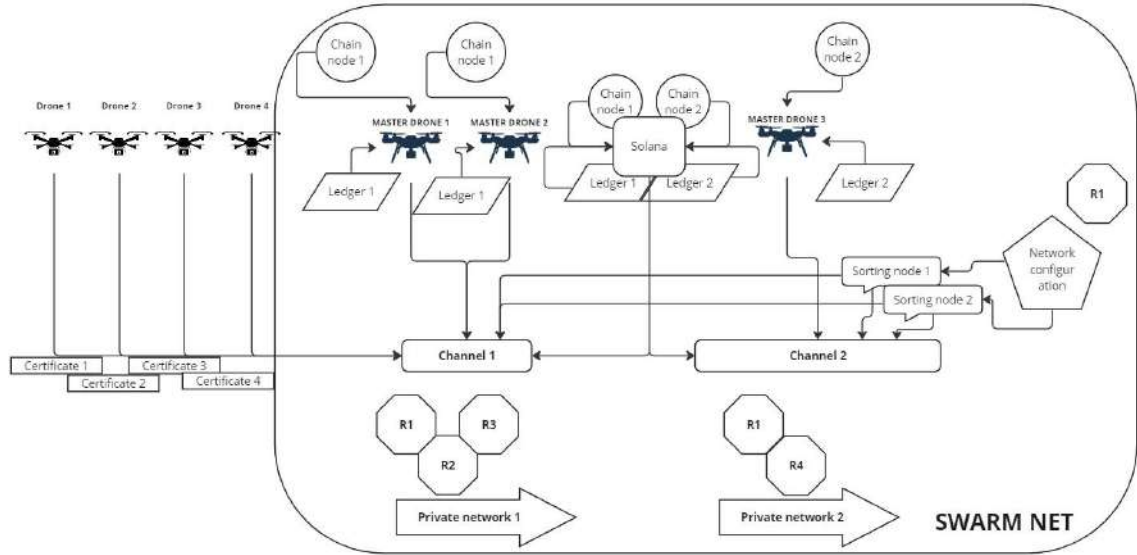


Figure 1: Network structure of UAV Swarm Net

According to the actual needs of drone operations, drone cluster organizations are added to different private networks. Organization R1 joins each private network and monitors the behavior of each drone cluster in real-time as the private network administrator. Internal communication and business isolation rely on channels, and communication between different private networks and different channels is completed through cross-chain channels. Each channel maintains a blockchain ledger, and only organizations that join the channel have the right to access and deploy the ledger and chain code of the channel. In the network shown in Figure 1, organizations R1, R2, and R3 belong to private network 1 and participate in channel 1, and organizations R1 and R4 belong to private network 2 and participate in channel 2. Therefore, organizations R1, R2, and R3 are internal network nodes that have the right to access the ledger and chain code of channel 1, and can also access the ledger data of other channels (such as channel 2) by registering cross-chain services [21]. The master drone participates in the network as a peer node in the organization. All master drones in the channel are deployed with a copy of the ledger. This multi-copy approach effectively avoids single points caused by the hijacking of the drone. Submissive drones participate in the network as blockchain users. They have the certificates of the cluster organizations to which they belong. They access the chain code and ledger deployed on their master drone through the client [22]. The query results of the ledger will be returned quickly to the submissive drone, but the update of the ledger will go through more complex processes such as endorsement and sorting. In addition to the master drone, GCS also assumes the responsibility of a peer node. It participates in multiple channels at

the same time, so it deploys multiple Ledger and chain codes. The ordering node from R1 supports multiple channels at the same time to order transactions and distribute blocks.

3.2. Transaction process

The ledger and chain code together form the core of UAV-Swarm Net. The ledger records all the intelligence information collected by the drone cluster from the start of the network; the chain code [23] is used for data sharing and collaborative decision-making between drones. The cooperative operation process of the UAV cluster is divided into the reconnaissance stage, the decision-making stage, and the execution stage. In the reconnaissance stage, each submissive UAV writes the collected local intelligence information into the account book by accessing the chain code on its master UAV and then distributes it to other peer nodes in the channel, and a shared public operational map is formed. In the decision-making phase, the main drone generates decision proposals based on the target status and real-time situation, and executes its own deployed decision chain code, allowing other nodes in the same channel to participate in collaborative decision-making through endorsement [24]; verified decisions are released to submissive drones through chain code events. The final execution phase is implemented.

The specific transaction process of the reconnaissance phase is shown in Figure 2. The local intelligence information collected from the drone through sensors or cameras is stored in the local database in the form of key-value. Before uploading the information, the drone must first pass the client's CA service to register and obtain the certificate. Subsequently, the submissive drone initiates a transaction proposal to the master drone to which it belongs [25]. The proposal contains the contract identifier to be called, the contract method, and all the intelligence collected from the drone within a period. The master drone after verifying the signature - the proposal is broadcast to other peer nodes in the channel. Through the endorsement of the node, the UAV-SwarmNet network conducts a security appraisal of the intelligence from the drone.

The endorsement node will verify the following contents of the proposal:

1. Verify whether the format of the transaction proposal is complete to prevent information data from being tampered with;
2. Verify whether the transaction proposal has been submitted before to prevent replay attacks;
3. Use MSP to verify whether the submissive drone's signature is valid and determine whether the submissive drone has been invaded;
4. Verify whether the submissive drone is authorized to operate on the channel.
5. After the verification is passed, the endorsement node will input the transaction proposal as a call the parameters of the chain code function are intended to execute the chain code and generate transaction results. Then the transaction results and the signature of the endorsing node will be returned to the master drone as a "proposal response", and finally packaged and sent to the submissive drone. The submissive drone verifies the signature of the endorsement node and compares the proposal results. If the proposal results are consistent and the signature of the endorsement node meets the endorsement policy, the submissive drone will package the data and send it to the ordering node through the master drone. The ordering node sorts the transactions and generates a zone [26]. The block is delivered to the peer nodes in the channel. Eventually, all peer nodes in the channel receive the block and change the world state. The UAV clusters then share intelligence information through the channel to form a

common public operational map, which is the intelligence foundation laid for subsequent decision-making stages.

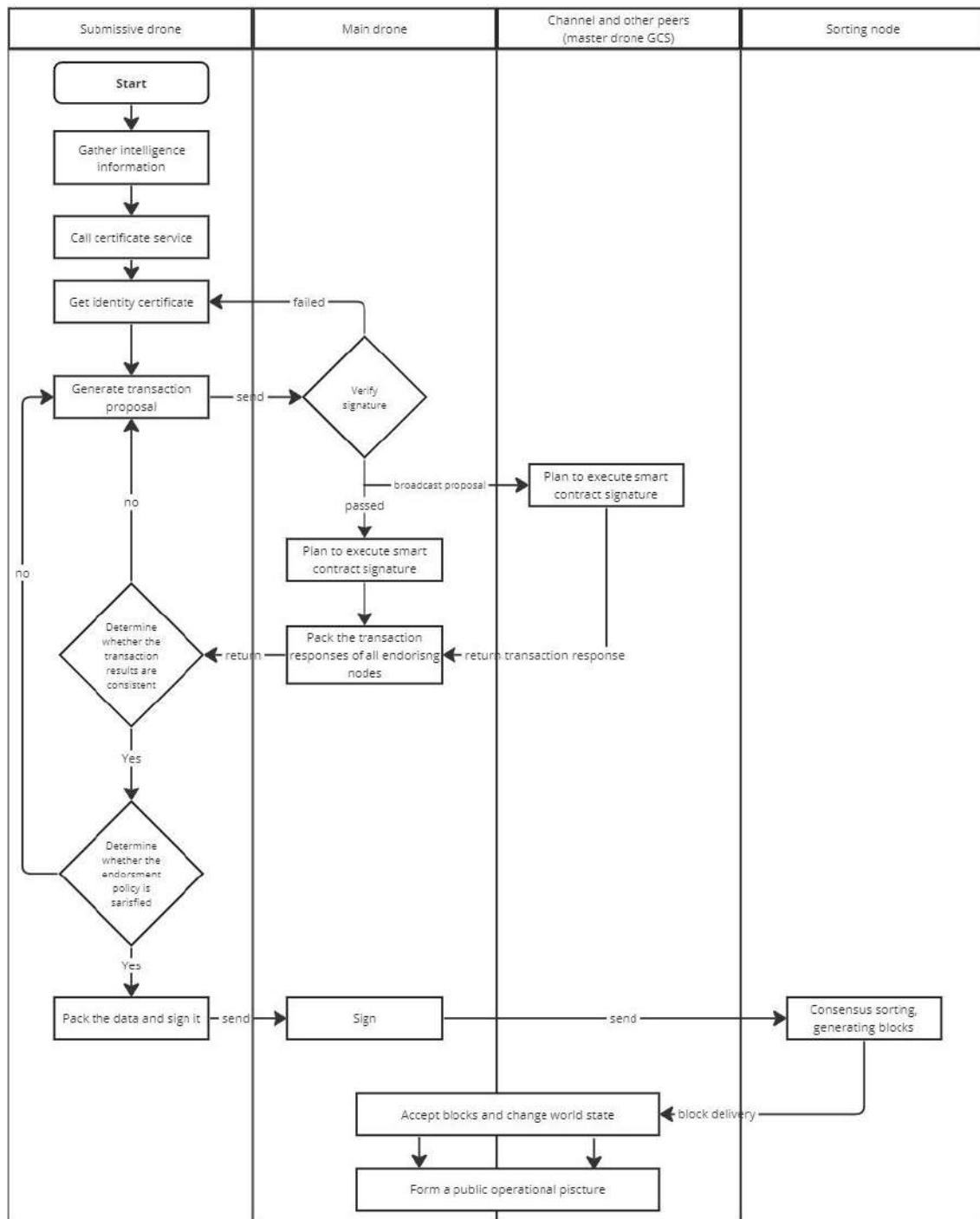


Figure 2: Transaction flowchart in the UAV reconnaissance phase

The chain code for uploading intelligence information contains a judgment on the authenticity of the intelligence. Once the judgment is wrong, UAV-Swarm Net will trace the responsibility based on the signature of the endorsement node, and adopt different punitive measures according to the intelligence level. If the intelligence level is higher, the uploader of the information the submissive drone, its master drone, and other peer nodes endorsing the transaction will be considered spy nodes, removed from the trust list of UAV-Swarm Net, and submitted to GCS for review and recycling. If a drone cluster If the main drone has deactivated its identity due to being implicated or suspended its work due to being invaded, one of the following two methods will be used to maintain the normal operation of the system:

1. GCS will assign a new main drone to take over the submissive drone;
2. The submissive drone with the most votes in the cluster is selected through the election and promoted to the master drone, becoming the new peer node of UAV-Swarm Net and taking over the task of controlling the cluster.

The transaction process in the decision-making phase and execution phase is roughly the same as that in the reconnaissance phase. The main difference is that the transaction proposal carrying the decision is initiated by the master drone and executed by the submissive drone. After discovering the mission target, the master drone conducts the transaction according to the current evaluation of the effectiveness of intelligence information and generates a combat plan. To avoid serious accidents caused by the main drone's decision-making errors, the UAS will assess the security level of the combat plan. Different security levels correspond to different endorsement strategies. For example, the plan to apply for a weapons strike adopts the highest security level and requires the endorsement of all master drones and GCS in the channel. When the decision is determined, the master drone releases a chain code event to the submissive drone. From none, the master drone registers these event types in advance, and when receiving notification of the occurrence of the event, it will immediately execute the corresponding command.

4. Model security analysis

First of all, UAV-Swarm Net provides a good guarantee for the confidentiality of data. UAV-Swarm Net sets network access restrictions based on the Public Key Infrastructure (PKI). Only issued by a specific organization CA can be obtained. Only users with certificates can obtain access to the channel through the verification of the MSP component, which lays a good foundation for the communication environment of the drone. In addition, UAV-Swarm Net uses the TLS transport layer for secure communication between communication nodes, through two-way verification ensuring the identity of both communicating parties and establishing a two-way encrypted channel. Intelligence data is converted into ciphertext using the encryption function when uploading to the chain. Only nodes with the key can interpret the intelligence information, effectively avoiding illegal intrusion into sensitive data.

Secondly, the immutability and multiple copies of the blockchain ledger ensure the integrity of the data, enabling the UAV system to resist improper modification and destruction of data by the outside world. Each main UAV and GCS in UAV-Swarm Net are deployed with an intelligence ledger of the channel to which it belongs, even if a master drone is hacked, the data can be restored by downloading the ledgers of other nodes. Both the intelligence upload of the submissive drone and the decision-making of the master drone require much time. GCS sites are endorsed by other master drones, and large GCS are usually difficult to hack, which greatly improves the reliability and credibility of intelligence and decision-making. The immutability of the ledger means that it is impossible to modify a certain block of intelligence information to change the current state database. Each block on the blockchain saves the on-chain data and the

hash value of the previous block. If the data of the block is tampered with by hackers, it will cause the hash of the current block. The hash value does not match the hash value recorded in the next block so that security threats can be detected and alarms triggered. The traceability of the blockchain also establishes a monitoring mechanism for the drone system, through block transaction information and endorsement Signatures can quickly locate the compromised node and remove it from the trust list to avoid greater losses.

Finally, the decentralized nature of the UAV-Swarm Net network ensures the availability of the UAV system, allowing the UAV to access data information timely and reliably. As long as most nodes in the network do not encounter hacker attacks, the normal operation of the entire system can be ensured. In addition, the UAV-Swarm Net network also has high scalability.

By modifying the network configuration and adding channels, the establishment of small UAV clusters and the formation of large UAV clusters can be flexibly realized. The use of endorsement strategies can meet different levels of security requirements in actual combat missions.

5. Advantages and implementation challenges of the CDSS

The CDSS, leveraging Solana blockchain technology, introduces several groundbreaking advantages over traditional UAV management models. Primarily, the integration of blockchain ensures a decentralized, tamper-proof system that enhances data integrity, confidentiality, and availability. This decentralized approach mitigates the risks associated with centralized control, where a single point of failure could compromise the entire operation. Moreover, the Solana blockchain's capability for high transaction throughput and low latency underpins a responsive and scalable system, capable of handling dynamic, complex swarm operations without sacrificing operational tempo or security.

Advantages:

1. **Enhanced Security and Robustness:** The use of blockchain technology ensures that data shared across the network is immutable and transparent. This significantly reduces the risk of fraudulent activities and enhances the trustworthiness of communication among drones.
2. **Decentralized Control:** Unlike traditional systems that rely on a centralized command structure, the CDSS employs a distributed architecture. This ensures that the system is more resilient to attacks and failures, as there is no single point of failure that could bring down the entire network.
3. **Improved Scalability:** The Solana blockchain's high throughput capabilities enable the CDSS to scale efficiently, accommodating a large number of drones without compromising performance. This is critical for expanding operations and integrating new drones into the swarm without significant overheads.
4. **Real-time Decision Making:** Blockchain's inherent data integrity and the speed of Solana's consensus mechanism facilitate real-time decision making. This allows for rapid responses to changing operational scenarios, significantly enhancing the effectiveness of drone swarm operations.

Despite these advantages, implementing a CDSS based on blockchain technologies is not without its challenges. These include:

1. **Technical Complexity:** The integration of blockchain technology into UAV operations introduces a layer of complexity in terms of system design, development, and maintenance. Ensuring that all components work seamlessly together requires a high level of expertise and resources.

2. **Energy Consumption:** Blockchain operations, particularly those involving consensus mechanisms, can be energy-intensive. This might impact the energy efficiency of drones, reducing operational endurance and necessitating more frequent recharging or refueling stops.
3. **Regulatory and Legal Challenges:** The deployment of blockchain-based drone systems faces regulatory scrutiny, especially concerning airspace regulations, drone operation standards, and data protection laws. Navigating these legal frameworks and obtaining necessary approvals can be a lengthy and complex process.
4. **Adoption Barriers:** Resistance to change and skepticism about new technologies can pose significant barriers to the adoption of blockchain-based drone management systems. Overcoming these requires extensive stakeholder engagement, demonstrations of efficacy, and addressing concerns regarding safety, privacy, and cost.

6. Conclusions

This article defines the UAV-Swarm Net network structure based on the private chain Solana, explores the method of integrating blockchain technology with drones, and establishes a complete security protection solution for the collaborative operations of drone clusters. Blockchain technology is playing an important role while the UAV system improves robustness, security, and scalability, it also has many limitations. For example, the main UAV needs to maintain chain codes and ledgers, which will occupy a lot of memory; the main UAV is responsible for decision-making, endorsement, and controlling multiple tasks such as submissive drones will have an impact on its combat performance; GCS has weak control rights over submissive drones and cannot convey commands promptly. Future research will explore the integration of artificial intelligence to automate decision-making processes within the UAV-Swarm Net, potentially reducing human oversight requirements and enhancing response times. Additionally, advancements in quantum-resistant cryptography are identified as crucial to safeguarding against evolving cyber threats.

References

- [1] S. Cohen, W. Nutt, Y. Sagie, Deciding equivalences among conjunctive aggregate queries, *J. ACM* 54 (2007). doi:10.1145/1219092.1219093.
- [2] J. Cohen (Ed.), Special issue: Digital Libraries, volume 39, 1996.
- [3] D. Kosiur, *Understanding Policy-Based Networking*, 2nd. ed., Wiley, New York, NY, 2001.
- [4] D. Harel, *First-Order Dynamic Logic*, volume 68 of *Lecture Notes in Computer Science*, Springer-Verlag, New York, NY, 1979. doi:10.1007/3-540-09237-4.
- [5] A. Z. Spector, Achieving application requirements, in: S. Mullender (Ed.), *Distributed Systems*, 2nd. ed., ACM Press, New York, NY, 1990, pp. 19–33. doi:10.1145/90417.90738.
- [6] B. P. Douglass, D. Harel, M. B. Trakhtenbrot, Statecharts in use: structured analysis and object-orientation, in: G. Rozenberg, F. W. Vaandrager (Eds.), *Lectures on Embedded Systems*, volume 1494 of *Lecture Notes in Computer Science*, Springer-Verlag, London, 1998, pp. 368–394. doi:10.1007/3-540-65193-4_29.

- [7] D. E. Knuth, The Art of Computer Programming, Vol. 1: Fundamental Algorithms (3rd. ed.), Addison Wesley Longman Publishing Co., Inc., 1997.
- [8] S. Andler, Predicate path expressions, in: Proceedings of the 6th. ACM SIGACT-SIGPLAN symposium on Principles of Programming Languages, POPL '79, ACM Press, New York, NY, 1979, pp. 226–236. doi:10.1145/567752.567774.
- [9] S. W. Smith, An experiment in bibliographic mark-up: Parsing metadata for xml export, in: R. N. Smythe, A. Noble (Eds.), Proceedings of the 3rd. annual workshop on Librarians and Computers, volume 3 of LAC '10, Paparazzi Press, Milan Italy, 2010, pp. 422–431. doi:99.9999/woot07-S422.
- [10] D. Harel, LOGICS of Programs: AXIOMATICS and DESCRIPTIVE POWER, MIT Research Lab Technical Report TR-200, Massachusetts Institute of Technology, Cambridge, MA, 1978.
- [11] K. L. Clarkson, Algorithms for Closest-Point Problems (Computational Geometry), Ph.D. thesis, Stanford University, Palo Alto, CA, 1985. UMI Order Number: AAT 8506171.
- [12] D. A. Anisi, Optimal Motion Control of a Ground Vehicle, Master's thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2003.
- [13] H. Thornburg, Introduction to bayesian statistics, 2001. URL: <http://ccrma.stanford.edu/jos/bayes/bayes.html>.
- [14] Poker-Edge.Com, Stats and analysis, 2006. URL: <http://www.pkredge.com/statsYYFWWQ.php>.
- [15] B. Obama, A more perfect union, Video, 2008. URL: <http://video.google.com/videoplay?docid=6528042696351994555>.
- [16] D. Novak, Solder man, in: ACM SIGGRAPH 2003 Video Review on Animation theater Program: Part I - Vol. 145 (July 27–27, 2003), ACM Press, New York, NY, 2003, p. 4. URL: <http://video.google.com/videoplay?docid=6528042696351994555>. doi:99.9999/woot07-S422.
- [17] R. Yang, F. R. Yu, P. Si, Z. Yang, Y. Zhang, Integrated blockchain and edge computing systems: a survey, some research issues and challenges, IEEE Commun. Surv. Tutorials, 21 (2019), 1508–1532. <http://doi.org/10.1109/COMST.2019.2894727>
- [18] T. Ma, H. Wang, L. Zhang, Y. Tian, N. Al-Nabhan, Graph classification based on structural features of significant nodes and spatial convolutional neural networks, Neurocomputing, 423 (2021), 639–650. <https://doi.org/10.1016/j.neucom.2020.10.060>
- [19] Y. Tian, B. Song, M. Murad, N. Al-Nabhan, Trustworthy collaborative trajectory scheme for continuous LBS, Int. J. Sens. Networks, 38 (2022), 58–69. <http://doi.org/10.1504/IJSNET.2022.120275>
- [20] L. Fu, Z. Li, Q. Ye, H. Yin, Q. Liu, X. Chen, et al., Learning robust discriminant subspace based on joint L2,p- and L2,s-norm distance metrics, IEEE Trans. Neural Networks Learn. Syst., 33 (2022), 130–144. <https://doi.org/10.1109/TNNLS.2020.3027588>
- [21] Q. Ye, P. Huang, Z. Zhang, Y. Zheng, L. Fu, W. Yang, Multiview learning with robust doublesided twin SVM, IEEE Trans. Cybern., 2021 (2021). <https://doi.org/10.1109/TCYB.2021.3088519>
- [22] Q. Ye, Z. Li, L. Fu, Z. Zhang, W. Yang, G. Yang, Nonpeaked discriminant analysis for data representation, IEEE Trans. Neural Networks Learn. Syst., 30 (2019), 3818–3832. <https://doi.org/10.1109/TNNLS.2019.2944869>

- [23] Z. Tong, F. Ye, M. Yan, H. Liu, S. Basodi, A survey on algorithms for intelligent computing and smart city applications, *Big Data Mining Anal.*, 4 (2021), 155–172. <https://doi.org/10.26599/BDMA.2020.9020029>
- [24] J. H. Anajemba, T. Yue, C. Iwendi, M. Alenezi, M. Mittal, Optimal cooperative offloading scheme for energy efficient multi-access edge computation, *IEEE Access*, 8 (2020), 53931–53941. <https://doi.org/10.1109/ACCESS.2020.2980196>
- [25] S. Guo, X. Hu, S. Guo, X. Qiu, F. Qi, Blockchain meets edge computing: a distributed and trusted authentication system, *IEEE Trans. Ind. Inf.*, 16 (2020), 1972–1983. <https://doi.org/10.1109/TII.2019.2938001>
- [26] P. Zhang, C. Tian, T. Shang, L. Liu, L. Li, W. Wang, et al., Dynamic access control technology based on zero-trust light verification network model, in *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*, (2021), 712–715. <https://doi.org/10.1109/CISCE52179.2021.9445896>

The modified algorithm tree method in the geological data classification problem

Igor Povkhan ¹, Oksana Mulesa ¹, Olena Melnyk ¹, Vasyl Morokhovych ¹

¹ Uzhhorod National University, Zankoveckoy str., 89B, Uzhhorod, 88000, Ukraine

Abstract

This study presents the development of an advanced algorithmic tree synthesis method predicated on a set configuration of initial data for the task of geological data recognition. The devised classification tree algorithm of the second type demonstrates precise classification of the complete training dataset, adhering to the established classification schema. It boasts high interpretability, a straightforward structure, and incorporates autonomous algorithms for classification and scheme recognition as vertices within a graphical framework. The refined construction methodology for the tree algorithm facilitates handling substantial volumes of discrete data across diverse categories, ensuring remarkable accuracy of the classification schema. Moreover, it judiciously utilises hardware resources during the creation of the definitive classification schema and supports the development of models with specified accuracy levels. The paper advocates a novel synthesis approach for recognition algorithms, drawing on a repository of extant algorithms and theoretical recognition methods. Employing the proposed second type tree algorithm, a suite of models has been constructed that adeptly classifies extensive arrays of geological data. The constructed models of classification trees have verified the absence of errors in both training and testing datasets, substantiating the efficacy of the second type tree method algorithm.

Keywords

Algorithmic tree, classifier, pattern recognition, feature, initial sample.

1. Introduction

Classification and image recognition represent critical problem domains within the sphere of artificial intelligence, notable for their extensive diversity, varying degrees of structural complexity, and significant applicability across numerous sectors of human economic and social endeavours. In disciplines such as geology, where the challenges of classification are tackled through sophisticated information systems, the importance and intensity of research in this area are well-documented [1-10]. These classification challenges demand the development and decomposition of mathematical models tailored to the specific systems under study. Presently, the field of artificial intelligence lacks a universally applicable approach capable of addressing the full spectrum of these complex problems. However, several broadly applicable theories and methodologies have emerged, with neural networks being particularly prominent due to their versatility in addressing a wide array of classification challenges [11-14]. In practical scenarios, specifically configured artificial neural networks often outperform traditional algorithms and established decision tree models, such as gradient boosting methods, especially in tasks involving unstructured data, discrete image sets, or textual content. Conversely, when dealing with structured datasets comprising large volumes of massive discrete data, which exhibit diverse feature spaces, decision tree-based methods and algorithms exhibit distinct advantages [15]. Generally, classification tree methodologies facilitate effective data processing across various magnitudes, presenting the input information in its inherent form. Numerous contemporary strategies and concepts are focused on developing recognition systems (RS) and classifications using logical/algorithmic classification tree models (LCT/ACT structures). The growing interest in tree-like graph-schematic representations of classifiers is driven by their numerous

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ igor.povkhan@uzhnu.edu.ua (I. Povkhan); oksana.mulesa@uzhnu.edu.ua (O. Mulesa); olena.melnyk@uzhnu.edu.ua (O. Melnyk); morv77@ukr.net (V. Morokhovych)

ORCID: 0000-0002-1681-3466 (I. Povkhan); 0000-0002-6117-5846 (O. Mulesa); 0000-0001-7340-8451 (O. Melnyk); 0000-0002-4939-6566 (V. Morokhovych)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

advantageous properties [16]. One promising area of application for the classification tree model, specifically within the realm of algorithmic trees, is in the classification of geological informations [22].

2. Formal problem statement

Let be $H_1, H_2, \dots, H_k$ the system of classes (images) defined on the set G consisting of objects $x_i, (i = 1, \dots, m)$. The nature of the division of the set G into the corresponding classes is specified using the following training sample (TS):

$$\left((x_1, f_R(x_1)), (x_2, f_R(x_2)), \dots, (x_m, f_R(x_m)) \right). \quad (1)$$

Let us note that here $x_h \in G, f_R(x) \in \{0, 1, \dots, k-1\}, (h = 1, 2, \dots, m), k$ – the number of TS classes, m – the total number of TS objects, and $f_R(x)$ is some finitely significant function that determines the division of the set G into corresponding images. The ratio $f_R(x_h) = l, (l = 0, 1, \dots, k-1)$ means that $x_h \in H_l$. We note that each TS of the form (1) can be according (with the help of some algorithm or representation method) to a wholly defined LCT, which matches $f_R(x_h)$ the objects of TS (1) with the value of the function $x_h, (h = 1, \dots, m)$, which specifies the partition R on the set G . Therefore, the task will be to build a structure of the classification tree (LCT/ACT), the structure of which would be optimal $f_R(x_j) \rightarrow opt$ with the initial data of TS.

3. Literature review

The current study delves into the theory of fixed-type decision trees, focusing on algorithm trees and the classification of discrete objects [14, 23, 25]. Notably, research [20] underscores that the classification rules and decision schemes, derived from any branching feature selection method or algorithm, manifest a tree-like logical structure. A typical decision tree classifier comprises an organized sequence of nodes, features, and attributes structured into layers or levels, each established during a specific phase of the classification tree synthesis [15]. A significant challenge identified in [18] is the effective construction of recognition tree structures, which can take the form of tree-like structures or algorithm graphs (ACT structures). Consequently, decision tree methodologies facilitate the creation of innovative classifiers based on a modular principle, utilizing well-known recognition algorithms [19-21]. The study [14] explores fundamental issues related to the generation of decision tree structures, particularly when features are low-informative, including their sets and combinations. Within the sphere of intelligent data analysis, the invariant capacity of LCT/ACT structures to execute one-dimensional branching allows for the analysis of the influence, importance, and quality of individual variables. This capability is essential for managing different types of variables as predicate sets. The persistent challenge with decision tree methods and structures is evaluating the quality and efficiency of the branches (generalized features) that serve as autonomous classification algorithms [15]. Logical decision tree classification methods are prevalently employed in intelligent data analysis, aiming to synthesize operational models that predict the value of a target variable based on an initial set of data formatted as a structured training sample [19]. From an applied perspective, numerous methods and algorithms grounded in the decision tree concept are utilized for classification tasks; however, C4.5/C5.0 and CART have emerged as particularly popular. The C4.5/C5.0 methods employ a theoretical-informational criterion for node or vertex selection, whereas the CART algorithm relies on the Gini index, which assesses the relative distances between class distributions within the metric of the training sample [20, 21]. The set of methods and algorithms for branching feature selection (ACT structures) is based on optimally approximating the initial training set using a ranked series of classification algorithms [22]. A key issue within LCT/ACT methods, as discussed in [23], involves choosing an effective branching criterion—that is, selecting nodes, attributes, and features of discrete objects for LCT schemes and algorithms for ACT. These foundational issues are thoroughly examined in another paper [24], which addresses the qualitative evaluation and informativeness of individual discrete features, their sets, and fixed combinations, ultimately supporting the efficient implementation

of a branching mechanism within the logical/algorithmic tree structure. Concerns regarding the convergence of the classification tree construction process, including the selection of stopping criteria for the synthesis of logical and algorithmic trees, remain significant [25]. The concept of classification trees accommodates the use of not only individual attributes and object features but also their combinations and sets as features, attributes, and nodes of the recognition tree structure. By adopting independent individual recognition algorithms (evaluated using training data) instead of object attributes as branches, a novel ACT structure is realized [21-24]. This research specifically targets the exploration of fixed-type ACT structures within the practical domain.

4. The general second type trees method algorithm

Let the initial TS of the general form (1) be given as a sequence of training pairs of known classification (power m) and some system (set) of independent and autonomous recognition (classification) algorithms for the initial TS $\alpha_1(x), \alpha_2(x), \dots, \alpha_n(x)$. Next, it is necessary to enter the following sets, which represent the breakdown of the data of TS by the corresponding classification algorithms a_i :

$$G_{a_1, \dots, a_i} = \{x \in G / \alpha_i(x) = 1\}, (i = 1, \dots, n). \quad (2)$$

Note that to simplify explanations, each autonomous classification algorithm $\alpha_i(x)$ generates output values only within the binary set $\{0,1\}$, particularly $\alpha_i(x) = 1$ in the case of successful object classification x and $\alpha_i(x) = 0$ in the opposite case.

Note that the system of sets $G_{a_1, \dots, a_i}$ will represent a complete step-by-step division of the set G (with an increase in the size of i the involved classification algorithms), which is implemented by independent algorithms $\alpha_1, \alpha_2, \dots, \alpha_n$. Note that depending on the initial selection of a set of classification algorithms, $\alpha_1, \alpha_2, \dots, \alpha_n$ some of the sets $G_{a_1, \dots, a_i}$ may be empty (in case one or more algorithms are not suitable for approximating the current TS) [21].

At the next stage, we denote by the value $S_{a_1, \dots, a_n}$ the number of occurrences of those training pairs $(x_s, f_R(x_s))$, $(1 \leq s \leq m)$ which satisfy the basic condition of belonging, in the initial TS $x_s \in G_{a_1, \dots, a_i}$.

Accordingly, by the value $S_{a_1, \dots, a_i}^j$, $(j = 0, 1, \dots, k-1)$ we denote the number of occurrences in the TS of those pairs $(x_s, f_R(x_s))$ $(s = 1, 2, \dots, m)$, which satisfy the conditions $x_i \in G_{a_1, \dots, a_n}$ and $f_R(x_s) = j$.

So, taking into account the above, what was said and by analogy with the methods of selection of sets of elementary features, the following values can be introduced, which should be considered as a certain criterion of branching in the structure of the ACT:

$$\delta_{a_1, \dots, a_i} = \frac{S_{a_1, \dots, a_i}}{m}, \psi_{a_1, \dots, a_i}^j = \frac{S_{a_1, \dots, a_i}^j}{S_{a_1, \dots, a_i}}, \rho_{a_1, \dots, a_i} = \max_j \psi_{a_1, \dots, a_i}^j. \quad (3)$$

Note that if the object $x_s \notin G_{a_1, \dots, a_i}$ is for all $s = 1, \dots, m$, then it is clear that $\delta_{a_1, \dots, a_i} = 0$ and $\psi_{a_1, \dots, a_i}^j = 0$ for $j = 0, 1, \dots, k-1$.

Particularly the quantity $\delta_{a_1, \dots, a_i}$ characterizes the frequency of occurrences of members of the sequence $x_1, x_2, \dots, x_m$ (discrete objects) in the set $G_{a_1, \dots, a_i}$, and accordingly, the quantity $\psi_{a_1, \dots, a_i}^j$ characterizes the frequency of belonging to some object x of the image (class) H_j , provided that $x \in G_{a_1, \dots, a_i}$. It should be noted that the given condition is equivalent to the condition that in the sequence of algorithms $a_1, \dots, a_i$ there is such an algorithm a_y that $a_y(x) = 1$. Then the value $\delta_{a_1, \dots, a_i}$ characterizes the information efficiency of recognizing the belonging of some object x to one of the classes $H_0, H_1, \dots, H_{k-1}$ provided that $x \in G_{a_1, \dots, a_i}$.

At the next stage, a fundamental question arises again regarding the object's belonging to x classes $H_0, H_1, \dots, H_{k-1}$ (the question of forming a classification rule). It is clear that the object should be assigned x to the class H_j for which a simple relation is fulfilled:

$$\rho_{a_1, \dots, a_i} = \psi_{a_1, \dots, a_i}^j. \quad (4)$$

Note that here $\{0 \leq j \leq k-1\}$, and relation (4) represents a certain classification rule, and it is clear that the greater the value of the value of $\rho_{a_1, \dots, a_i}$, the higher the effectiveness of the rule.

Since the only information that represents the partitioning of images $H_0, H_1, \dots, H_{k-1}$ is the initial TS, then the class H_j is understood as the set of all training pairs $(x_s, f_R(x_s))$ of TS that satisfy the ratio $f_R(x_i) = j$, that is, the condition of belonging.

It is clear that an algorithmic tree is not the only possible construction (structure) classification algorithm that can be organized in the form of a tree-like recognition model (several types of such structures can be proposed). Next, we will propose one scheme for organizing a set of classification and recognition algorithms $(\alpha_1, \alpha_2, \dots, \alpha_m)$ in the form of an ACT model, which we will call an algorithmic classification tree of the second type.

Let us note that a set of autonomous classification and recognition algorithms $(\alpha_1, \alpha_2, \dots, \alpha_m)$ can act as a set of primary features (attributes) for an arbitrary discrete object of x_i some initial TS of the general form (1). Moreover, with regard to a fixed discrete object x_i of the initial TS, information about the appearance of the generalized feature (GF), which is built by the current classification algorithm, and information about the general possibility of recognizing this discrete object (presence of failure, incorrect classification, impossibility) will be necessary for this ACT scheme GF to describe this object, etc.).

Therefore, let each training pair $(x_i, f_R(x_i))$ of the TS correspond to its training pair of the following form:

$$(x_i(\varphi(\alpha_1), \varphi(\alpha_2), \dots, \varphi(\alpha_m)), f_R(x_i)), \varphi(\alpha_j) \in \{0,1\} \text{ where.} \quad (5)$$

Moreover $\varphi(\alpha_j) = 1$, if this discrete object is approximated by some GF f_l , which is built by the α_j set classification algorithm $(\alpha_1, \alpha_2, \dots, \alpha_m)$ at the corresponding stage of ACT generation. Similarly $\varphi(\alpha_j) = 0$, if for a given discrete object the algorithm α_j did not build a suitable GF (which would ensure its approximation, classification), this situation also includes failures and classification errors (errors of the first and second kind).

By the algorithmic tree of the second type, we will understand some tree-like construction, the general view of which is presented in (Fig. 1), at the vertices of which there are appropriate labels (classification and recognition algorithms, α_j as well as sets of GFs that they generate at a specific step of the ACT construction procedure). Note that the logical tree of this construction belongs to the class of regular logical trees of full complexity (this logical tree will be equivalent to a logical function of four arguments, the arguments of which take values from the set $\{0,1\}$).

The following basic ACT scheme for synthesizing a tree of algorithms of the second type based on a branched selection of generalized features allows us to build ACT structures of arbitrary complexity and efficiency (Fig. 1).

Stage of initial selection and evaluation of independent classification algorithms. At the initial stage, it is necessary to select and evaluate the basic set (fixed set) of classification and recognition algorithms $(\alpha_1, \alpha_2, \dots, \alpha_m)$ from the initial algorithm library. Note that this procedure is performed based on the selected (fixed) performance criterion, followed by ranking – interactive or randomly. The performance criterion may vary depending on the type of act structure that is being built and cannot be changed during the classification tree synthesis process. The set of autonomous algorithms $(\alpha_1, \alpha_2, \dots, \alpha_m)$, as well as their total number in the set, are selected depending on the applied aspects of the problem and can be selected even on the basis of a complete search of the algorithm library (of course, with significant losses of hardware resources and processor time). At the initial stage of synthesis of the second type of ACT model, by selecting (ranking) a set of classification algorithms and their total number, the final structural complexity of the algorithm tree can be controlled.

Stage of synthesis of the algorithm tree structure and generalized features. At the next stage, the central task is to build a complete regular classification tree (fixed LCT structure), where the corresponding tiers of the structure contain the selected classification algorithms $(\alpha_1, \alpha_2, \dots, \alpha_m)$, fixed at the first stage of constructing classifier sets.

A special feature of the algorithm tree of the second type is that in the constructed classification tree structure (LCT structure), each vertex has two transitions to the next level, denoted by a value from the binary set $\{0,1\}$. This is why the structure of the algorithm tree is represented using a regular LCT construct. Based on this, all attributes (labels) of the same type (classification algorithms and generated generalized features) are located at each of the levels of this structure. In such a regular classification tree structure, nodes are independent algorithms (classifiers) $(\alpha_1, \alpha_2, \dots, \alpha_m)$. Generalized feature sets (GFs) of f_j are also generated during the

synthesis step of the algorithms tree structure. Therefore, we can conclude that the algorithm tree generates a tree of generalized features.

The idea of the second stage of synthesis of the algorithm tree structure (ACT type II model) is the procedure for synthesizing a set of generalized features f_j (vertices of the generalized features tree) based on pre-selected sets of independent classification and recognition algorithms α_i . Note that the total number of GFs f_j generated by the corresponding classification algorithm depends on the initial parameters of the ACT model and synthesis parameters, the specifics of the application problem, and the resource constraints of the classification tree synthesis system.

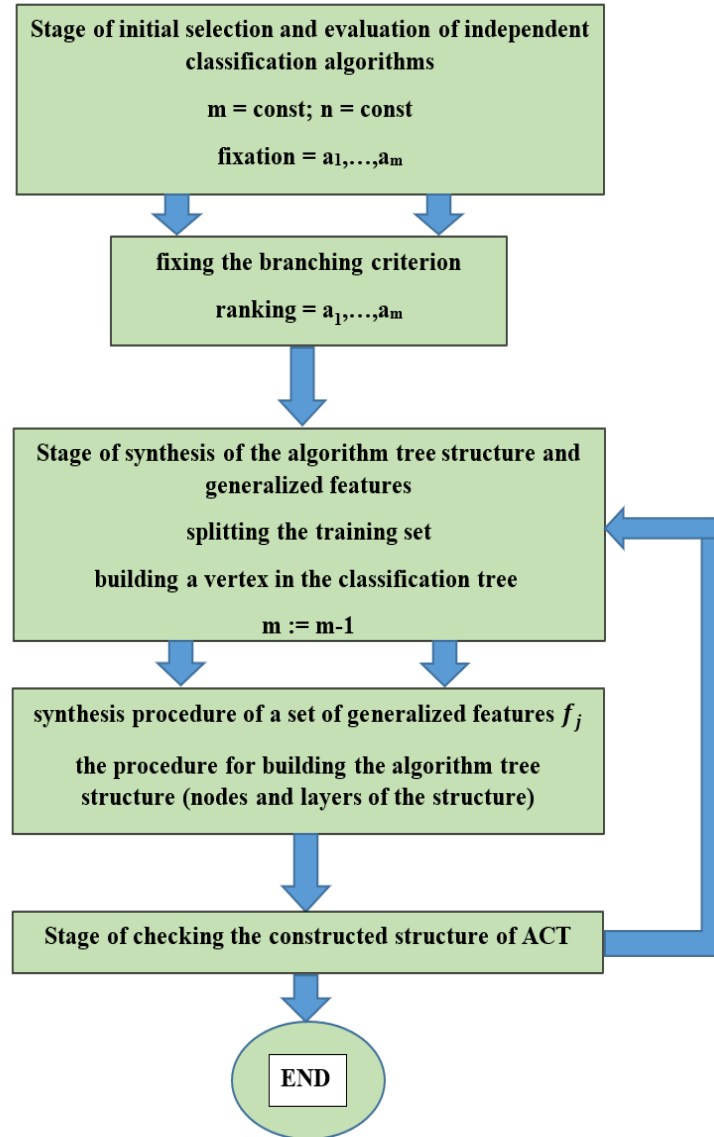


Figure 1: The general block - diagram of the second type tree method

At the end of the second stage, after the formation of a set of synthesized generalized features f_j for a given application problem is completed, they are located in the corresponding nodes, tiers of the tree of algorithms of the second type (the structure of the tree of generalized features is constructed).

Stage of checking the constructed structure of ACT. At the final stage of synthesizing the second type of algorithm tree, you need to check the constructed ACT model. For each element (object) of the test sample, the corresponding values of $\varphi(\alpha_j)$ are calculated. This value is calculated based on a set of previously constructed generalized features - for each node of the corresponding tree level. The constructed generalized features define the corresponding route

(bounded classifier) in the structure of the tree of algorithms of the second type. For such a GFs structure, each of the nodes in the algorithm tree, in the event of a possible approximation of an object of unknown classification, increases the corresponding counter of the class belonging to it and leaves it unchanged in the event of a classification error or failure. This procedure allows you to make a final assessment of the effectiveness of the constructed tree of algorithms of the second type.

5. Experiments and results

The experimental validation of the proposed second type algorithm tree construction scheme underscores its capability to tune the complexity and accuracy of the resulting classification tree model. The model comprises various autonomous classification algorithms which, during the modeling process, evolve into a hierarchical structure of generalized features. The selection of an optimal model from the array of constructed Algorithmic Classification Trees (ACTs) for a specific task hinges on evaluating multiple parameters and the effectiveness of the model, which is typically assessed through techniques such as cross-validation against the training set (TS) data. An essential stage in this process involves identifying the most critical parameters of the model, such as the feature space size, the number of vertices, transitions, and algorithms. This step is crucial for estimating the ACT's error relative to the input data set, facilitating comparison, and aiding in the selection of a specific ACT model from the pre-defined ensemble.

Quality criteria of the constructed algorithm trees are paramount and depend on several factors including model error, the robustness of the initial TS data set, the size of the testing sample, and the dimensional characteristics of the problem (e.g., the number of model parameters). At the optimization stage of the constructed ACT model, priority is given to minimizing errors across the training and test datasets for each class defined by the initial conditions of the current applied problem.

A significant ongoing challenge is reducing the complexity and structural pruning of the ACT model. This reduction pertains to the overall count of functions (classifiers) and algorithms within the ACT framework, the total number of vertices (generalized features), and the number of transitions within the structure, as well as optimizing total memory usage and processing time of the information system. Consequently, the defining measure of the quality and efficiency of a constructed model, whether ACT or Logical Classification Trees (LCT), is determined by an overall integral quality indicator:

$$Q_{Main} = \frac{Fr_{All}}{V_{All} \cdot \sum_i p_i} \cdot e^{-\frac{Er_{All}}{M_{All}}}. \quad (6)$$

Note that in formula (6), the set of parameters p_i represents the most important characteristics of the constructed classification tree that is evaluated:

- 1) Er_{All} – the total number of errors of the ACT model on the data arrays of the initial test and training samples;
- 2) M_{All} – the total capacity (volume) of data arrays of training and test samples;
- 3) Fr_{All} – the number of vertices of the obtained ACT model with the resulting values f_R (recognition functions, i.e. leaves of the classification tree);
- 4) V_{All} – represents the total number of all types of vertices in the structure of the ACT model;
- 5) O_{Uz} – the total number of generalized features used in the classification tree model;
- 6) P_{All} – the total number of transitions between vertices in the structure of the constructed classification tree model;
- 7) N_{Alg} – the total number of different autonomous classification algorithms a_i used in the classification tree model.

Note that this integral indicator of the quality of the ACT model will take values from zero to one. The smaller it is, the worse the quality of the constructed classification tree will be, and the larger the indicator, the better the resulting model will be.

The Orion software complex was developed at the Uzhhorod National University based on classification tree methods to generate autonomous recognition systems. The algorithmic library of the system includes 18 recognition algorithms, among which tree schemes of algorithms of three types are implemented.

The primary task on which the effectiveness of algorithm tree methods was tested was the task of recognizing geological data – the task of separating oil-bearing and water-bearing strata. The initial parameters of this applied problem of geological data classification are presented in (Table 1).

Information about objects of two classes is presented in the TS. At the examination stage, the constructed classification system should effectively recognize objects of unknown classification relative to these two classes. Before starting work, the training sample was automatically checked for correctness - finding and removing errors of the first kind. The system implements a retraining and error correction scheme in the classification tree (REC algorithm).

The training sample of the presented problem consisted of 1342 objects, of which 761 were oil-bearing objects. The effectiveness of the constructed ACT model was evaluated on a test sample of 267 objects. The data from training and test samples were obtained based on geological exploration in the territory of the Transcarpathian region in the period from 2001 to 2019. A fragment of the main results of the above experiments, constructed models of LCT/ACT of various types, are presented in (Table 2).

Table 1
Initial parameters of the classification problem

Description of classes H_i tasks	The dimension of the feature space N	The power of data array of the primary $IS - M$	The total number of classes by data splitting $IS - l$	Relation of objects of different classes $IS - H_i / M$
Oil-bearing layers (H_1)	(12/10)	1342	2	761 / 1342
Aquifers (H_2)	(12/10)	1342	2	581 / 1342

(Table 3) presents information on the classification models' generation time, the total number of vertices, and elementary and generalized features on the basic hardware configuration Intel i7-12700H. All constructed schemes of classification trees (structures of LCT/ACT) provided the necessary level of accuracy given by the task condition, speed, and consumption of the system's working memory.

Table 2
Comparison table of built ACT/LCT models for classification of geological data

Classification tree model No	Method of synthesis of classification tree structure	Integral feature of model quality Q_{Main}	The overall indicator of the structural complexity of the classification tree model S_{Main}	The number of errors and failures to classify the LCT/ACT model on the data set Er_{All}
No. 1	The method of full LCT based on the selection of elementary traits (extensive selection of features)	0.004786	121	7

The total number of elementary / generalized features in the structure of the classification tree O_{el}/O_{Uz}	56 (el.)	72 (el.)	40 (el.)	17 (g.)	41 (g.)	30 (g.)	18 (g.)	47 (g.)	21 (g.)	35 (g.)

Therefore, the algorithmic tree classification method proposed in the paper (second-type ACT methods) was compared with the complete LCT method and the limited method of selection of elementary features and showed a generally acceptable result.

6. Conclusion

The developed models of classification trees (ACT/LCT structures) have successfully met the requirements for quality and speed in geological data classification schemes while maintaining a compact structure (parameter S_{Main}). The sets of independent classification algorithms selected for generating GF groups also demonstrated their effectiveness within the scope of this applied problem. Notably, the models of ACT employing basic geometric classifiers were found to be the most effective upon critical evaluation. Furthermore, the composite ACT structures resulted in a relatively low number of classification errors in both the training and testing datasets. The full model of the second type ACT, based on geometric classifiers, showed promising results ($Q_{Main}=0,003033$), largely due to the inclusion of a universal algorithm of hyperspheres in the scheme. In contrast, the structure of the first type ACT exhibited superior quality ($Q_{Main}=0,005287$) compared to second type algorithm trees. This superiority is attributed to the more complex construction of the model ($S_{Main}=52$), which, consequently, required longer generation times. However, it is essential to consider the limitations of the selected geometric classifiers, which may not always provide effective approximation of the TS data. A notable drawback of the ACT models presented, identified during this task, is the relatively high time consumption during the synthesis stage of the classification tree models, especially when compared to the LCT structures. The time difference in constructing the first type of ACT models, which includes a step-by-step assessment of feature informativeness, was nearly 34% greater than that of LCT.

The scientific novelty lies in the fact that for the first time a modified method for constructing algorithm trees based on evaluating and ranking a set of autonomous recognition algorithms for generating a classification tree structure (ACT model) has been proposed.

The practical implications of these findings are significant. The proposed method for constructing ACT models (of the second type) enables the creation of economical and efficient classification models with specified accuracy. This method has been integrated into the algorithm library of the "ORION" system, addressing various applied classification challenges and demonstrating a high degree of versatility across a range of applications. The efficacy of the classification tree models and the associated software have been confirmed through practical applications. Looking ahead, future research could focus on the further development of ACT methods, including the introduction of new types and schemes of classification trees. Additionally, optimizing the software implementations of the proposed ACT method and its practical validation on a variety of precise classification and recognition tasks could provide valuable insights and enhancements to the field.

Acknowledgements

The presented work was carried out within the framework of projects: Expanding Opportunities of High Technologies for Higher Education Institutions (Skills2Scale) - HEI Initiative; Innovative teaching methods to support partnership relations - "InovEduc" - grant project No. CBC01008 of the Norwegian State Fund with the solidarity budget of the Slovak Republic within the framework of the SK08 cross-border cooperation program; Research work "Modeling and prediction of emergency situations in the Carpathian region and the countries of Central Eastern Europe", state registration number of the work - 0106V00285, work category - fundamental research, 01 Fundamental research on the most important problems of natural, social and humanitarian sciences.

References

- [1] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer, Berlin, 2008.
- [2] J.R. Quinlan, *Induction of Decision Trees*, *Machine Learning* 1 (1986) 81-10.
- [3] L.L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and regression trees*, Chapman and Hall/CRC, Boca Raton, 1984.
- [4] A. Kintonova, M. Mussaif, G. Gabdreshov, Improvement of iris recognition technology for biometric identification of a person, *Eastern-European Journal of Enterprise Technologies* 6(2(120)) (2023) 60–69. doi: 10.15587/1729-4061.2022.269948.
- [5] Y.V. Bodyanskiy, A.Y. Shafronenko, I.P. Pliss, Credibilistic fuzzy clustering based on evolutionary method of crazy cats, *System Research and Information Technologies* 3 (2021) 110–119. doi: 10.20535/SRIT.2308-8893.2021.3.09.
- [6] M. Miyakawa, Criteria for selecting a variable in the construction of efficient decision trees, *IEEE Transactions on Computers* 38(1) (1989) 130-141.
- [7] H. Koskimaki, I. Juutilainen, P. Laurinen, J. Roning, Two-level clustering approach to training data instance selection: a case study for the steel industry, in: *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008*, IEEE, Los Alamitos, 2008, pp. 3044-3049. doi: 10.1109/ijcnn.2008.4634228.
- [8] S.F. Jaman, M. Anshari, Facebook as marketing tools for organizations: Knowledge management analysis, in: S.F. Jaman, M. Anshari, *Dynamic perspectives on globalization and sustainable business in Asia*, IGI Global, Hershey, 2019, pp. 92-105. doi: 10.4018/978-1-5225-7095-0.ch007.
- [9] V.E. Strilets, S.I. Shmatkov, M.L. Ugryumov et al, *Methods of machine learning in the problems of system analysis and decision making*, Karazin Kharkiv National University, Kharkiv, 2020.
- [10] R.L. De Mántaras, A distance-based attribute selection measure for decision tree induction, *Machine learning* 6(1) (1991) 81-92.
- [11] K. Karimi, H. Hamilton, Generation and Interpretation of Temporal Decision Rules, *International Journal of Computer Information Systems and Industrial Management Applications* 3 (2011) 314-323.
- [12] B. Kamiński, M. Jakubczyk, P. Szufel, A framework for sensitivity analysis of decision trees, *Central European Journal of Operations Research* 26(1) (2017) 135-159.
- [13] H. Deng, G. Runger, E. Tuv, Bias of importance measures for multi-valued attributes and solutions, in: *Proceedings of the 21st International Conference on Artificial Neural Networks*, volume 2 of ICANN 2011, Springer-Verlag, Berlin, 2011, pp. 293-300. doi: 10.1007/978-3-642-21738-8_38.
- [14] S.A. Subbotin, Construction of decision trees for the case of low-information features, *Radio Electronics, Computer Science, Control* 1 (2019) 121-130. doi: 10.15588/1607-3274-2019-1-12.
- [15] A. Shyshatskyi, Complex Methods of Processing Different Data in Intellectual Systems for Decision Support System, *International Journal of Advanced Trends in Computer Science and Engineering* 9(4) (2020) 5583–5590. doi: 10.30534/ijatcse/2020/206942020.

- [16] A. Painsky, S. Rosset, Cross-validated variable selection in tree-based methods improves predictive performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(11) (2017) 2142-2153. doi:10.1109/tpami.2016.2636831.
- [17] D. Imamovic, E. Babovic, N. Bijedic, Prediction of mortality in patients with cardiovascular disease using data mining methods, in: *Proceedings of the 19th International Symposium INFOTEH-JAHORINA, INFOTEH 2020*, IEEE, Los Alamitos, 2020, pp. 1-4. doi:10.1109/INFOTEH48170.2020.9066297.
- [18] S.B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, *Informatica*, 31 (2007) 249-268.
- [19] Y.I. Zhuravlev, V.V. Nikiforov, Recognition algorithms based on the calculation of estimates, *Cybernetics* 3 (1971) 1-11.
- [20] I. Povkhan, O. Mulesa, O. Melnyk, Y. Bilak, V. Polishchuk, The Problem of Convergence of Classifiers Construction Procedure in the Schemes of Logical and Algorithmic Classification Trees, in: *Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems, CMIS-2022*, Volume 3137 of CEUR Workshop Proceedings, CEUR-WS, Zaporizhzhia, Ukraine, 2022, pp. 1-13.
- [21] I. Povkhan, A constrained method of constructing the logic classification trees on the basis of elementary attribute selection, in: *Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems, CMIS-2020*, CEUR Workshop Proceedings, Volume 2608, 2020, pp. 843-857.
- [22] I. Povkhan, M. Lupei, The algorithmic classification trees, in: *Proceedings of the IEEE Third International Conference on Data Stream Mining & Processing, DSMP 2020*, IEEE, Los Alamitos, 2020, pp. 37-44.
- [23] I. Povkhan, M. Lupei, M. Kliap, V. Laver, The issue of efficient generation of generalized features in algorithmic classification tree methods, in: *Proceedings of the International Conference on Data Stream Mining and Processing, DSMP 2020*, IEEE, Los Alamitos, 2020, pp. 98-113.
- [24] I. Povkhan, Classification models of flood-related events based on algorithmic trees, *Eastern-European Journal of Enterprise Technologies*, 6(4) (2020) 58-68. doi: 10.15587/1729-4061.2020.219525.
- [25] J. Rabcan, V. Levashenko, E. Zaitseva, M. Kvassay, S. Subbotin, Application of Fuzzy Decision Tree for Signal Classification, *IEEE Transactions on Industrial Informatics*, 15(10) (2019) 5425-5434. doi: 10.1109/tii.2019.2904845.

A Dynamic Blurring Approach with EfficientNet and LSTM to Enhance Privacy in Video-Based Elderly Fall Detection

Ivan Ursul¹, Junaid Hussain Muzamal²

¹ Ivan Franko National University of Lviv, Lviv, Universitetska, 1, 79090, Ukraine

² Fast – National University of Computer and Emerging Sciences, Lahore, Pakistan

Abstract

This research paper introduces a novel approach to address privacy concerns in video-based elderly fall detection systems without compromising such technologies' efficacy and real-time response. The methodology integrates EfficientNetB0 for robust feature extraction from video sequences and Long Short-Term Memory networks for accurate fall classification. Despite achieving exemplary performance metrics, including 100% scores in accuracy, Area Under the Curve, recall, and Precision, the pervasive issue of privacy infringement in video surveillance remains a significant challenge. To tackle this, we propose a dynamic blurring technique that selectively obscures identifiable features within video frames, such as faces and distinguishing clothing, thus maintaining individual anonymity. This method ensures that the privacy of the monitored individuals is preserved while retaining the essential details necessary for the fall detection algorithm to function effectively. This paper details this privacy-preserving technique and demonstrates its feasibility without detracting from the system's performance. Our findings indicate that integrating dynamic blurring into the fall detection pipeline offers a promising solution to the privacy concerns associated with video-based monitoring systems. It protects sensitive personal information while providing high care and safety. This research contributes to the broader discourse on ethical technology use in healthcare. Moreover, it emphasizes the importance of balancing advanced monitoring capabilities with the imperative of privacy preservation.

Keywords

Elderly Fall Detection, Privacy Preservation, Video Surveillance, EfficientNetB0, Long Short-Term Memory (LSTM), Dynamic Blurring, Real-Time Monitoring, Feature Extraction

1. Introduction

The growing demographic of the elderly population has precipitated an increased incidence of falls [1], a leading cause of morbidity and mortality among this group [2]. The recent technological solutions for fall detection have emerged as a critical component in mitigating these risks [3]. Among these, video-based fall detection systems have shown significant promise due to their non-invasiveness and capability for real-time monitoring [4]. However, video surveillance in healthcare, particularly in homes and care facilities, raises significant privacy concerns [5]. This research aims to find a balance between ensuring safety through surveillance and upholding the right to privacy.

There has been a significant relationship between the efficacy of video-based fall detection and the imperative to protect individual privacy [6]. While effective in identifying falls, traditional approaches often overlook the privacy implications of constant video monitoring [7]. Possible solutions include avoiding video data, implementing basic obfuscation techniques [8], compromising effectiveness, or insufficient privacy [9]. Previous research has proposed various methods, including wearable devices [10-13] and environmental sensors [14-16], to circumvent the associated privacy issues. However, these alternatives fall short in accuracy and real-time response capabilities compared to video-based systems.

In response to these challenges, this paper proposes an innovative solution that retains the advantages of video surveillance while addressing privacy concerns. Our approach employs dynamic blurring, selectively obscuring identifiable features within video frames. Thus, individuals are anonymized without compromising the system's ability to detect falls. This

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ Ivanon2@gmail.com (I. Ursul); junaidhocane6728@gmail.com (J. H. Muzamal)

ORCID 0009-0002-9879-8008 (I. Ursul); 0009-0007-1598-8161 (J. H. Muzamal);

© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



method differs from existing solutions by offering a real-time, privacy-preserving mechanism that does not detract from the system's performance. Integrating EfficientNetB0 [17] for feature extraction and Long Short-Term Memory (LSTM) [18] networks for fall events ensures high precision in fall detection.

This research aims to develop a fall detection system that fulfills the need for efficient, real-time monitoring with the imperative of privacy preservation. Our objectives include designing and implementing a dynamic blurring technique within a video-based fall detection framework. Moreover, we also aim to evaluate this system's accuracy and privacy protection performance and demonstrate its applicability in real-world settings. This research can potentially contribute to the development of ethically responsible technological solutions in healthcare, particularly in the context of elderly care. This work seeks to pave the way for broader acceptance by addressing the privacy concerns associated with video-based monitoring. Moreover, deploying such systems enhances the safety and well-being of the elderly population.

2. Literature Review

The exploration of fall detection systems, particularly for the elderly, is an area of research that has seen substantial evolution over time. Dean et al. [19] 2006 implemented the first real-time fall detection system using a triaxial accelerometer. At that time, most traditional techniques centered around simplistic, mechanical solutions and gradually transitioned towards incorporating technology [20]. Among the earliest methods were basic alert systems, which relied on the user to trigger an alert manually in case of a fall [21]. While pioneering for their time, these systems were limited by their dependence on the users to activate the alarm post-fall, which could be compromised due to injury.

Advancements in technology brought in a new wave of methodologies, primarily categorized into sensor-based [14-16], wearable devices [10], [12], and video surveillance systems [22], [23-25], alongside other innovative approaches. Sensor-based systems often utilize accelerometers and gyroscopes to detect sudden movements or orientations indicative of a fall. Wearable devices, such as smartwatches [26], integrate these sensors and offer portability. However, sensor-based and wearable systems face challenges related to user compliance, discomfort, and the potential for false positives due to non-fall-related abrupt movements [27]. In contrast, video surveillance systems offer a less intrusive alternative, capturing a broader context of the individual's environment [28]. This method's appeal lies in its passive nature, requiring no active input or wearables from the monitored individuals. Despite these advantages, video-based systems have challenges [29]. High-quality video processing demands significant computational resources, and managing vast data volumes poses storage and efficiency concerns. Moreover, the critical issue of privacy infringement emerges, given the intrusive nature of continuous video monitoring [9].

Traditional algorithms such as Support Vector Machines (SVMs) [30] and Decision Trees [31] were widely employed in the early stages of machine learning applications for fall detection. These methods primarily relied on handcrafted features extracted from sensor data or basic video analytics, including motion vectors and silhouette shapes. Flow-based methods, particularly optical flow [32], were also prominent, enabling the detection of movement patterns by analyzing the apparent motion of objects, surfaces, and edges. While effective to a certain extent, these approaches faced limitations in handling the high variability and complexity of human falls. They often struggled to distinguish falls from other activities involving rapid movements, leading to high false alarm rates [33], [34]. Additionally, their dependency on manually crafted features restricted their adaptability, as these features might not generalize well across different scenarios.

The recent emergence of deep learning architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) has changed many dynamics [35]. Advanced models such as ResNet [36], LSTM [37], and YOLO especially marked a leap forward in fall detection. CNNs, with their ability to perform automatic feature extraction, have proven particularly adept at analyzing spatial characteristics in video frames [38]. While RNNs and LSTMs excel in capturing temporal dependencies, it is crucial to understand the sequence of movements leading to a fall. YOLO [39], an object detection model, brought further advancements by enabling real-time processing. Despite their successes, the search for enhanced performance

led to exploring hybrid methods that combine multiple deep learning models. For instance, integrating CNNs with LSTMs allows for the effective processing of video data both spatially and temporally, offering a better understanding of fall events [40]. These hybrid approaches [41], alongside innovative methods within deep learning frameworks, promise to address the dynamics of fall detection [42].

Recent advancements aim to address these privacy concerns while maintaining system efficacy. Techniques such as dynamic blurring and real-time anonymization have been explored to obscure identifiable features in video feeds. This can help safeguard individual privacy without significantly compromising detection capabilities. Despite these efforts, there is a gap in the literature concerning developing a system that seamlessly integrates high detection accuracy with robust privacy protection. Our contribution to this field addresses this gap by proposing a novel fall detection system that employs EfficientNetB0 for advanced feature extraction and LSTM networks for accurate temporal classification, complemented by a dynamic blurring mechanism to ensure privacy. This integrated approach promises high performance, as evidenced by optimal accuracy, recall, and precision scores. Moreover, it introduces a viable solution to the privacy concerns that have long shadowed video-based monitoring systems. By achieving this delicate balance, our research paves the way for the broader acceptance of video-based fall detection systems, ensuring the safety of the elderly population.

3. Methodology

This research presents a methodological framework to address the challenge of detecting falls through video surveillance while safeguarding their privacy. The foundation of the proposed approach is mathematical models and techniques to ensure precision, efficiency, and reliability. The proposed method integrates state-of-the-art EfficientNetB0 for spatial feature extraction and LSTM networks for temporal sequence analysis. Additionally, we introduce a dynamic blurring mechanism formulated to preserve privacy by selectively obscuring identifiable features within video frames. Figure 1. provides the overall architecture of the proposed approach.

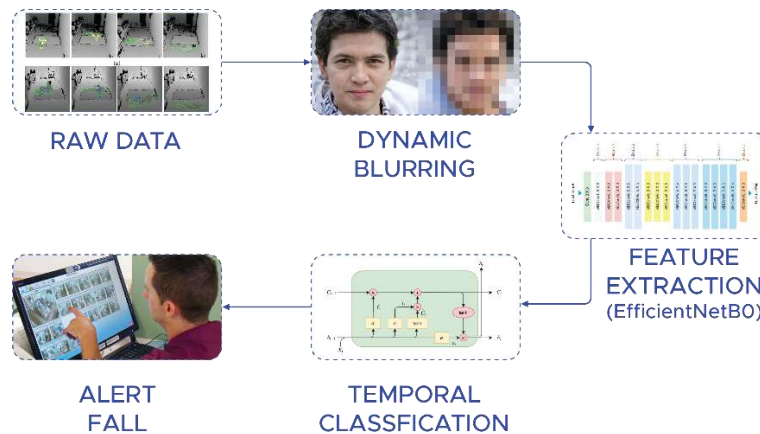


Figure 1. Overall Architecture of the Proposed Methodology

3.1. Dataset and Processing

The dataset employed in this study was sourced from the UR Fall Detection Dataset [43], encompassing 70 sequences, of which 30 are fall events, and 40 represent activities of daily living (ADL). The fall events were captured using two Microsoft Kinect cameras, accompanied by accelerometric data, whereas the ADL events were documented using a single camera (camera 0) alongside accelerometer data. The accelerometric data was acquired through PS Move (60Hz) and x-IMU (256Hz) devices. The dataset is structured such that each sequence comprises depth and RGB images from both camera perspectives (parallel to the floor and ceiling-mounted), synchronization data, and raw accelerometer readings. Each video stream is archived separately

as a sequence of PNG images. The depth data, stored in PNG16 format, necessitates rescaling to represent depth in millimeters (D) as follows accurately:

$$D_i(x, y) = \frac{V(x, y) \cdot S_i}{65535} \quad (1)$$

Where $D_i(x, y)$ denotes the depth at position (x, y) for the i th camera, $V(x, y)$ represents the pixel value at position (x, y) in the PNG16 image, and S_i is the scale ratio for the i -th camera. The scale ratios are defined as $S_0 = 6000$ for fall sequences using camera 0, $S_1 = 3640$ for fall sequences using camera 1, and $S_0 = 7000$ for ADL sequences using camera 0. The preprocessing of video data involves a series of steps to prepare the frames for feature extraction. Initially, each video is accessed frame by frame using OpenCV's VideoCapture functionality. Subsequently, each frame is resized to a uniform dimension of 224×224 pixels to align with the input requirements of the EfficientNetB0 model. This resizing operation can be mathematically represented as a function R that maps the original frame dimensions to the target dimensions, preserving the aspect ratio and interpolating pixel values as necessary:

$$R: \mathbb{R}^{w \times h \times 3} \rightarrow \mathbb{R}^{224 \times 224 \times 3} \quad (2)$$

Where w and h denote the original width and height of the frame, respectively. After resizing, the frames undergo normalization to scale the pixel values to the $[0, 1]$ range, facilitating more stable and efficient model training. The normalization process for a frame F can be defined as:

$$F_{normalized} = \frac{F}{255} \quad (3)$$

This operation ensures that each pixel value in the frame is proportionally reduced to a decimal between 0 and 1, thus standardizing the input data for subsequent processing through the EfficientNetB0 architecture.

3.2. Feature Extraction Using EfficientNetB0

The feature extraction component of our methodology is built upon the EfficientNetB0 architecture, a cutting-edge CNN known for its scalability and efficiency. EfficientNetB0 uniformly scales the network's depth, width, and resolution, optimizing its performance across various constraints. EfficientNetB0 are its convolutional operations, which form the backbone of its feature extraction capabilities. A convolutional operation on an input image or feature map can be mathematically described as:

$$F_{out}(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b K(i, j) \cdot F_{in}(x - i, y - j) \quad (4)$$

Where F_{out} is the output feature map, F_{in} is the input image or feature map, K is the kernel or filter of size $(2a + 1) \times (2b + 1)$, and (x, y) denotes the pixel coordinates. This operation is applied across the entire input feature map, extracting features through the weighted summation of pixel values within the kernel's receptive field. EfficientNetB0 also leverages batch normalization to enhance training stability and convergence. Batch normalization can be defined as:

$$BN(x) = \gamma \left(\frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \right) + \beta \quad (5)$$

Where x is the input to the batch normalization layer, μ_B and σ_B^2 are the mean and variance of the batch, respectively, γ and β are learnable parameters of the layer, and ϵ is a small constant added for numerical stability. Furthermore, EfficientNetB0 employs depthwise separable convolutions, a technique that reduces computational cost without sacrificing depth or expressivity. A depthwise separable convolution comprises two stages: depthwise and pointwise convolution. The depthwise convolution applies a single filter per input channel, and the

pointwise convolution then combines the output channels using a $1 \times 11 \times 1$ convolution. This can be represented as:

$$DW(x, y, c) = \sum_{i=-a}^a \sum_{j=-b}^b K_c(i, j) \cdot F_{in}(x - i, y - j, c) \quad (6)$$

$$PW(x, y, c) = \sum_{c'=1}^C \hat{K}(c') \cdot DW(x, y, c) \quad (7)$$

Where DW denotes the output of the depthwise convolution for channel c , PW is the output of the pointwise convolution for channel c' , K_c is the kernel for the depthwise convolution, and $\hat{K}$ is the $1 \times 11 \times 1$ kernel for the pointwise convolution. C is the number of channels. Activation functions such as the Swish function, defined as $f(x) = x \cdot \text{sigmoid}(x)$, are applied after convolutional operations to introduce non-linearity, enabling the network to learn complex features. By integrating these elements, EfficientNetB0 provides a powerful and efficient framework.

3.3. Dynamic Blurring for Privacy Preservation

we address privacy concerns in video-based monitoring by implementing dynamic blurring for privacy preservation. This process involves the selective obfuscation of regions of interest (ROI) within video frames, specifically targeting identifiable features of individuals to maintain anonymity while preserving the utility of the data for fall detection. The identification of ROIs for blurring is governed by a detection function $D(F_{in}, \theta)$, where F_{in} represents an input frame, and θ denotes the parameters of the detection model, which may include facial recognition, pose estimation, or other relevant feature detection algorithms. The output of this function is a set of bounding boxes $B = \{b_1, b_2, \dots, b_n\}$, where each b_i specifies the coordinates and dimensions of an ROI within the frame. The dynamic blurring is then applied to these identified ROIs using a Gaussian blur operation, mathematically described as:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (8)$$

Where (x, y) are the coordinates relative to the center of the kernel, and σ is the standard deviation, which controls the extent of blurring. The size of the kernel, $k \times k$, is chosen based on the desired level of blurriness, typically set to several times the value of σ to ensure that the edges of the kernel contribute negligibly to the blur. The application of the Gaussian blur to an ROI b_i within the frame F_{in} can be represented as:

$$F_{blurred}(x, y) = (F_{in} * G)(x, y) = \sum_{m=-a}^a \sum_{n=-b}^b F_{in}(x - m, y - n) \cdot G(m, n, \sigma) \quad (9)$$

for all (x, y) within b_i , where $*$ denotes the convolution operation, and a and b are half the width and height of the Gaussian kernel, respectively.

3.4. Temporal Analysis with LSTM

The LSTM network is a specialized RNN designed to model temporal dependencies in sequence data effectively. Its architecture is uniquely suited to address the vanishing gradient problems, enabling it to capture long-term dependencies. An LSTM unit comprises three main gates: the input gate (i), the forget gate (f), and the output gate (o), each responsible for regulating the flow of information. We utilized bidirectional LSTM with the following structure, as shown in **Figure 2**.

Layer (type)	Output Shape	Param #
bidirectional_1 (Bidirectional)	(None, 400, 200)	1104800
dropout_2 (Dropout)	(None, 400, 200)	0
lstm_3 (LSTM)	(None, 100)	120400
dropout_3 (Dropout)	(None, 100)	0
dense_1 (Dense)	(None, 1)	101
=====		
Total params: 1,225,301		
Trainable params: 1,225,301		
Non-trainable params: 0		

Figure 2. The architecture used for the LSTM Network

3.5. Classification Framework

Following the extraction of temporal features, the next step involves classification, which is classified between fall and non-fall events. This process typically involves passing the LSTM output through one or more fully connected layers in a SoftMax layer for binary classification:

$$h_z = W_h \cdot h_t + b_h \quad (10)$$

$$p = \text{softmax}(z) = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (11)$$

where h_t is the output from the LSTM at time t , W_h and b_h are the weights and biases for the dense layer, respectively, z is the logit, and p represents the probabilities for each class obtained through the SoftMax function. The class with the highest probability is selected as the predicted class for each input sequence. This framework facilitates the effective classification of video sequences into fall or non-fall categories based on the temporal patterns identified by the LSTM network.

3.6. Training Process

The training of the integrated model is underpinned by a mathematical framework that includes the definition of a loss function, the selection of an optimization algorithm, and the application of regularization techniques to prevent overfitting. The loss function L quantifies the discrepancy between the predicted outputs p and the true labels y . For binary classification tasks, such as fall detection, the binary cross-entropy loss is commonly used:

$$L(y, p) = \frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (15)$$

Where N is the number of samples, y_i is the true label, and p_i is the predicted probability for the i -th sample. The optimization of the model parameters is achieved through Stochastic Gradient Descent (SGD), which iteratively updates the weights W based on the gradients of the loss function:

$$W_{t+1} = W_t - \eta \nabla L \quad (16)$$

Where η is the learning rate, and ∇L denotes the gradient of the loss function for the weights at time t . L2 regularization and dropout techniques are applied to mitigate overfitting by adding a penalty term to the loss function or randomly omitting units from the network during training, respectively. The backpropagation process facilitates the computation of gradients ∇L through the network, employing the chain rule to propagate errors from the output layer back through the LSTM and EfficientNetB0 layers, enabling the model to learn and adjust its parameters to minimize the loss function.

4. Results and Analysis

Figure 3. (Left) provides ‘Model Accuracy’ plots, which show the proportion of correctly classified instances (accuracy) against the number of epochs for both the training and validation datasets. It is observed that the training accuracy shows a consistent upward trend, indicating that the model is learning and improving its performance on the training data as the epochs progress. While generally following an upward trajectory, the validation accuracy exhibits some fluctuations. This could indicate the model’s encounters with challenging or previously unseen data in the validation set. These expected fluctuations indicate how the model might perform when exposed to new data. It is worth noting that both the training and validation accuracies converge to high values close to 1.0, suggesting that the model has achieved a high level of proficiency in distinguishing between fall and non-fall events. **Figure 3. (Right)** graph shows the model’s loss over the same number of epochs. For the training set, the loss decreases sharply and continues to decline steadily, which is typical behavior as the model adjusts its weights to minimize the prediction error. For the validation set, the loss decreases in tandem with the training loss but with notable spikes at specific points. These spikes often signify that the model made predictions significantly off the actual labels for some batches in the validation set. This can happen if the model encounters data points that differ from the learned patterns during training.

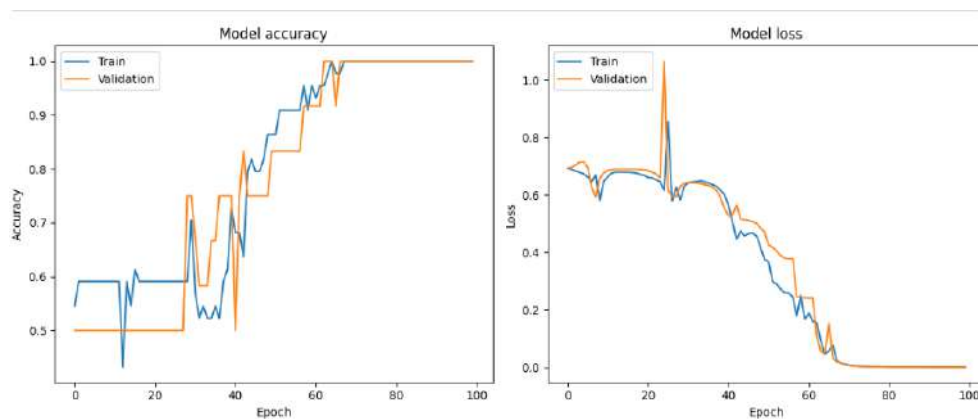


Figure 3. Accuracy and Loss graphs of the proposed LSTM model

The confusion matrix in **Figure 4.** provides a quantitative assessment of the model’s classification accuracy. It shows the number of true positive (TP) and true negative (TN) predictions, along with false positive (FP) and false negative (FN) predictions. In this case, the matrix reveals perfect classification on the test data, with all fall events being correctly identified (6 TP) and no ADL events being misclassified as falls (0 FN), implying an exceptional level of model performance.

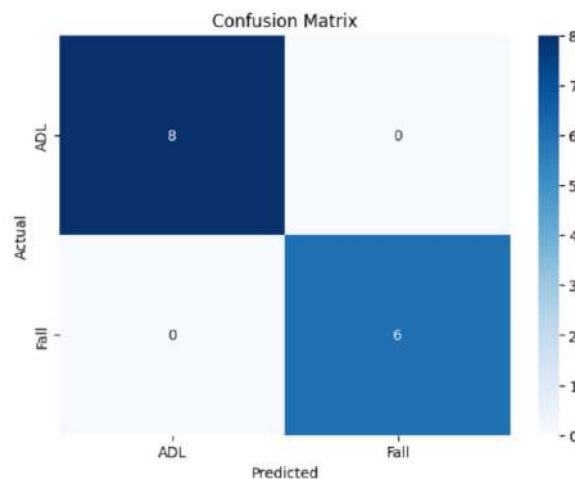


Figure 4. Confusion metric of unseen test videos

The Receiver Operating Characteristic (ROC) curve in **Figure 5** plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The area under the curve (AUC) in this ROC curve approaches 1, which suggests excellent model performance, with a high true positive rate and a low false positive rate across threshold values.

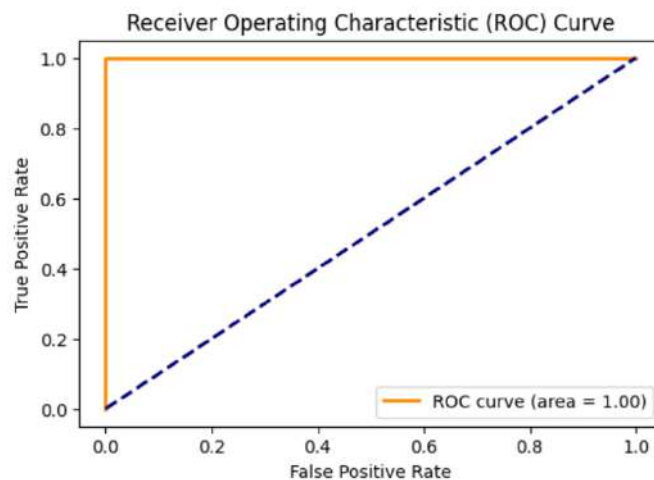


Figure 5. ROC curve of the proposed model on unseen test data

Figure 6 shows the model's performance on real test videos; the model correctly predicted this particular scenario as a 'Fall,' corroborated by the RGB image on the right. This clearly shows an individual in a prone position on the floor. Moreover, the depth image reveals the successful application of the dynamic blurring method. The individual's features are indistinguishable, and the privacy-preserving objective of the method is evident. The contours and the general posture of the person are discernible, which is sufficient for fall detection purposes, but the finer details necessary for personal identification have been effectively obfuscated. The blurring technique implemented in the system is designed to activate upon detecting a human figure within the video frame, applying a Gaussian blur where the person is detected. This ensures that any potentially sensitive information is rendered non-identifiable, addressing privacy concerns paramount in real-world applications of surveillance-based systems. The obscured depth image confirms that the privacy-preserving measures do not impede the algorithm's ability to detect a fall.

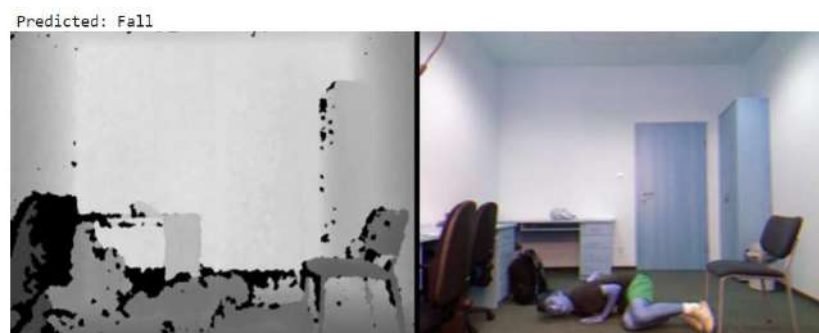


Figure 6. Model evaluation on the test video from the fall folder

Figure 7 shows the system's prediction for this scene, labeled 'ADL,' which is validated by the RGB image on the right. It depicts an individual in an upright position, supporting the prediction that no fall has occurred. The prediction's accuracy is a testament to the model's ability to effectively discern between falls and non-fall events. Furthermore, similar to the previous fall scenario, the depth image demonstrates the application of the dynamic blurring technique. The individual's detailed features are indistinct, ensuring privacy is maintained. Despite the blurring, essential characteristics for ADL recognition, such as the vertical orientation of the body and the absence of unusual postures associated with falls, are preserved and remain detectable by the system.

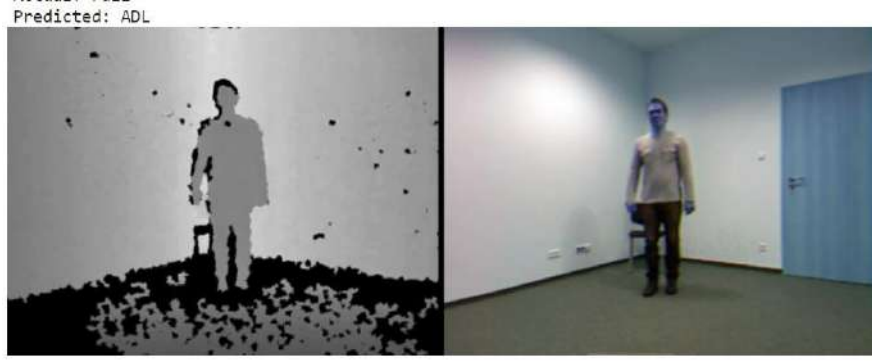


Figure 7. Model Evaluation on the test video from the ADL folder

The analysis of the presented results underscores the robustness and reliability of the implemented fall detection model. This is evidenced by the convergence of the accuracy and loss metrics, the unequivocal classification outcomes depicted in the confusion matrix, and the favorable diagnostic characteristics portrayed by the ROC curve. These results collectively affirm the model's efficacy in accurately detecting fall events. It preserves the privacy of individuals through dynamic blurring, as no identifiable features are discernible in the depth visualizations.

Table 1. Comparative analysis of results with other state-of-the-art models

Model/Study	TPR	TNR	Accuracy
Eltahir et al. [40]	95.88	97.02	97.56
Chan Su [38]	98.07	99.03	98.06
Single stream (RGB) [22]	100	96.61	96.99
Single stream (OF) [22]	100	96.34	96.75
Multi-stream (RGB+OF+PE) [22]	100	98.61	98.77
EfficientNet-B0 [41]	93.33	100	97.14
Improved YOLOv5s [39]	–	–	97.2
A single-frame human binary image with YOLOv5s [42]	–	–	96.7
Our Method	100	100	100

As shown in **Table 1**, the comparative analysis of fall detection methodologies yields a substantive understanding of the advancements and varying efficacies of diverse approaches in this research domain. The table encapsulates the True Positive Rate (TPR), True Negative Rate (TNR), and overall Accuracy, serving as pivotal metrics for the assessment of each method. The model presented by Eltahir et al. [40] manifests a commendable balance between sensitivity and specificity, with a TPR of 95.88% and a TNR of 97.02%, culminating in an accuracy of 97.56%. Chan Su's model slightly improves sensitivity at 98.07% and specificity at 99.03%, with an analogous accuracy of 98.06%. These two models set a robust baseline in fall detection, evidencing high efficacy. The single-stream models using RGB and Optical Flow (OF) data individually attain a TPR of 100%, indicative of their flawless identification of fall events. However, their specificity scores, 96.61% and 96.34%, respectively, although high, suggest a slightly less robust capacity to classify non-fall activities accurately. This slight discrepancy is reflected in their accuracy scores, which, while impressive at 96.99% and 96.75%, do not reach the pinnacle of Chan Su's model.

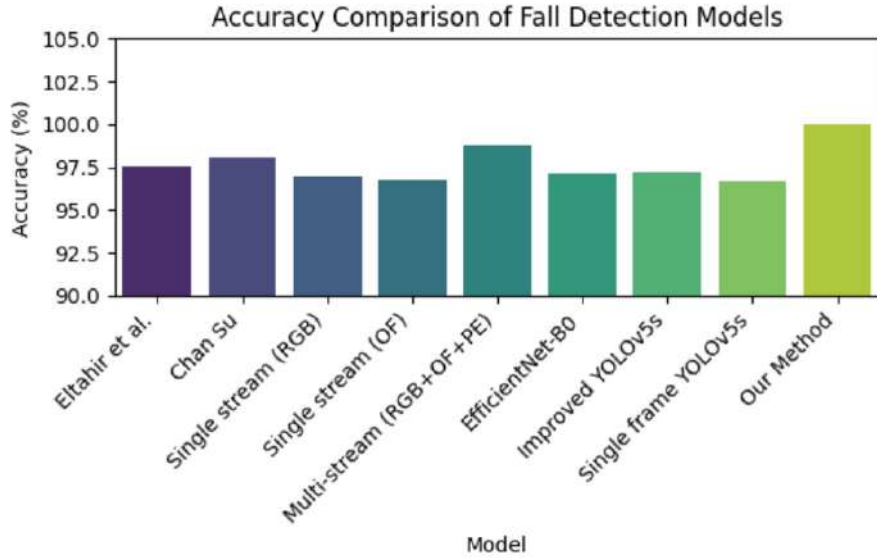


Figure 8. Comparison of Accuracy with other state-of-the-art models

The multi-stream approach amalgamating RGB, OF, and Pose Estimation (PE) data represents a significant leap forward, yielding a perfect TPR and an enhanced TNR of 98.61%, leading to an accuracy of 98.77%. This approach underscores the utility of integrating multiple data streams for improved specificity without compromising sensitivity. EfficientNet-B0, despite a lower TPR of 93.33%, achieves a perfect TNR of 100%. This accentuates the model's exceptional performance in identifying non-fall events, though it falls short of the multi-stream model's balanced accuracy. The improved YOLOv5s model and the single frame human binary image approach using YOLOv5s do not disclose TPR or TNR but report accuracies of 97.2% and 96.7%, respectively. While these figures suggest competent models, the lack of detailed TPR and TNR data precludes a complete comparative analysis. Our proposed methodology establishes a new benchmark, recording a flawless TPR and TNR of 100% and an unmatched accuracy of 100%. This unprecedented performance indicates a superior ability to correctly identify fall incidents and an unparalleled precision in confirming non-fall activities.

5. Conclusion

In this research, we have successfully developed and evaluated a novel video-based fall detection system that prioritizes privacy without compromising the real-time detection efficacy of elderly falls. By integrating EfficientNetB0 and LSTM networks, our methodology ensures robust feature extraction and accurate fall event classification. The introduction of dynamic blurring as a privacy-preserving technique represents a significant advancement, allowing for anonymizing identifiable features within video frames while maintaining the system's operational integrity. Our findings reveal that this approach achieves perfect accuracy, recall, precision, and area AUC scores. It also effectively addresses the critical privacy concerns of video surveillance in sensitive environments such as homes and elderly care facilities. Implementing dynamic blurring ensures that the privacy of monitored individuals is safeguarded, setting a new precedent in the ethical application of surveillance technologies in healthcare.

Our future research will focus on further enhancing the adaptability and generalizability of the system across diverse settings and populations. This includes exploring additional privacy-preserving mechanisms and integrating multimodal data sources to enrich the system's contextual understanding. This research contributes significantly to elderly care technology, presenting a practical solution to the long-standing challenge of balancing effective fall detection with stringent privacy requirements. Our work advances the technological capabilities in this domain and addresses critical ethical considerations. This paves the way for broader acceptance and deployment of video-based monitoring systems in healthcare settings.

References

- [1] W. W. Fu, T. S. Fu, R. Jing, S. R. McFaull, and M. D. Cusimano, "Predictors of falls and mortality among elderly adults with traumatic brain injury: a nationwide, population-based study," *PloS One*, vol. 12, no. 4, p. e0175868, 2017.
- [2] A. Kehoe, J. E. Smith, A. Edwards, D. Yates, and F. Lecky, "The changing face of major trauma in the UK," *Emerg. Med. J.*, vol. 32, no. 12, pp. 911–915, 2015.
- [3] S. K. Gharghan and H. A. Hashim, "A comprehensive review of elderly fall detection using wireless communication and artificial intelligence techniques," *Measurement*, p. 114186, 2024.
- [4] D. Egeonu and B. Jia, "A systematic literature review of computer vision-based biomechanical models for physical workload estimation," *Ergonomics*, pp. 1–24, Jan. 2024.
- [5] P. Khatiwada, B. Yang, J.-C. Lin, and B. Blobel, "Patient-Generated Health Data (PGHD): Understanding, Requirements, Challenges, and Existing Techniques for Data Security and Privacy," *J. Pers. Med.*, vol. 14, no. 3, p. 282, 2024.
- [6] M. Qaraqe *et al.*, "PublicVision: A Secure Smart Surveillance System for Crowd Behavior Recognition," *IEEE Access*, vol. 12, pp. 26474–26491, 2024.
- [7] R. Rajagopalan, I. Litvan, and T.-P. Jung, "Fall prediction and prevention systems: recent trends, challenges, and future research directions," *Sensors*, vol. 17, no. 11, p. 2509, 2017.
- [8] V. Mehta, A. Dhall, S. Pal, and S. S. Khan, "Motion and region aware adversarial learning for fall detection with thermal imaging," in *2020 25th international conference on pattern recognition (ICPR)*, IEEE, 2021, pp. 6321–6328.
- [9] S. Ravi, P. Climent-Pérez, and F. Florez-Revuelta, "A review on visual privacy preservation techniques for active and assisted living," *Multimed. Tools Appl.*, vol. 83, no. 5, pp. 14715–14755, Jul. 2023, doi: 10.1007/s11042-023-15775-2.
- [10] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sens. J.*, vol. 15, no. 8, pp. 4544–4553, 2015.
- [11] Alrasheedy, M. N., Muniyandi, R. C., & Fauzi, F. (2022, October). Text-Based Emotion Detection and Applications: A Literature Review. In 2022 International Conference on Cyber Resilience (ICCR) (pp. 1-9). IEEE.
- [12] Kwong, A., Muzamal, J. H., & Khan, Z. (2022, November). Privacy Pro: Spam Calls Detection Using Voice Signature Analysis and Behavior-Based Filtering. In 2022 17th International Conference on Emerging Technologies (ICET) (pp. 184-189). IEEE.
- [13] E. Torti *et al.*, "Embedded real-time fall detection with deep learning on wearable devices," in *2018 21st euromicro conference on digital system design (DSD)*, IEEE, 2018, pp. 405–412.
- [14] Y. S. Delahoz and M. A. Labrador, "Survey on fall detection and fall prevention using wearable and external sensors," *Sensors*, vol. 14, no. 10, pp. 19806–19842, 2014.
- [15] S. Nooruddin, M. M. Islam, F. A. Sharna, H. Alhetari, and M. N. Kabir, "Sensor-based fall detection systems: a review," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 5, pp. 2735–2751, 2022.
- [16] A. Singh, S. U. Rehman, S. Yongchareon, and P. H. J. Chong, "Sensor technologies for fall detection systems: A review," *IEEE Sens. J.*, vol. 20, no. 13, pp. 6889–6919, 2020.
- [17] B. Koonce, "EfficientNet," in *Convolutional Neural Networks with Swift for Tensorflow*, Berkeley, CA: Apress, 2021, pp. 109–123. doi: 10.1007/978-1-4842-6168-2_10.
- [18] A. Graves, "Long Short-Term Memory," in *Supervised Sequence Labelling with Recurrent Neural Networks*, vol. 385, in Studies in Computational Intelligence, vol. 385. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 37–45. doi: 10.1007/978-3-642-24797-2_4.
- [19] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 1, pp. 156–167, 2006.
- [20] A. K. Bourke, J. V. O'brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait Posture*, vol. 26, no. 2, pp. 194–199, 2007.
- [21] D. A. Ganz, Y. Bao, P. G. Shekelle, and L. Z. Rubenstein, "Will my patient fall?," *Jama*, vol. 297, no. 1, pp. 77–86, 2007.

- [22] S. A. Carneiro, G. P. da Silva, G. V. Leite, R. Moreno, S. J. F. Guimaraes, and H. Pedrini, "Multi-stream deep convolutional network using high-level features applied to fall detection in video sequences," in *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, IEEE, 2019, pp. 293–298. Accessed: Mar. 18, 2024.
- [23] N. Kaur, S. Rani, and S. Kaur, "Real-time video surveillance based human fall detection system using hybrid haar cascade classifier," *Multimed. Tools Appl.*, pp. 1–19, 2024.
- [24] A. Núñez-Marcos and I. Arganda-Carreras, "Transformer-based fall detection in videos," *Eng. Appl. Artif. Intell.*, vol. 132, p. 107937, 2024.
- [25] J. Moore *et al.*, "Contextualizing remote fall risk: Video data capture and implementing ethical AI," *NPJ Digit. Med.*, vol. 7, no. 1, p. 61, 2024.
- [26] V. Fula and P. Moreno, "Wrist-Based Fall Detection: Towards Generalization across Datasets," *Sensors*, vol. 24, no. 5, p. 1679, 2024.
- [27] A. Bansal, R. Sharma, and M. Kathuria, "A Vision-Based Approach to Enhance Fall Detection with Fine-Tuned Faster R-CNN," in *2023 International Conference on Advanced Computing & Communication Technologies (ICACCTech)*, IEEE, 2023, pp. 678–684.
- [28] J. Gutiérrez, V. Rodríguez, and S. Martin, "Comprehensive review of vision-based fall detection systems," *Sensors*, vol. 21, no. 3, p. 947, 2021.
- [29] S. Ezatzadeh and M. R. Keyvanpour, "Fall detection for elderly in assisted environments: Video surveillance systems and challenges," in *2017 9th international conference on information and knowledge technology (ikt)*, IEEE, 2017, pp. 93–98. Accessed: Mar. 19, 2024.
- [30] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and Adaboost-based classification," *J. Electron. Imaging*, vol. 22, no. 4, pp. 041106–041106, 2013.
- [31] F.-Y. Leu, C.-Y. Ko, Y.-C. Lin, H. Susanto, and H.-C. Yu, "Fall detection and motion classification by using decision tree on mobile phone," in *Smart Sensors Networks*, Elsevier, 2017, pp. 205–237. Accessed: Mar. 19, 2024.
- [32] Y.-Z. Hsieh and Y.-L. Jeng, "Development of home intelligent fall detection IoT system based on feedback optical flow convolutional neural network," *Ieee Access*, vol. 6, pp. 6048–6057, 2017.
- [33] P. Vallabh and R. Malekian, "Fall detection monitoring systems: a comprehensive review," *J. Ambient Intell. Humaniz. Comput.*, vol. 9, no. 6, pp. 1809–1833, 2018.
- [34] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *Biomed. Eng. OnLine*, vol. 12, no. 1, p. 66, 2013, doi: 10.1186/1475-925X-12-66.
- [35] Y. Fan, M. D. Levine, G. Wen, and S. Qiu, "A deep neural network for real-time detection of falling humans in naturally occurring scenes," *Neurocomputing*, vol. 260, pp. 43–58, 2017.
- [36] D. Singh, M. Gupta, and R. Kumar, "BGR Images-Based Human Fall Detection Using ResNet-50 and LSTM," in *Third Congress on Intelligent Systems*, vol. 608, S. Kumar, H. Sharma, K. Balachandran, J. H. Kim, and J. C. Bansal, Eds., in *Lecture Notes in Networks and Systems*, vol. 608., Singapore: Springer Nature Singapore, 2023, pp. 175–186. doi: 10.1007/978-981-19-9225-4_14.
- [37] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 1, pp. 314–323, 2018.
- [38] C. Su, J. Wei, D. Lin, L. Kong, and Y. L. Guan, "A novel model for fall detection and action recognition combined lightweight 3D-CNN and convolutional LSTM networks," *Pattern Anal. Appl.*, vol. 27, no. 1, pp. 1–16, 2024.
- [39] T. Chen, Z. Ding, and B. Li, "Elderly fall detection based on improved YOLOv5s network," *IEEE Access*, vol. 10, pp. 91273–91282, 2022.
- [40] M. M. Eltahir *et al.*, "Deep Transfer Learning-Enabled Activity Identification and Fall Detection for Disabled People.," *Comput. Mater. Contin.*, vol. 75, no. 2, 2023, Accessed: Mar. 18, 2024.
- [41] S. Hwang, M. Ki, S.-H. Lee, S. Park, and B.-K. Jeon, "Cut and continuous paste towards real-time deep fall detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 1775–1779. Accessed: Mar. 18, 2024.
- [42] Y. Wang, R. Song, and X. Zhang, "Real-time human fall recognition based on deep learning methods and single depth image with privacy requirements," in *2022 37th Youth Academic*

Annual Conference of Chinese Association of Automation (YAC), IEEE, 2022, pp. 1548–1553.
Accessed: Mar. 18, 2024.

- [43] B. Kwolek and M. Kepski, “Human fall detection on embedded platform using depth maps and wireless accelerometer,” *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, 2014.

Computer Modelling, Analysis of the Main Information Properties of Memristor and Its Application in Secure Communication System

Volodymyr Rusyn¹, Sergey Subbotin², George Vorobets¹ and Oleksandr Vorobets¹

¹ Yuriy Fedkovych Chernivtsi National University, Kotsybinsky str. 2, Chernivtsi, 58012, Ukraine

² National University "Zaporizhzhia polytechnic", Zhukovsky str., 64, Zaporizhzhia, 69063, Ukraine

Abstract

In this paper, we present equation and system parameters that realize memristive chaotic system. The analysis and computer modelling of the main information properties of memristor are shown. For this, we created programming interface using one of the modern software LabView that allows to generate, analysis and research of the main information properties of memristor, focusing on time series of the three coordinates and phase portraits. The created another programming interface that demonstrates the masking and decrypt of the information are also presented.

Keywords

Memristor, information properties, LabView, security system

1. Introduction

Chaos theory has been established since the 1970's due to its applications in many different engineering research areas, such as robotics [1], electronic circuits of chaotic generators [2-11], information security [12-14], etc. Many chaotic systems are realized using Arduino [15] and FPGA [16-21].

In 1971, Leon Chua suggested the existence of another basic element of the electrochemical system, which realizes the connection between magnetic flux and charge. In 2008, scientists managed to discover a new component. This component is called a memristor, this term was formed as a result of merging the words "memory" and "resistor" (electrical resistance). This term refers to a bipolar device, the electrical resistance of which varies depending on the amount of charge flowing through it. The resistance of the memristor is called the memristor (M), which is defined as the ratio of the change in the magnitude of the magnetic field flux to the change in charge, and its value depends on how long the electric current flows through the memristor, that is, the amount of charge that passed through it.

The main advantage of a memristor is properties that are not stored in the form of a charge. These properties differ from most types of modern semiconductor memory elements. This is the main advantage, because it is not afraid of charge leakage, which is the main evil that everyone wants to get rid of when switching to "nanochips". Another advantage of the memristor is its energy independence. These properties ensure that the memristor is stored at the same time as the existing materials used to make it. The successful production of the memristor by Hewlett Packard (HP) greatly increased the interest of researchers in the memristor. Many scientists began to use the memristor in analog systems and in digital media. Others have used memristors as resistive memory modules and logic programs. The potential of using a memristor in neuromorphic applications has also been studied by many researchers [22-25].

Memristors, first of all, are extremely promising elements of binary non-volatile memory, in which information is recorded by switching electrical resistance between resistive states with low (ON) and high (OFF) resistance. Moreover, their electronic states "adjust" to input signals,

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ rusyn_v@ukr.net (V. Rusyn); subbotin@zntu.edu.ua (S. Subbotin); g.vorobets@chnu.edu.ua (G. Vorobets); o.vorobets@chnu.edu.ua (O. Vorobets)

🆔 0000-0001-6219-1031 (V. Rusyn); 0000-0001-5814-8268 (S. Subbotin); 0000-0001-8125-2047 (G. Vorobets); 0000-0003-3195-8214 (O. Vorobets)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

just as living organisms do. This is why memristors can imitate the functioning of biological synapses connecting neurons in the brain. The main function of the latter is to transmit a signal from one neuron to another, and the magnitude of the connection between neurons depends on the relative time of their “firing”. It is this mechanism that is responsible for associative learning, that is, the brain’s ability to find connections between different events, so the memristor can be considered a prototype of an “electronic synapse”.

Research in the field of artificial intelligence, and more specifically on the creation of an artificial brain based on memristors, is also being conducted at Michigan State University under the leadership of Wei Lu. Here, a memristor model was built based on a layer of a mixture of silver and silicon and tungsten electrodes, and in the near future the scientists plan to create large circuits consisting of thousands of such elements.

The already studied properties of memristors allow us to say that on their basis it is possible to create computers of a fundamentally new architecture, with performance significantly exceeding semiconductor ones. Modern computers are built on the von Neumann architecture: both data and programs are stored in the machine’s memory in binary code, with the computing module separated from the storage devices, and the programs executed sequentially, one after the other. Progressive in the middle of the last century, such an architecture today no longer meets the requirements for computer technology: programs have become much more complex, and the volume of processed data has grown by orders of magnitude, if not tens of orders of magnitude.

A computer based on memristors could be a significant step forward, since it is capable of simulating the functioning of the human brain, which does not have any single center for collecting and processing information. Each block receives, processes and transmits its own data arrays to other blocks, muscles, and sensory organs, which are insignificant compared to the entire volume of incoming information. According to recent estimates, to build a model of the human cerebral cortex from modern computer components, you will need at least 150,000 processors and 144 TB of RAM alone, and we are not even talking about the intelligence of a baby.

In a memristor computer, many modules operate in parallel and independently of each other, and the ability to store and operate an unlimited set of values from 0 to 1 means that executable programs are not limited to binary code. Moreover, individual computer hardware components - processors, video chips, memory and hard drives - will become unnecessary in principle; the machine will be an architecturally homogeneous device where all data will be simultaneously stored and all operations with it will be carried out. For an upgrade, it will be enough to install additional memristor modules, and for repairs, replace the failed ones.

The memristor computer will not need to be “booted”: immediately after turning it on, it will be ready to continue working, and from the very point where it was interrupted. Compared to modern technology, the energy consumption of memristor machines will be negligible, and the computing power will be simply gigantic.

Considering that there is literally one step left before the mass production of memristors, it is very possible that a memristor computer will become an intermediate step on the path to a quantum computer.

2. Computer modelling of the main information properties of memristor

The common thread that binds these four elements together as the four basic elements of circuit theory is the fact that the characteristics of these elements relate the four variables in electrical engineering (voltage, current, flux and charge) intimately. Fig. 1 shows this relationship graphically.

Nonlinear behavior of memristive information properties is described by next equation system (1):

$$\begin{cases} \dot{x} = \frac{y}{c}; \\ \dot{y} = \frac{-1}{L} [x + \beta(z^2 - 1)y]; \\ \dot{z} = -y - \alpha z + yz. \end{cases} \quad (1)$$

A chaotic time series is a sequence of data that exhibits chaotic behavior over time when plotted. A characteristic of chaotic signals is that they are sensitive to initial conditions. This means that a small change in the initial conditions can cause the results to look very different over time. Chaotic signals often look chaotic and unpredictable. They look like random noise, but they have a underlying deterministic dynamic.

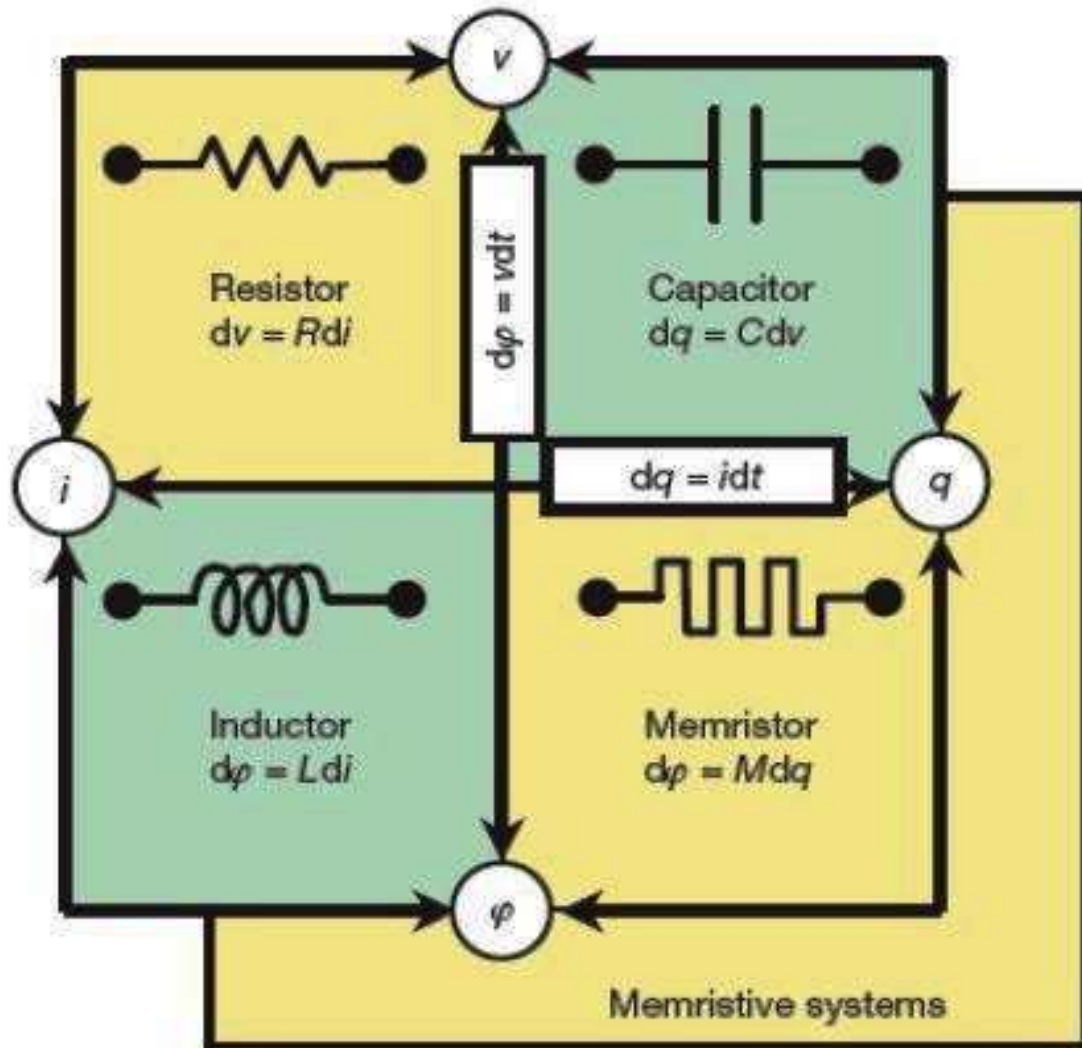


Figure 1: The four basic circuit elements

The logistic map, the Henon map, the Lorenz and Rossler systems are examples of chaotic signals [26-30]. These chaotic signals used in different physical systems and mathematical models, such as economics, radioengineering, physics, mathematics [31-43].

Time series for X, Y and Z coordinates are shown in Fig. 2. The parameter values are: $C = 1$, $L = 3$, $\alpha = 0.6$, $\beta = 1.5$, $dt = 0.01$, number of iterations $N = 10000$. For modelling was used modern software environment LabView.

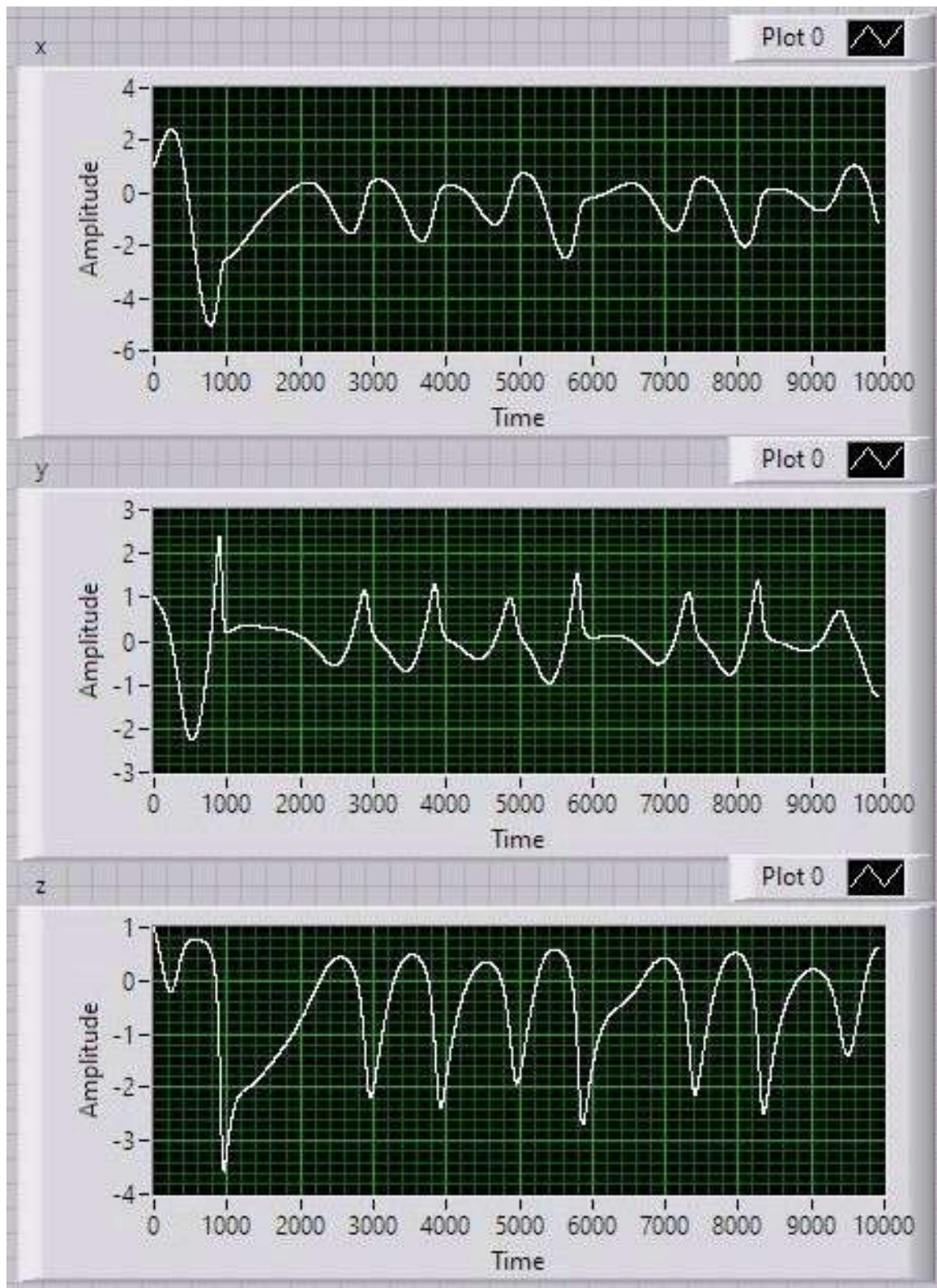


Figure 2: Time series for X, Y and Z coordinates

Until recently, specialists, when solving problems in their own subject area, were forced to resort to the help of professional programmers, who, as a rule, were not knowledgeable in this area. Such intermediation most often increased material and time costs, and most importantly, reduced the quality of research and development. Even the use of specialized software only partially solved this problem. The emergence of the latest generation of software products with very user-friendly interfaces, adapted to the mentality and professional skills of specialists, has made it possible for specialists to use them directly, without resorting to the help of intermediaries. These new software products include LabVIEW, which has a very convenient user

interface and powerful graphical programming tools. With each subsequent version of LabVIEW, the level of intellectualization of the user interface and ease of use increases.

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is a platform and development environment for a visual programming language created by National Instruments. The purpose of this language is to automate the use of computational and measuring laboratory equipment.

For more than 35 years, engineers and scientists have used NI LabVIEW to develop measurement systems, test benches, and control systems. LabVIEW is based on the graphical programming language G. In addition to being programmable, LabVIEW provides a wide range of tools and libraries, from interactive wizards and user interfaces to built-in compilers, linkers, and debugging tools.

There are two main differences between LabVIEW and other programming languages. First, LabVIEW implements the concept of graphical programming, so the source code is a block diagram (icons of language elements connected together), which is then compiled into machine code. Despite this approach, LabVIEW uses the same programming constructs and techniques as other languages: data types, loops, variables, recursion, event handling, and object-oriented programming.

The second distinctive feature of LabVIEW is its support for executing code written in the G language in a data stream mode (stream programming), while traditional text-based languages (for example, C and C++) provide code execution as a sequence of commands. Stream programming languages (such as G, Agilent VEE, Microsoft Visual Programming Language and Apple Quartz Composer) are based on the concept of data flow, which determines the sequence of execution of the functional units of the program.

At first it may seem that the difference between this approach and the traditional one is not significant, but in practice it turns out differently. Namely, stream programming in the LabVIEW environment allows the developer to fully focus on the data and how to process it. Program nodes - functions, loops and other language constructs - receive data through inputs, process them and output data using outputs. As soon as the parameter values arrive at each of the input terminals of the node, the node code is executed (processing of the incoming data), after which the values of the output parameters are available at the output terminals of the node for their further transmission to other nodes according to the logic of the data flow. Accordingly, of two sequentially connected nodes, the second one can be executed only after receiving data from the previous one.

Because the G language concept is easy to understand, LabVIEW provides equally user-friendly and intuitive development environment tools. For example, unique debugging tools allow you to visually display the process of data propagation along wires, as well as display the corresponding values at the inputs and outputs of code nodes (we are talking about execution animation).

LabVIEW's graphical language and flow programming concept allows you to solve problems more conveniently and efficiently than traditional text-based languages. The key features of LabVIEW programming—namely, intuitive, graphical code and data-driven execution—make programming more thought-like than other languages. Despite the high level of code abstraction, the performance of programs written in LabVIEW remains comparable to languages like C, thanks to the built-in code compiler.

Another information property of memristor are phase portraits (Fig. 3).

For qualitative analysis of nonlinear circuits, the phase space method is used. The current values of the state variables of the analyzed circuit are plotted on its coordinate axes. Their changes in time correspond to certain curves—phase trajectories—in the space of state variables. The set of all phase trajectories—the phase portrait of the circuit—represents a comprehensive description of all its possible modes.

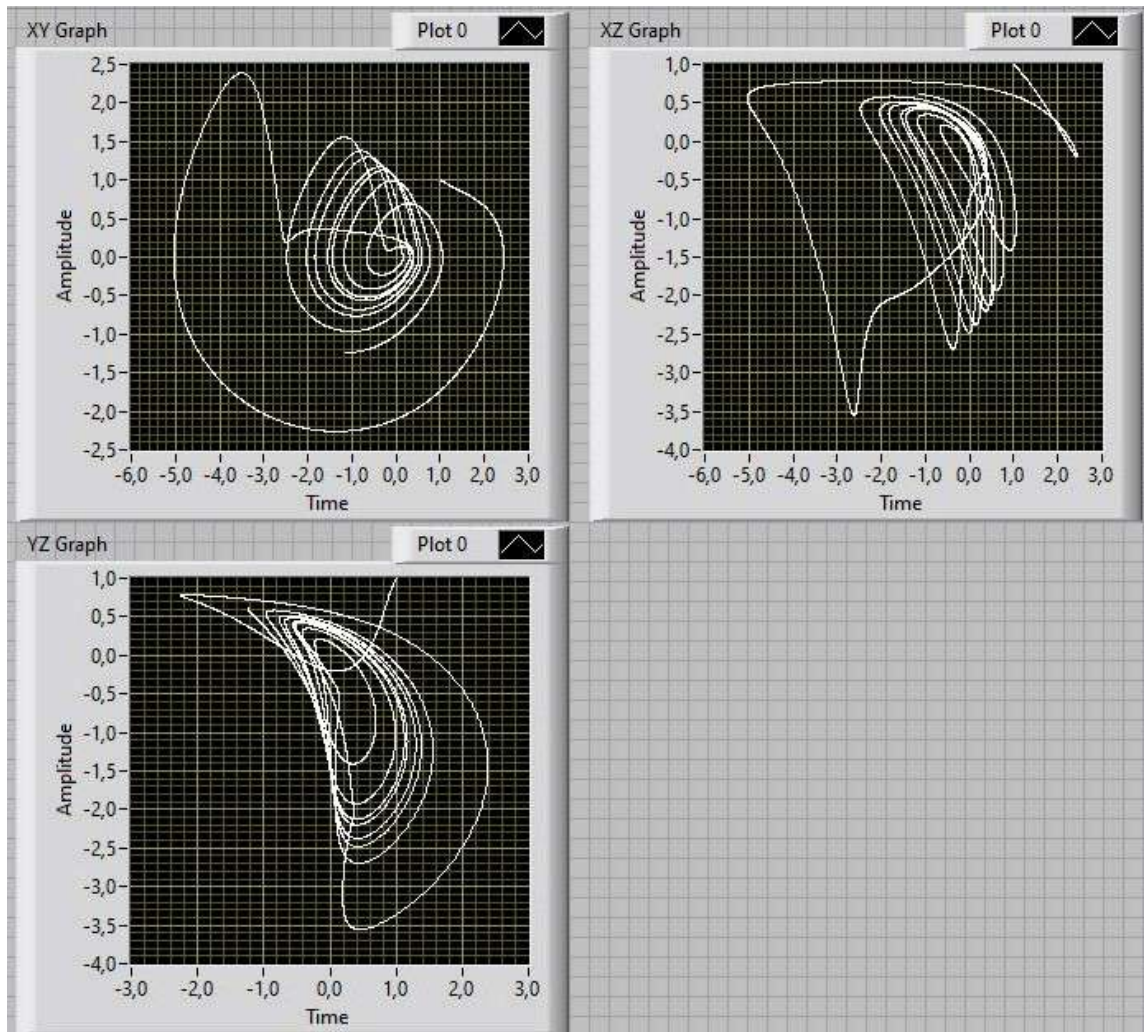


Figure 3: Phase portraits in XY, XZ and YZ projections

3. Masking and decryption of the information using memristor

Protecting information from unauthorized access is one of the most pressing tasks in telecommunications. Often in telecommunication systems, we deal with digital signals as an information carrier, for which there are already effective protection methods based on cryptography algorithms. In the case of analog signals, other methods of information protection are required, one of which is the masking method. The essence of this method is that it is necessary to find a specific signal source that has the same spectral properties as the protected information signal. Accordingly, if such a signal is mixed with an information signal, the resulting signal turns out to be uninformative, and as a result, the information signal turns out to be protected.

The last decade has been known for extreme interest in the possibility of using dynamic chaos for data encryption. At a conceptual level, there is a peculiar relationship between chaotic systems and cryptographic systems. Therefore, in nonlinear dynamics, and in cryptography, nonlinear transformation of information materializes. Such a transformation is deterministic (for example, performed by a computer), on the other hand, it must be unpredictable for an external observer. Therefore, the word “deterministic chaos” is quite “suitable” for cryptography.

The block scheme demonstrates process of transmitting and receiving information using chaotic signal for masking and decryption (Fig. 4). In this case, the transmitter and the receiver systems are identical.

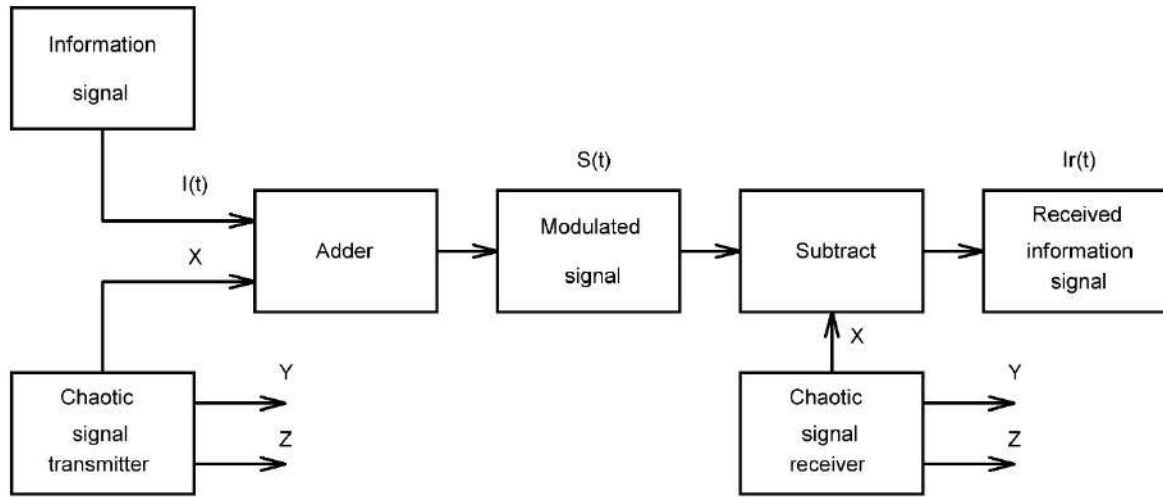


Figure 4: The block scheme for transmitting and receiving information using chaotic signal for masking and decryption

The masking of the information based on nonlinear properties is provided by blending information with the chaotic signal that was generated of the different coordinates. As an information, there was used a sinusoidal signal with amplitude of 5 V and system parameters $\alpha = 0.6$, $\beta = 1.5$, $C = 1$, $L = 3$, $dt = 0.01$, number of iterations $N = 10000$, dynamic variables $x = y = 1$. Dynamic variables and system parameters are the keys for the masking information.

Algorithm for the decryption has opposite effect.

The program interface, which demonstrates the masking and decrypt of the information based on a properties of memristor system (1) is presented in Fig. 5.

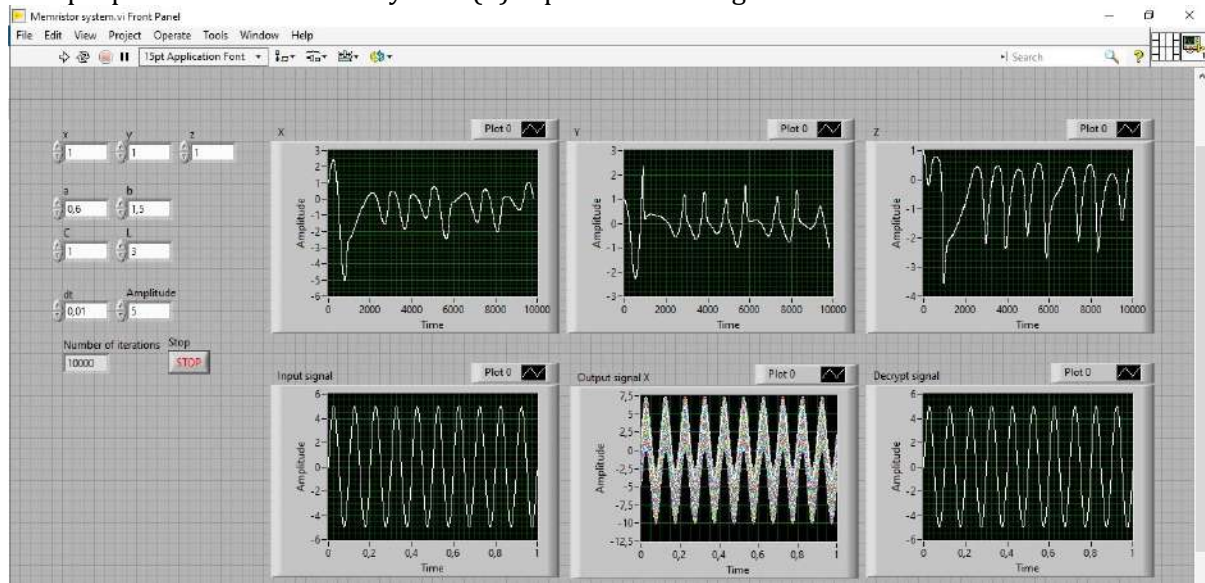


Figure 5: Software interface shows masking and decryption of the information

4. Discussion

A memristor is a two-contact device, which means that a memristor can be integrated into multilayer matrices of mutually perpendicular metal nanowires. Its architecture is significantly more compact compared to modern semiconductor silicon processors, besides, the logic of controlling such a matrix of intersecting nanowires is significantly simpler. The fundamental and main difference between a memristor and most types of modern semiconductor memory is that it does not retain its properties in the form of a charge. A memristor is not characterized by charge leakage during recording, thus it can be said that it is completely non-volatile. It is implied that the information recorded in the memristor can store

digital information for several tens of years. The duration of information storage depends on the material of the memristor.

Compared to these digital memory stores, the memristor has a much longer lifetime. With the help of memristor technology, it is possible to design a new generation of computer equipment that will surpass computers in such parameters as: speed, energy efficiency, the ability to process large streams and volumes of information. Memristor technology is also widely used in robotics. A memristor can be used as a bipolar key, a memory element, or a component of a logic element.

5. Conclusions

There was used one of the modern software LabView (LabView-2020 (64-bit version for Windows) for the analysis and modelling of the main information properties of memristor. Equation and system parameters that realized memristive chaotic system are shown. The main information properties of memristor such as a time series of the three chaotic X, Y and Z coordinates and phase portraits in XY, XZ and YZ projections are presented. Masking and decrypting of the information are presented using programming interface. Memristor that generate nonlinear oscillations can be used as one of the main parts in modern communication systems for security of information.

References

- [1] P. Milicka, P. Cížek, and J. Faigl, On chaotic oscillator-based central pattern generator for motion control of hexapod walking robot. *CEUR Workshop Proceedings* 1649 (2016): 131-137.
- [2] M. Guzan, M. Olejar, V. Cviklovic, and J. Petrzela, Simulation of the Chua's Circuit with Audio Output, in: 26th International Conference Radioelektronika, Kosice, 2016, art. no. 121867.
- [3] S. Vaidyanathan, A. Sambas, D. Clemente-Lopez, J.M. Munoz-Pacheco, A.S.T. Kammogne, V.F. Mawamba, S.S. Martin, S.A. Safaan, Electronic Circuit Design and FPGA-Based Hardware Implementation of a New Multistable 4-D Hyperchaotic Four-Leaf System. *International Journal of Computer Applications in Technology*, 72 (2023): 13-28. DOI: 10.1504/IJCAT.2023.132548.
- [4] T. Bonny, S. Vaidyanathan, A. Sambas, K. Benkouider, W.A. Nassan, O. Naqaweh, Multistability and Bifurcation Analysis of a Novel 3D Jerk System: Electronic Circuit Design, FPGA Implementation, and Image Cryptography Scheme. *IEEE Access*, 11 (2023): 78584-78600. DOI: 10.1109/ACCESS.2023.3299171.
- [5] V. Rusyn, M.A. Mohamed, D. Purwandari, M. Mamat, J. Titaley, B. Pinontoan, Chaotic and Controlling Regimes of a New Modified Chua's Generator. *Journal of Advanced Research in Dynamical and Control Systems*, 12(2) (2020): 556-561. DOI: 10.5373/JARDCS/V12I2/S20201077.
- [6] S. Vaidyanathan, I.M. Moroz, A. Sambas, A New Two-Scroll 4-D Hyperchaotic System with a Unique Saddle Point Equilibrium, Its Bifurcation Analysis, Circuit Design and a Control Application to Complete Synchronization. *Archives of Control Sciences*, 33 (2023): 277-298. DOI: 10.24425/acs.2023.146276.
- [7] S. Vaidyanathan, K. Benkouider, A. Sambas, Modelling, Bifurcation Analysis and Multistability of a New 4D Hyperchaotic System with No Balance Point, Its Master-Slave Synchronisation and Multisim Circuit Simulation. *International Journal of Modelling, Identification and Control*, 40 (2022): 279-293. DOI: 10.1504/IJMIC.2022.125551
- [8] M. Kopp, A. Kopp, A New 6D Chaotic Generator: Computer Modelling and Circuit Design. *Int. J. Eng. Technol. Innov.*, 12 (2022): 288-307. doi.org/10.46604/ijeti.2022.9601.
- [9] V. Rusyn, M. Mohamad, J. Titaley, N. Nainggolan, M. Mamat, Design, Computer Modelling, Analysis and Control of the New Chaotic Generator, *Journal of Advanced Research in Dynamical & Control Systems*, 12(2) (2020): 2306-2311. DOI: 10.5373/JARDCS/V12I2/S20201276.
- [10] S. Vaidyanathan, L.G. Dolvis, K. Jacques, C.H. Lien, A. Sambas, A New Five-Dimensional Four-Wing Hyperchaotic System with Hidden Attractor, Its Electronic Circuit Realisation and

- Synchronisation via Integral Sliding Mode Control. *International Journal of Modelling, Identification and Control*, 32 (2019): 30-45. doi.org/10.1504/IJMIC.2019.101959.
- [11] C.-H. Lien, S. Vaidyanathan, A. Sambas, Sukono, M. Mamat, W.S.M. Sanjaya, Subiyanto, A New Two-scroll Chaotic Attractor with Three Quadratic Nonlinearities, Its Adaptive Control and Circuit Design. *IOP Conference Series: Materials Science and Engineering*, 332 (2018), art.no. 012010. DOI 10.1088/1757-899X/332/1/012010.
- [12] V. Rusyn, A. Sambas, Mujiarto, Information security system based on chaotic signals. *CEUR Workshop Proceedings*, 3039 (2021): 294-299.
- [13] V. Rusyn, Mujiarto, M. Mamat, F. Azharul, W.S. Mada Sanjaya, A. Sambas, E. Dwipriyoko, A. Sutoni, Computer Modelling of the Information Properties of Hyper Chaotic Lorenz System and Its Application in Secure Communication System. *Journal of Physics: Conference Series*, 1764, (2021), art.no. 012205. DOI: 10.1088/1742-6596/1764/1/012205.
- [14] V. Rusyn, Modeling and Research Information Properties of Rucklidge Chaotic System using Labview, in: *10th International Conference on Chaotic Modeling and Simulation*, 2017, pp. 739-744.
- [15] V. Rusyn, S. Subbotin, A. Sambas, Analysis and experimental realization of the logistic map using Arduino Pro Mini. *CEUR Workshop Proceedings* 2608, (2020) 300-310.
- [16] O. Guillen-Fernandez, A. Melendez-Cano, E. Tielo-Cuautle, J.C. Nunez-Perez, J. de Jesus Rangel-Magdaleno, On the Synchronization Techniques of Chaotic Oscillations and Their FPGA-based Implementation for Secure Image Transmission. *PLoS ONE*, vol. 14(2) (2019), art. no. e0209618. doi: 10.1371/journal.pone.0209618.
- [17] V.-A. Adeyemi, J.-C. Nuñez-Perez, Y.S. Ibarra, F.-J. Perez-Pinal, E. Tlelo-Cuautle, FPGA Realization of the Parameter-Switching Method in the Chen Oscillator and Application in Image Transmission. *Symmetry*, 13 (2021): 923. doi.org/10.3390/sym13060923.
- [18] O. Guillén-Fernández, M.F. Moreno-López, E. Tlelo-Cuautle, Issues on Applying One-and Multi-step Numerical Methods to Chaotic Oscillators for FPGA Implementation. *Mathematics*, 9 (2021): 151. doi.org/10.3390/math9020151.
- [19] X. Peng, Y. Zeng, M. Wang, Z. Li, Generating Multi-layer Nested Chaotic Attractor and Its FPGA Implementation. *Chinese Physics B*, 30(6) (2021), art. no. 060509. doi.org/10.1088/1674-1056/abda34.
- [20] A. Sambas, S. Vaidyanathan, T. Bonny, S. Zhang, Sukono, Y. Hidayat, G. Gundara, M. Mamat, Mathematical Model and FPGA Realization of a Multi-Stable Chaotic Dynamical System with a Closed Butterfly-Like Curve of Equilibrium Points. *Appl. Sci.*, 11(2) (2021): 788. doi.org/10.3390/app11020788.
- [21] D. Enzeng, Y. Mingfeng, H. Fangfang, T. Jigang, D. Shengzhi, Topological Horseshoe Analysis and FPGA Implementation of a Classical Fractional Order Chaotic System. *IEEE Access*, 7 (2019): 129095-129103. doi.org/10.1109/ACCESS.2019.2938556.
- [22] V. Rusyn, S. Khrapko, Memristor: modeling and research of information properties, in: Ch. H. Skiadas, I. Lubashevsky (Eds.), *Springer Proceedings in Complexity*. Springer, Cham, 2019, pp. 229-238. DOI: 10.1007/978-3-030-15297-0_21.
- [23] L. Chua, Memristor-The Missing Circuit Element. *IEEE Transactions on Circuit Theory*, 18 (1971): 507-519. doi: 10.1109/TCT.1971.1083337.
- [24] Z.F. Wang, L. Shi, H. Wu, N. Helian, O.L. Chua, Fractional memristor. *Applied Physics Letter*, 111 (2017): art.no. 243502. doi.org/10.1063/1.5000919.
- [25] D. Yu, C. Zheng, H. H. C. Iu, T. Fernando, L.O. Chua, A new circuit for emulating memristors using inductive coupling. *IEEE Access*, 5 (2017): 1284-1295. doi.org/10.1109/ACCESS.2017.2649573.
- [26] R.M. May, Simple mathematical models with very complicated dynamics. *Nature*, 261 (1976): 459-467. doi.org/10.1038/261459a0.
- [27] M. Hénon, A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics*, 50 (1976): 69-77. doi:10.1007/BF01608556.
- [28] O.E. RöSSLer, An equation for continuous chaos. *Physics Letters A*, 57 (1976): 397-398. doi.org/10.1016/0375-9601(76)90101-8.
- [29] G.F.V. Amaral, C. Letellier, L.A. Aguirre, Piecewise affine models of chaotic attractors: The Rossler and Lorenz systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 16 (2006): 013115. doi: 10.1063/1.2149527.

- [30] A. Sambas, W.S. Mada Sanjaya, M. Mamat, Halimatussadiyah, Design and analysis bidirectional chaotic synchronization of Rossler circuit and its application for secure communication, *Applied Mathematical Sciences*, 7 (2013): 11-21. doi: 10.12988/ams.2013.13002.
- [31] V. Rusyn, O. Savko, Modeling of Chaotic Behavior in the Economic Model, in: 8th International Conference on Chaotic Modeling and Simulation, 2015, pp. 705-712.
- [32] S. Vaidyanathan, A.T. Azar, K. Rajagopal, A. Sambas, S. Kacar, Ü. Çavusoglu, A new hyperchaotic temperature fluctuations model, its circuit simulation, FPGA implementation and an application to image encryption. *IJSPM*, 13 (2018): 281-296. <http://dx.doi.org/10.1504/IJSPM.2018.093113>.
- [33] R. Voliansky, O. Statsenko, O. Sergienko, M. Zhelinskyi, O. Sinkevych, N. Volianska, Chua's Circuit with Nonlinear Energy Storages and Its Synchronization, in: 2023 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), 2023, doi:10.1109/UkrMiCo61577.2023.10380417.
- [34] A. Sambas, S. Vaidyanathan, M. Mamat, W.S. Mada Sanjaya, A six-term novel chaotic system with hidden attractor and its circuit design, *Studies in Systems, Decision and Control*, 133 (2018): 365-373. doi: 10.1007/978-3-319-71243-7_16.
- [35] R. Voliansky, A. Sadovoi, Chua's circuits interval synchronization, in: 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), 2017, doi:10.1109/INFOCOMMST.2017.8246434.
- [36] V. Rusyn, A. Samila, C. Skiadas, Computer Modeling and Practical Realization of Chaotic Circuit with a Light-Emitting Diode. *Proceedings of SPIE - The International Society for Optical Engineering*, 11369 (2020): art. no. 113690D. DOI: 10.1117/12.2550813.
- [37] R. Voliansky, O. Kluev, O. Sadovoi, O. Sinkevych, N. Volianska, Chaotic Time-variant Dynamical System, in: 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2020, doi:10.1109/TCSET49122.2020.235503.
- [38] R. Voliansky, A. Pranolo, A.P. Wibawa, Haviluddin, Transformation of 3-D Jerk Chaotic System into Parallel Form, in: 2018 International Symposium on Advanced Intelligent Informatics (SAIN), 2018, doi:10.1109/SAIN.2018.8673346.
- [39] V. Rusyn, M. Sadli, M. Mamat, Mujiarto, W.S.M. Sanjaya, Computer modelling of a new simple chaotic generator, *Journal of Physics: Conference Series*, 1477(2020): art.no. 022010. doi:10.1088/1742-6596/1477/2/022010.
- [40] M.I. Kopp, A.V. Tur, V.V. Yanovsky, Chaotic Dynamics of Magnetic Fields Generated by Thermomagnetic Instability in a Nonuniformly Rotating Electrically Conductive Fluid. *Journal of Physical Studies*, 27 (2023): art. no. 2403. DOI: 10.30970/JPS.27.2403.
- [41] S. Vaidyanathan, M. Feki, A. Sambas, C. H. Lien, A new biological snap oscillator: its modelling, analysis, simulations and circuit design. *International Journal of Simulation and Process Modelling*, 13(5), (2018) pp. 419-432. doi.org/10.1504/IJSPM.2018.10015882.
- [42] C.H. Skiadas, C. Skiadas, Chaotic Modelling and Simulation: Analysis of Chaotic Models, Attractors and Forms. In: Taylor & Francis Group, (2008), pp. 1-345.
- [43] C.H. Skiadas, Exact solutions of stochastic differential equations: Gompertz, generalized logistic and revised exponential. *Methodology and Computing in Applied Probability*, 12 (2010): 261-270.

Application of digital images processing for expanded gamut printing with effect of saving material resources

Bohdan Kovalskiyi ¹, Mariia Semeniv ^{1,2}, Nataliia Zanko ¹ and Vitalii Semeniv ¹

¹ Ukrainian Academy of Printing, PidHolosko Str. 19, Lviv, 79020, Ukraine

² Lviv Polytechnic National University, 12 S. Bandery str., Lviv, 79013, Ukraine

Abstract

This work concerns expanding the capabilities of modern technologies for high-quality color reproduction of digital images on material media. The research focuses on replacing spot colors with classic four-color printing, with the addition of two or three colored inks. The Fogra 55 characterization data was processed for multi-primary printing with additional colors Orange, Green, and Violet. New methods of pre-press preparation were applied, which are based on the processing of the colors of the original in the ICaS color space. As a result of the analysis of the projections of the color gamut, the formulated principle of color division of the image with two color inks and black ink was confirmed. According to the analytical solutions of the Neugebauer equations, no more than two colors and black are defined for the reproduction of spot colors. The study compared the color differences of color separation by classical and ICaS-ColorPrint technology. The quality of color reproduction was evaluated regarding the color differences ΔE (CIE 2000) in CIE L*a*b* color space for selected PANTONE+ Solid Coated library sample colors. The small values of color differences when converting color information using the ICaS-ColorPrint technology shows that there is an alternative to the expensive process of spot color printing.

Keywords

Multi-primary Printing, Expanded Gamut, spot colors, additional inks, ICC profile, analytical solutions, color separation, color synthesis, saving color inks

1. Introduction

One of the most important tasks of modern printing is high-quality printing of color images as close as possible to the original in terms of color reproduction. The basis of modern printing technologies is printing with four process CMYK (Cyan, Magenta, Yellow, Black) colors. In most cases, four-color printing meets the customer's requirements. However, this printing process has a significant drawback: its color gamut is limited compared to the gamut of the monitor or the gamut of real colors. In this case, you can improve the quality of color reproduction by introducing additional spot colors. The primary US spot color specification tool is the PANTONE Color Formula Guide. The main component of this system is a fan decks with 2,161 samples of inks applied to coated and uncoated paper [1].

Spot inks are used in printing process in order to print a specific color on products, for example, the brand color of the logo: Starbucks Green, the iconic Red of Coca Cola, Cadbury's Purple Reign, McDonald's the Yellow arch on a Red background, Pepsi the Red, Blue and Background Blue [2]. For the purpose of reproduction a high-art and advertising publications, two or three additional spot colors are used to emphasize the expressive features of the products. The printing process with spot colors will ensure a more accurate reproduction of solid colors such as: Orange, Green, Purple. For example, Orange colors will appear purer when Orange ink is added to the printing process, instead of obtaining the corresponding color by mixing Yellow and Magenta inks [3]. Due to the fact that one color can be obtained not at the expense of applying four layers of ink, but at the expense of only one layer, when printing with spot colors. It is

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ bkovalskiyi@ukr.net (B. Kovalskiyi); krykmary17@gmail.com (M. Semeniv); zankon@i.ua (N. Zanko); 1986ministr@gmail.com (V. Semeniv)

🆔 0000-0001-9088-1144 (B. Kovalskiyi); 0000-0003-0798-1916 (M. Semeniv); 0000-0002-4445-3576 (N. Zanko); 0000-0001-6764-1820 (V. Semeniv)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

possible to achieve an accurate reproduction of the brand color and a more dense and uniform cover of the printed area with ink, avoiding the graininess characteristic of process colors [4]. Spot colors are generally used to reach colors that are outside the gamut of process CMYK printing. However, printing with spot colors requires estimating ink quantities, use of a separate printing unit with requirements for cleanup and changeover for the next print job. The excess of spot color inks must be stored, creating inventory, or discarded in an environmentally sound manner. The whole process of spot color printing is inefficient – It is expensive and arduous and represents custom manufacturing, which is difficult to justify in a challenged economy [5].

2. Literature Review

Therefore, in recent years, the popularity of research in the field of multicolor printing has increased. These studies concern the possibility of replacing spot colors with classic four-color printing with the addition of two or three color inks. Such printing was named: Extended Gamut Printing [3], Expanded Gamut Printing [5], Multi-primary Printing [6].

Expanded Gamut (EG) printing requires a new approach in terms of color management software and the number of colorants. Dr. Abhay Sharma in his research explains what results to expect from this new approach [7, 8, 9]. EG printing extends the color gamut of the conventional printing process by using additional inks such as Orange, Green and Violet (OGV). According to research by the Toronto State Research University, the Expanded Gamut seven-color printing system provides reproduction of a larger range of colors and 90% of spot colors from the PANTONE Solid Coated library [8]. Advantages of EG printing include the ability to reproduce a large number of spot colors without spot color inks, better print accuracy, batch job capability, and fewer ink changes and washes. Barbara Braun-Metz, CEO of ColorLogic GmbH, considers that basically, an expanded gamut system can skip the extra work of printing with spot color inks. Instead, it can use a set of fixed expanded gamut inks to match the color the customer wants [5].

The relevance of multicolor printing is confirmed by a research project from the Fogra Institute called “Professional color communication in multi-primary printing” [6]. This project, which was carried out in 2019-2021, developed FOGRA55 characterization data and an ICC-profile for multi-primary printing, which also uses seven colors: Cyan, Magenta, Yellow, Black, Orange, Green, Violet (CMYKOGV) [6].

Use of additional colors complicates the process of color separation of images for the output of printing plates. Because it increases the variability and ambiguity of the reproduction of a particular color in the image. It's necessary during in the process of color separation of images to take into account the actual color characteristics of the inks and the technological conditions of the printing process as much as possible. This task is assigned to the ICC profiles. ICC profiles are used in the color management system, which are built on the characteristic data of experimental measurements. Output profiles explain how colors are represented in a specific printing method by establishing a link between the device and a standard color space like CIELAB or CIEXYZ. This connection is made using tables for mapping, which are then refined using interpolation, or through a series of parameters for transformations [10].

Equations proposed by physicist and researcher in image reproduction G. Neugebauer describe the principle of image synthesis on the imprint with colored inks. In his work [11], based on the International Commission on Illumination (CIE) XYZ color space, he formulated analytical equations for color synthesis in multicolor printing. The construction of the Neugebauer equations is based on the Demichel equations [12], which determine the relative areas of screen elements for all primary colors.

In conventional offset printing, the creation of different screen elements follows the principles of subtractive color mixing. This process involves working with three transparent ink layers: Yellow (Y), Magenta (M), and Cyan (C) [13]. When tiny color elements are combined together to form one complete color, it's known as spatial mixing. This process follows a law, which in vector color representation can be stated as: the resulting color vector from spatial mixing equals the sum of products of the color element vectors multiplied by the respective areas each color occupies. [14].

An illustration of the Neugebauer model begins by examining the simplest printing systems, which have two primary colors, F1 and F2. The printing area will be divided into four segments of primary colors (Figure 1): W (white), F1, F2, and F12.

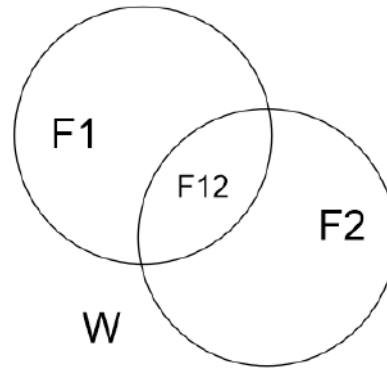


Figure 1: Schematic distribution of four colors W, F1, F2, F12 in a two-color printing system

The authors of the present paper have applied these equations for color conversion from the RGB model to CMYK through the orthogonal ICaS color space. It has been investigated the reproduction of colors on different paper types and determined that this process is non-linear. The non-linearity index is a parameter that takes into account the technological conditions of the printing process, such as the type of paper, the number of lines per inch, the type of the copy layer, the method of screening, the specifics of the printing process, and others. Thus made it possible to determine the optimal ink values for color reproduction by significantly saving color inks and develop the ICaS-ColorPrint information technology [15]. The overall objective is to perform analytical solutions [15, 16] and different image processing methods [17, 18] to evaluate spot color reproduction in multicolor printing.

3. Material and methods

3.1. Recalculating colors using orthogonal color space

Digital images consist of pixels that are defined by the amounts of Red, Green, and Blue light that make up their colors. On the other hand, printing devices use Cyan, Magenta, Yellow, and Black inks as their primary colors. There are different CMYK color spaces and gamuts that describe specific printing conditions.

Regardless of their source, all digital images are RGB color files. A color separation is needed to convert color information from RGB to CMYK or CMYKOGV. The authors generalized the characteristics of existing tangent color spaces to the additive and subtractive system to create a color space that will provide perfect conditions for recalculating colors [15]. The orthogonal condition of chromatic vectors results in a new color space. ICaS color space integrates two RGB and CMYK color representation systems on color charts. The transition to the ICaS color space is based on the Hartley orthogonal transformation [19, 20]. The formula for converting the coordinates R, G, and B into the coordinates of the space ICaS (I, C, S):

$$\begin{bmatrix} I \\ C \\ S \end{bmatrix} = \begin{bmatrix} 0.57735 & 0.57735 & 0.57735 \\ 0.57735 & 0.21132 & -0.78867 \\ 0.57735 & -0.78867 & 0.21132 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

The ICaS color space has a fundamental advantage as it uses three new coordinates to describe and quantify the colors of an image. These coordinates include an achromatic coordinate (I) and two chromatic coordinates (C) and (S). The achromatic coordinate (I) solely characterizes the neutral-gray colors of the original image. For any chosen color $F_i(R_i, G_i, B_i)$, the chromatic coordinates (C) and (S) on the plane, known as the chromatic CaS-color diagram, depict the color characteristics of the chosen color (as shown in Figure 2).

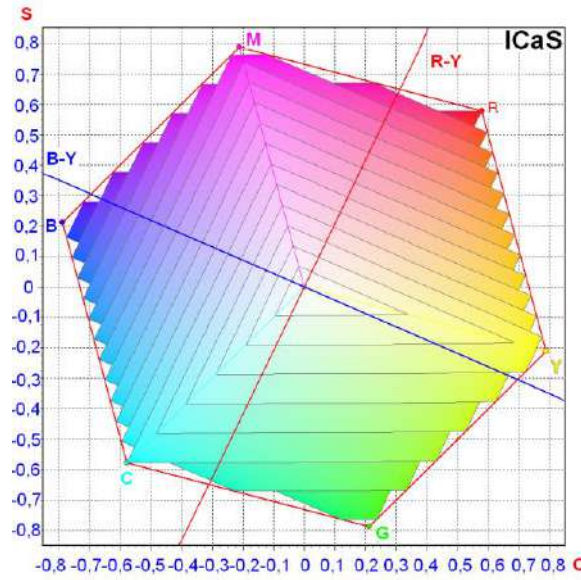


Figure 2: Planes of constant brightness of RGB colors on the CaS chromatic diagram

The placement of the color points on the CaS-color diagram and, accordingly, their coordinates in the ICaS color space will be analyzed for further research.

3.2. The method of determining the non-linearity of printing process

It was mentioned earlier that printing process is far-out from ideal one and has a certain amount of non-linearity, which distorts the reproduction of color. Therefore, it is important to get the parameter that will make it possible to compensate non-linearity on the stage of image preparation before printing.

In conventional offset printing, the different screen elements are formed strictly by the law of Subtractive Mixture. One works with the three transparent ink layers, which are Yellow (Y), Magenta (M), and Cyan (C) while maintaining the thickness of the ink layer. Tone values are simulated by screening. The size of screen dots can vary from zero (white paper) to 100% (full-tone ink layer). The index γ takes into account the various factors of influence on the amount of non-linearity of the particular color reproduction process. The dot gain mainly affects the non-linear color reproduction. In order to compensate for an increase in the size of the raster dots, a value of the dot gain quantity is performed on the value of the non-linearity index in the prepress stage.

The non-linearity of reproduction of Cyan ink gradations is determined by the color component R (Red). Similarly, the non-linearity of Magenta ink reproduction is determined by G (Green), and Yellow is determined by B (Blue). As for the additional inks OGV, we have analyzed that the nonlinearity should be calculated in the following way:

$$\begin{aligned} R &= R_G + (R_0 - R_G)(1 - S_G)^{\gamma_G}; & B &= B_{Or} + (B_0 - B_{Or})(1 - S_{Or})^{\gamma_{Or}}; \\ G &= G_V + (G_0 - G_V)(1 - S_V)^{\gamma_V} \end{aligned} \quad (2)$$

The nonlinearity parameter is determined by examining the curves depicting the relationship between the color value and the relative dot area of the printing ink. We opted for the Nonlinear Curve Fit Tool (NLFit) to assess the nonlinearity of the curves. The NLFit tool includes over 200 fitting functions that are utilized across various disciplines. Function Allometric2 is an extension of Classical Freundlich Model commonly employed to approximate growth curves and represented by the equation $y=a+bx^c$, where y is the dependent variable, x is the independent variable, and a , b , and c are model parameters [21]. The coefficient of determination (R^2) is utilized as a measure of how well the model accounts for the variability in the data. The values R^2 that are close to 1 indicates a better fit of the model to the data [22]. The coefficients of determination for the curves were computed, yielding an average value of 0.999.

According to this method, the value of the non-linearity γ parameter was determined for the standardized and production conditions for obtaining colored inks [15].

3.3. Equations for modeling the color gamut of inks

In CMY subtractive color model the basic process inks are selected as complementary colors to additive color system RGB. The theoretical model of multi-primary color mixing is described by the vector equation:

$$\mathbf{F}_{AU} = \mathbf{F}^{(0)} + \mathbf{F}^{(1)} + \mathbf{F}^{(2)} + \mathbf{F}^{(3)} + \dots + \mathbf{F}^{(N)} \quad (3)$$

$$\mathbf{F}^{(0)} = \mathbf{F}_W \prod_{i=1}^N (1 - S_i) = \sigma_{min} \mathbf{F}_W, \quad \mathbf{F}^{(N)} = \mathbf{F}_{1,2,\dots,N} \prod_{i=1}^N S_i = \sigma_{max} \mathbf{F}_{1,2,\dots,N} \quad (4)$$

where $\mathbf{F}_{AU}$ – vector of any color on the imprint in color space; $\mathbf{F}^{(0)}$ – vector that is proportional to the vector of paper's color $\mathbf{F}_W$; $\mathbf{F}^{(N)}$ – color vector, which is formed by mixing the N number of inks; $\mathbf{F}^{(1)}, \mathbf{F}^{(2)}, \mathbf{F}^{(3)}$ – color vector, which is formed by mixing the inks of 1, 2 and 3 accordingly; S_i – relative dot area of the i -th ink; $\sigma_{min}, \sigma_{max}$ – minimum and maximum dot area of mixing of inks; N – number of inks.

Based on the use of the color space ICaS, the Neugebauer equation for seven-color printing can be written down by the general vector equation:

$$\mathbf{F}_{AU} = \mathbf{F}^{(0)} + \sum_{n=1}^N \sigma_n \mathbf{F}_n^{(1)} + \sum_{n=1}^N \sum_{m=n+1}^N \sigma_n \sigma_m \mathbf{F}_{n,m}^{(2)} + \sum_{n=1}^N \sum_{m=n+1}^N \sum_{k=m+1}^N \sigma_n \sigma_m \sigma_k \mathbf{F}_{n,m,k}^{(3)} + \dots + \sigma_1 \sigma_2 \sigma_3 \dots \sigma_N \mathbf{F}_{1,2,3,\dots,N}^{(3)} \quad (5)$$

$\mathbf{F}^{(0)}, \mathbf{F}_n^{(1)}, \mathbf{F}_{n,m}^{(2)}, \mathbf{F}_{n,m,k}^{(3)}, \dots, \mathbf{F}_{1,2,3,\dots,N}^{(3)}$ – vectors of the binomial decomposition are written by the sum of the base vectors $\mathbf{F}_n$ of color inks and vectors $\mathbf{F}_{n,m}, \mathbf{F}_{n,m,k}, \dots, \mathbf{F}_{1,2,3,\dots,N}$ of all combinations of mutual superimposition of different inks. To solve equation (5) it is necessary to have numerical values of 2^N base vectors. This is the fundamental complexity of the solution of the Neugebauer equations, since the same color $\mathbf{F}_{AU}$ can be practically synthesized by different ratios of colored and black inks.

Modified Neugebauer equations in the color space ICaS have certain properties:

1) The index (γ) is a characteristic of the nonlinearity of the printing process, in which single-color images with sufficient accuracy are described by the linear equation:

$$\mathbf{F}_{ICaS}^{(1)} = \hat{\mathbf{H}} \mathbf{F}_i + \sigma_i \hat{\mathbf{H}} (\mathbf{F}_W - \mathbf{F}_i), \quad i = 1, 2, 3, \dots, N \quad (6)$$

where $\mathbf{F}_{ICaS}^{(1)}$ – vector of color in color space ICaS; $\hat{\mathbf{H}}$ – operator of the unitary transformation of Hartley; $\mathbf{F}_W$ – vector of color paper; $\mathbf{F}_i$ – vector of color of i -th ink; $\sigma_i = (1 - S_i)$, S_i – relative dot area of the i -th ink.

2) All the colors of the image which synthesized by black ink will be placed strictly on the achromatic coordinate (I) in the ICaS color space;

3) The chromatic coordinates of two colors are proportional to the coefficient equal to the scalar value σ_K , in case of characteristic by the same color tone and difference in achromatic proportions:

$$\mathbf{F}_{ICaS}^{(3)} = \sigma_K [\hat{\mathbf{H}} \mathbf{F}_{ij} + \sigma_i \hat{\mathbf{H}} (\mathbf{F}_j - \mathbf{F}_{ij}) + \sigma_j \hat{\mathbf{H}} (\mathbf{F}_i - \mathbf{F}_{ij}) + \sigma_i \sigma_j \hat{\mathbf{H}} (\mathbf{F}_W - \mathbf{F}_i - \mathbf{F}_j + \mathbf{F}_{ij})] \quad (7)$$

4) The general modified Neugebauer vector equation of color synthesis is reduced to a system of three nonlinear equations with three unknowns, which has analytical solutions in accordance with the formulated principle of color reproduction.

3.4. Test chart of spot colors

A color chart consisting of 22 PANTONE+ Solid Coated library color samples was created using Adobe Photoshop. Each sample has its CIE L*a*b* values that must be followed during the printing process. In the case of printing with spot colors, the accuracy of PANTONE color reproduction will be high. However, if spot color reproduction is done with CMYK or CMYKOGV process inks, it may not be possible to avoid color differences. The main objective of the study was to determine the ink values of the seven-color printing process required for reproducing the spot colors of the PANTONE Matching System. Additionally, the study aimed to calculate the color differences

between the original CIE L*a*b* values of spot colors and analytical solutions of the modified Neugebauer equations that are used for image separation in ICaS-ColorPrint information technology [15]. The selected colors were closely distributed in hue and chroma. In Figure 3 the selected colors and their respective hue values are shown.

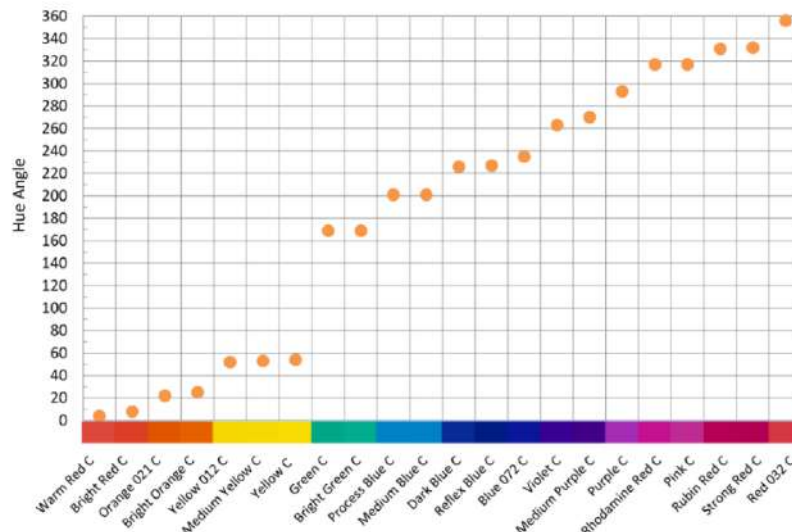


Figure 3: Hue angle distribution for 22 PANTONE+ Solid Coated library color samples

The image of test chart colors was converted to multichannel ICC profile FOGRA55-beta-TAC300-CL.icc based on FOGRA55 characterization data. As a result, we obtained seven channels CMYKOGV instead of the three RGB ones. Thus, we received new CIE L*a*b* values and relative dot areas of printing inks. We used the Color Picker panel and window Info in Adobe PhotoShop software to determine the required values.

The second stage of our research was processing a test chart with color samples in the ICaS-ColorSynthesis software [16]. First, it is necessary to enter values of the basic vectors (in color space) of the primary and secondary colors of extended gamut printing and the non-linearity indices. The method of determining the non-linearity of printing process was highlighted in the chapter 2.2. From Figure 4 it can be seen the user interface of the ICaS-ColorSynthesis software: data entry area (right panel) and modeling of the color gamut of inks (graph on the left).

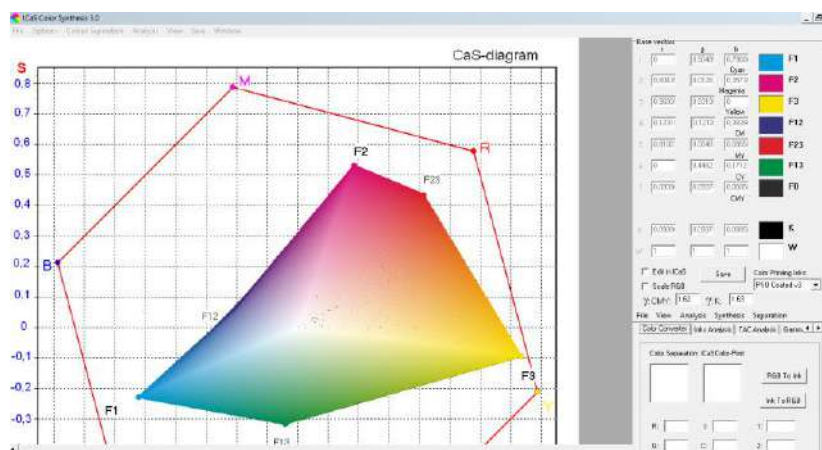


Figure 4: The ICaS-ColorSynthesis software for image color separation

The Color Picker tool is used to set the color. The Color Converter palette shows the results of color separation based on analytical solutions of the equations of color synthesis. As a result, we obtained the relative dot areas of process inks. The process of modeling color synthesis is also shown.

4. Results and Discussion

4.1. Modeling the color gamut of inks

The CIE L*a*b*, XYZ and RGB values were recalculated using chromatic adaptation, which takes into account the color of the paper and the light source. Fogra 55, the characterization data, contains the CIE L*a*b* colorimetric measurements of scale patches obtained under standardized printing conditions, namely, offset printing on coated paper with CMYKOGV inks.

Table 1

Input data for plotting curves and calculated numerical values of the non-linearity γ parameter

(1-S)	R _{CYAN}	G _{MAGENTA}	B _{YELLOW}	R _{GREEN}	G _{VIOLET}	B _{ORANGE}
1	0.89701	0.875329	0.869968	0.881894	0.872785	0.89147
0.902	0.753952	0.732769	0.73083	0.742872	0.733763	0.721304
0.8	0.624219	0.603766	0.603665	0.61091	0.6018	0.567003
0.749	0.564092	0.544304	0.545472	0.549996	0.540886	0.497691
0.6	0.403011	0.387562	0.389167	0.392947	0.383837	0.32403
0.502	0.308377	0.294554	0.29801	0.303161	0.294051	0.232559
0.251	0.112094	0.105386	0.10847	0.121898	0.112788	0.072472
0.2	0.080972	0.07636	0.078672	0.092362	0.083252	0.051508
0	0	0	0	0	0	0
non-linearity parameter	1.61923	1.65693	1.6332	2.07319	1.70881	1.70881

The non-linearity of the gradation reproduction of the printouts of CMYK inks was determined, which on average is 1.64 for CMY and 1.63 for Black. The non-linearity indices for Cyan are 1.62, Magenta is 1.66, Yellow is 1.63, Orange is 2.07, Green is 1.7, and Violet is 1.7 (as shown in Figure 5a, 5b). The values obtained indicate the increase in screen dots for printing inks. In order to compensate for this increasing, we need to consider the non-linearity index when preparing images for printing.

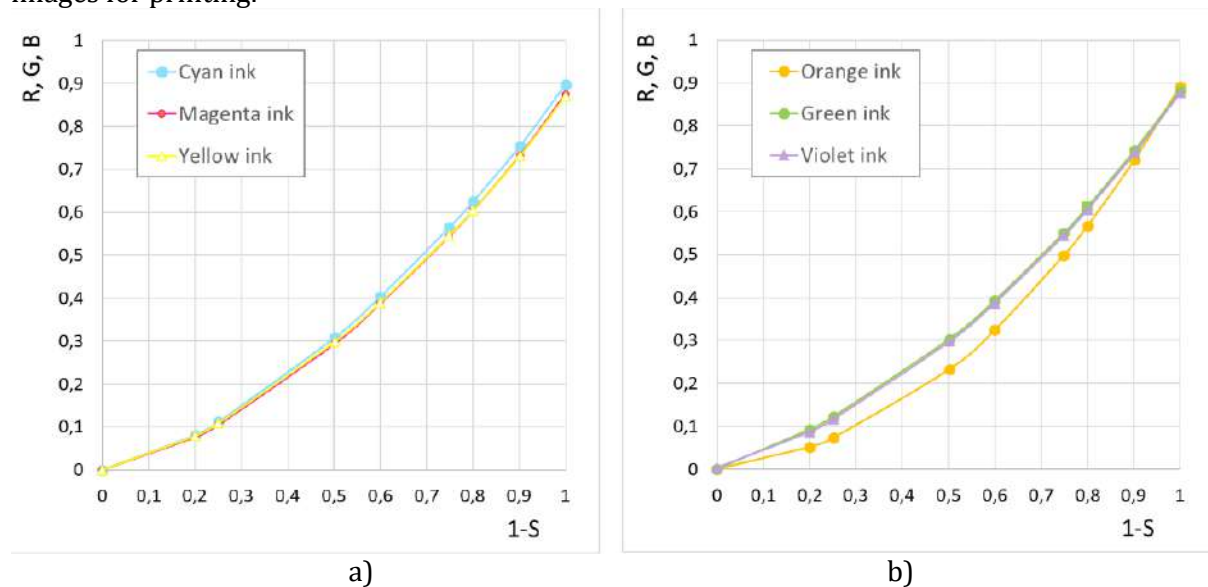


Figure 5: Non-linearity of each color inks CMY (a) and OGV (b) for the printing conditions Fogra 55

Based on the Fogra 55 colorimetric data and the calculated non-linearity indices, the coordinates of the basic color vectors of printing inks in the RGB space and, accordingly, in the ICaS color space were determined. To ensure that the previously formulated principle of reproducing an arbitrary color on the image with a maximum of two colored and black inks was followed, we modeled the color gamut of a seven-color printing process and compared it with

four-color printing process. When modeled the color gamut of the printing process (as shown in Figure 6b) we used modified Neugebauer equation (5, chapter 2.3), characterization data Fogra55 and non-linearity indices of printing inks.

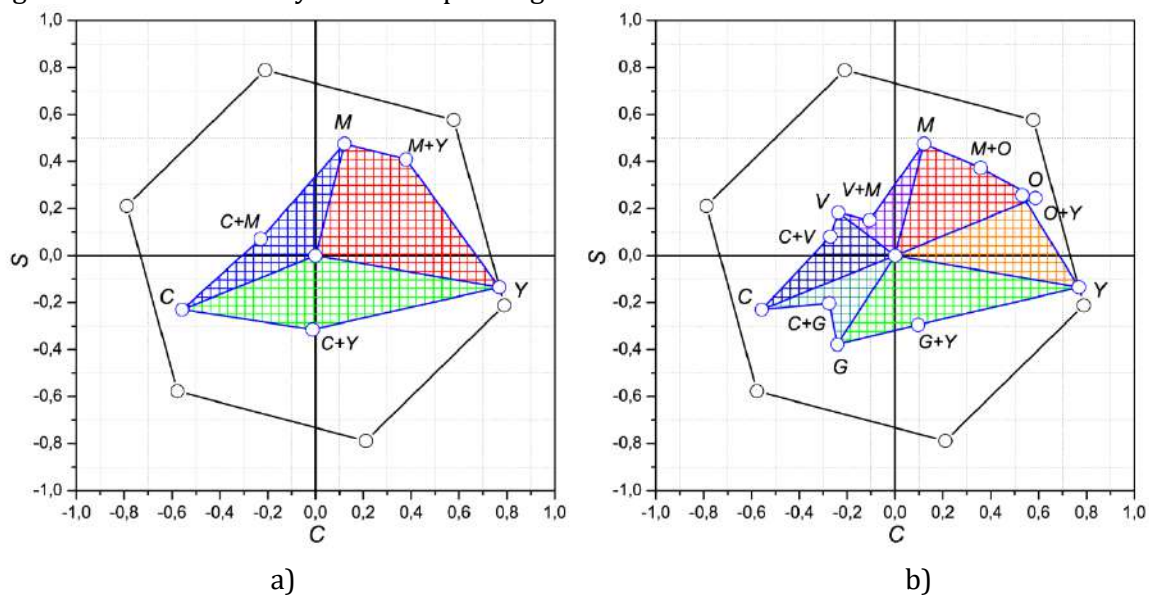


Figure 6: Model of color gamut of printing with four CMYK (a) and seven CMYKOGV (b) inks on the chromatic CaS-diagram

Based on the analysis of color gamut projections (as shown in Figure 6a, 6b), the color representation is reduced to six-color segments in the seven-color CMYKOGV printing process, depending on the hue. In the ICaS color space, the achromatic coordinate I of any arbitrary color in the digital image corresponds to the printout of the black (K) ink. Therefore, regardless of the number of N color printing inks, the ICaS color space completely separates the black (K) ink from the remaining N colored inks. This method offers the optimal color ink ratio for reproducing any color in the image. The use of additional colorants does not complicate the process of color separation, but only increases the number of areas of paired overlapping of inks.

As is well-known, RGB images displayed on a computer screen offer a wider range of colors compared to those printed using CMYK. To compare the color gamut of four and seven-color printing, we utilized a three-dimensional image, as projections of the color gamut onto a plane may not provide a comprehensive view. Figure 7: illustrates the comparison of color spaces between the original and printing processes. Gamut volume was determined using the online service Chromachecker [23]. Gamut volume quantifies the size of the color space. For AdobeRGB, the gamut volume is 1,325,000, for CMYK it is 405,912, and for CMYKOGV it is 570,877. The obtained values confirm that seven-color printing will enable the reproduction of more colors compared to four-color printing.

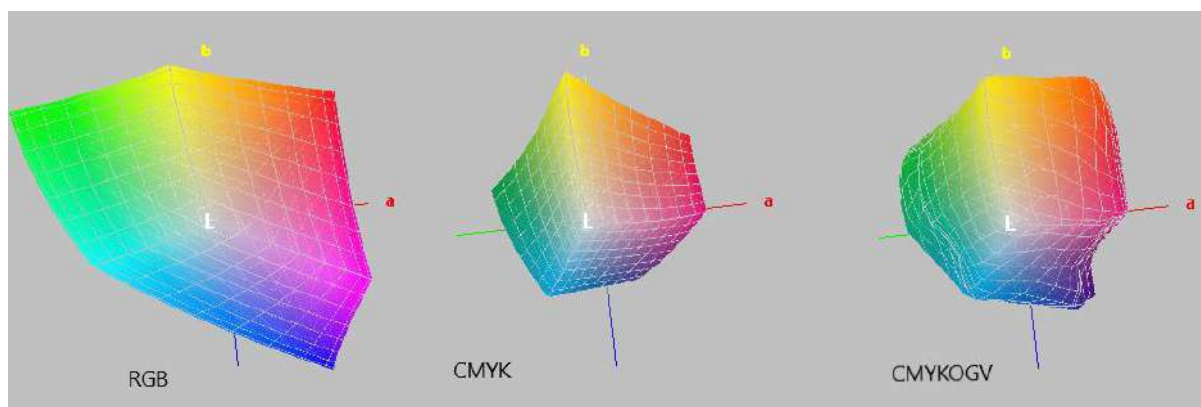


Figure 7: Three-dimensional images illustrate the color gamuts of RGB display colors, four-color CMYK, and seven-color CMYKOGV printing processes

4.2. Comparison of color differences

Thus, we have received the RGB values of spot color reproduction. Nevertheless, to calculate color differences, we needed CIE $L^*a^*b^*$ values. For this case, the CIE Color calculator was used [24]. The quality of color reproduction was evaluated in terms of the color differences ΔE (CIE 2000) in CIE $L^*a^*b^*$ color space for selected PANTONE+ Solid Coated library sample colors. Modern printing technologies ensure that the color of the print and the original match within $\Delta E = 1.5\text{--}6$ units CIE $L^*a^*b^*$. The color differences of spot colors given from ICC profile and calculate from ICaS-ColorPrint information technology based on analytical solutions. The value of $\Delta E > 6$ CIE $L^*a^*b^*$ corresponds to a significant noticeable difference between the two colors. Despite additional inks, color differences exceed acceptable limits. This means that it is not possible to achieve accurate reproduction the colors from the PANTONE library with Fogra55 conditions. There is a certain dependence between the color tone and the difference ΔE (CIE 2000). The largest value of color differences was obtained for samples of PANTONE colors of purple, violet and blue shades. The average value of color differences is 7.47 for ICC-profile and 6.13 for ICaS-ColorPrint information technology.

The Figure 8 shows the value of color differences of Pantone+ Solid Coated library color samples for which the reproduction process was simulated using the ICC profile and ICaS-ColorPrint information technology. In both cases, the Fogra55 characterization data for multi-primary printing were used. Obtained values of color differences from ICaS-ColorPrint of such color samples as: Bright Orange, Warm Red, Red 032, Rubine Red, Strong Red, Rhodamine Red and Process Blue, are smaller compared to the same colors from ICC profile. Among the 22 tested spot colors, only eight of them have smaller color differences of reproduction simulated using the ICC profile. As we can see, such color differences of spot colors is related to different ways of solving the problem of color synthesis, that is, with which inks and in what ratio.

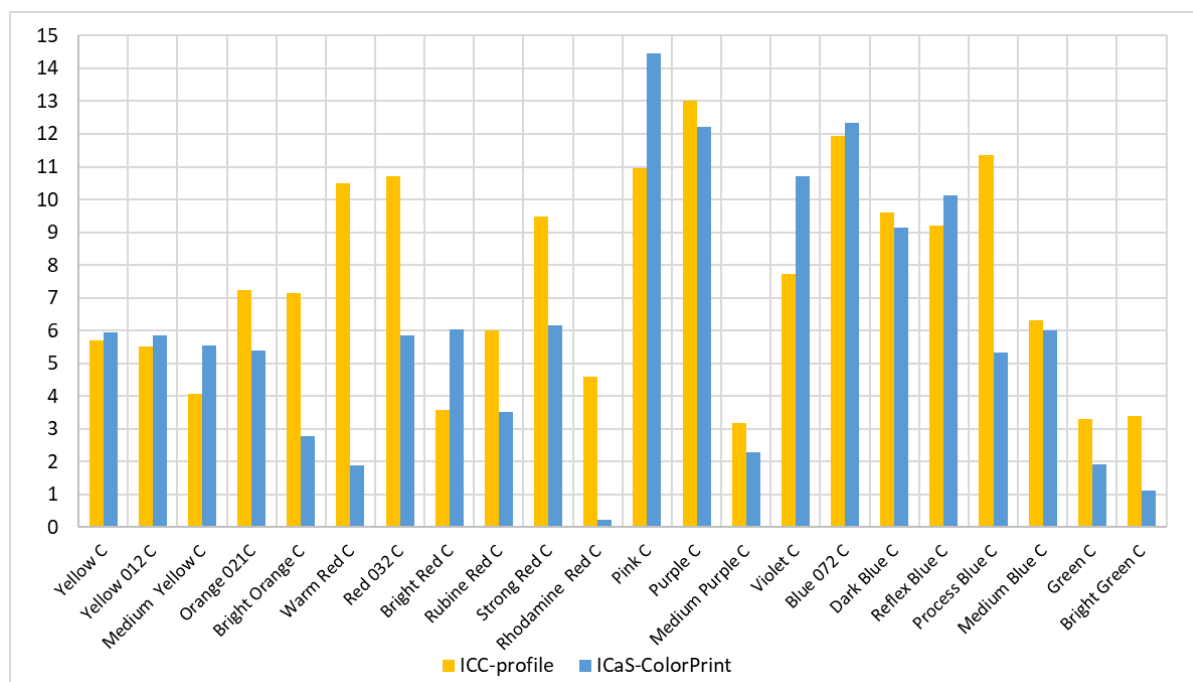


Figure 8: The color differences of the reproduction of spot colors when printing with an extended gamut using ICC profile and ICaS-ColorPrint information technology

4.3. Comparison of ink values

During the prepress stage of printing, it is crucial to consider the color reproduction characteristics of the printing process that will be used to print a digital image. To achieve this, a set of parameters must be established that takes into account the types of paper, inks, printing conditions, printing plates, and presses. These parameters are adjusted in the dialog box of standard programs that generate profiles. Initially, an algorithm must be chosen to replace a

certain amount of color inks with black. This is realized through the Gray Component Replacement (GCR) method, which replaces gray components with a specific percentage of black ink. GCR offers various options, including light, medium, heavy, and maximum. The selection of the Total Area Coverage (TAC) parameter depends on the type of paper being used. For offset printing, the TAC value ranges from 300-330%. Choosing a smaller TAC value is not recommended when using classic technology for color separation as it can lead to a decrease in color gamut during printing, resulting in lost contrast and reduced quality of the reproduction. However, lower TAC values can lead to a more stable printing process.

ICaS-ColorPrint is a color separation technology that replaces colored inks with black ink using the Maximum GCR algorithm. This means that for a given color, only two of the three color inks will be used along with black ink, depending on the color's position on the chromatic CaS-diagram. As a result, the TAC parameter will automatically be much lower than the permissible limits. We have determined the percentage of inks required to reproduce spot colors from the PANTONE+ Solid Coated library. With ICaS-ColorPrint technology, the reproduction of color is limited by its position on the CaS-diagram. This means that if a color coordinates between the base vectors of Magenta and Orange inks in the ICaS color space, the third ink used will always be black. The ICC-profile follows a different principle of color formation with inks, where the achromatic component of the color can be matched with a third color ink. However, not all CMYK colors are used in the profile, only those adjacent in color tone. There are color samples that use basic CMYK instead of additional colors for their synthesis. For example, consider the Bright Green C color. According to the ICC profile, 24% of Cyan, 16% of Yellow, and 93% of Green are required for its reproduction. However, using the analytical solutions of autotype synthesis equations, which are the basis of ICaS-ColorPrint technology, only 12% of Yellow, 100% of Green, and 14% of Black are needed to reproduce the same color.

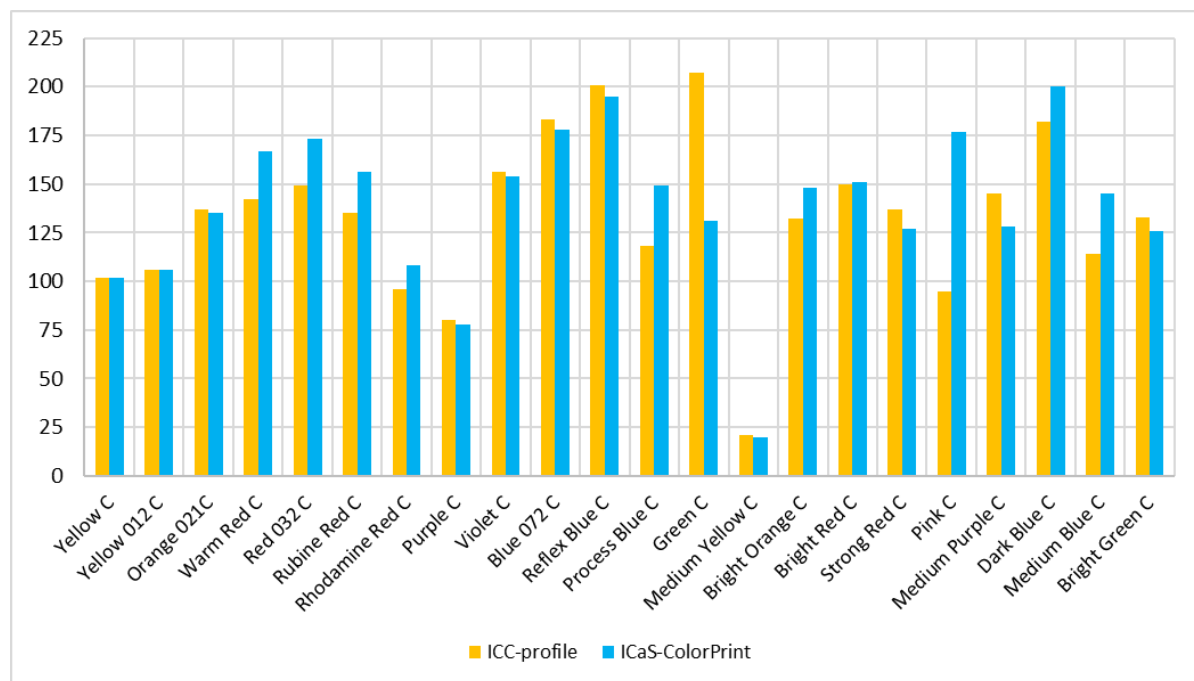


Figure 9: The total coverage of inks for spot colors reproduction in multi-primary printing

The study compared the dot areas of printing inks obtained from an ICC profile in Adobe PhotoShop with the values of inks obtained using the ICaS-ColorPrint technology (Figure 9). The analytical solutions of the equations reveal that spot colors can be reproduced using no more than two colored inks and black one. For instance, to reproduce Reflex Blue C Cyan, the ICC profile suggests using Cyan, Magenta, Yellow, Violet and Black inks. However, according to the ICaS-ColorPrint technology, only Violet and Black inks are required.

The average value required for total area coverage to reproduce the studied spot colors is 133% for the ICC profile based on Fogra55, and 139% for the ICaS-ColorPrint technology based on the analytical solutions. The maximum value of TAC is 207% and 200%, respectively. The

minimum value of TAC is 21% for an ICC-profile and 20% for an analytical model. We have observed that the value of inks falls in the range of 1-3%, which makes the process of manufacturing printing plates and the printing process more complicated.

ICaS-ColorPrint is a technology that uses an optimal ink ratio to produce any color in an image. This is achieved by solving modified Neugebauer equations which take into account printing parameters such as basic colors, their combination, and the non-linearities of the printing process. The addition of extra colors in the printing process simplifies the color separation process and optimizes it, as evidenced by the results.

Previous research has demonstrated that four-color printing using ICaS-ColorPrint technology can save a significant amount of ink compared to traditional methods. In this study optimal values of colors are also obtained. The study confirmed that using ICaS-ColorPrint technology leads to smaller color differences. This is due to the fact that we calculated the non-linearity of additional color printouts, which we took into account when determining the base color vectors of colors

5. Conclusions

A study of the possibility of replacing spot colors with classic four-color printing with the addition of two or three color inks was conducted. For this, the latest characterization data of Fogra 55 were used. The Fogra 55 characterization data has been processed for multi-primary printing with three additional colors: Orange, Green, and Violet. Introducing this into the classic printing process provides an expanded color gamut and the reproduction of a more significant number of colors from the PANTONE color library. It is shown that the ICaS color space provides optimal conditions for converting the color information of a digital original from the RGB model to the CMYK ink model. The use of modified Neugebauer equations for color conversion from RGB to CMYK through the orthogonal color space ICaS is substantiated.

The basic color vectors of pure inks in vector space and their double blends are defined. The indices of non-linearity of prints of additional inks have been determined. The non-linearity indices for Green and Violet are 1.8, and for Orange it is 2.07. The measured values indicate an increase in screen dots for printing inks. To compensate for this, must be considered the non-linearity index when preparing images for printing. We performed the analysis of the projections of the color gamut and confirmed the formulated principle of color separation of the image with two colored inks and black one. Depending on the hue, the color representation is reduced to six color segments in the seven-color CMYKOGV printing process (plus Orange, Green, and Violet).

The test chart image of 22 spot colors was generated for the investigation. The ICaS-ColorSynthesis software was used to separate colors using ICC profiles and analytical solutions of equations. This led to new CIE $L^*a^*b^*$ values for color samples, which could then be compared to the original values. The color differences obtained from ICaS-ColorSynthesis for colors such as Bright Orange, Warm Red, Red 032, Rubine Red, Strong Red, Rhodamine Red, and Process Blue were smaller than those obtained from ICC profile. The relative dot areas of the printing inks obtained from the ICC profile were analyzed and compared with the corresponding values obtained by solving the modified Neugebauer equations. According to the analytical solutions of the equations (ICaS-ColorPrint technology), it was found that no more than two colored inks and black one are defined for the reproduction of spot colors. The study confirmed that using the ICaS-ColorPrint technology for spot color reproduction will provide smaller values of color differences.

Thus, increasing the color gamut of the printing process through the addition of extra colors does not complicate the process of color separation of the image, but rather improves the accuracy of color reproduction, including spot colors. The color separation technology developed by the authors solves analytical solutions of the modified Neugebauer equations in a new orthogonal color space. This ICaS-ColorPrint technology takes into account new methods of determining the nonlinearity of the printing process, resulting in accurate color reproduction and saving of colored inks. The obtained solutions were compared with FOGRA experimental data from printing, produced under standardized conditions. A matching accuracy to experimental data ranging from 0.35 to 8.6 (ΔE), affirming the reliability of the developed ICaS-ColorPrint technology. A promising direction is the application of developed separation technology for digital and flexographic printing. In digital printing, ink is supplied in a controlled manner, making efficient use of color inks crucial. In flexographic printing for labeling and packaging,

substituting costly spot inks with triad inks and a few additional ones can conserve resources without compromising color reproduction quality.

References

- [1] Pantone Formula Guide, 2024. URL: <https://www.pantone.com/formula-guide-coateduncoated>.
- [2] Color Branding & Trademark Rights, 2024. URL: <https://www.colormatters.com/color-and-marketing/color-branding-legal-rights>.
- [3] What Is Extended Gamut Printing? 2023. URL: <https://www.oliverinc.com/blog/what-is-extended-gamut-printing>.
- [4] What Are Pantone Color Systems? URL: <https://www.pantone.com/color-systems/pantone-color-systems-explained>
- [5] A. Sharma, Expanded gamut study evaluates spot color reproduction in multicolor printing, 2020. URL: <https://www.flexography.org/industry-news/expanded-gamut-study-evaluates-spot-color-reproduction-multicolor-printing>.
- [6] Professional colour communication in multi-primary printing, 2021. URL: <https://fogra.org/en/research/prepress-technology/multiprimary-printing-13003>
- [7] A. Sharma, Understanding Color Management, 2nd. ed., Wiley, New York, NY, 2018. doi:10.1002/9781119223702.ch9
- [8] A. Sharma, Expanded Gamut Shoot-Out: Real Systems, Real Results, 2020. URL: <https://color.printing.org/wp-content/uploads/2020/01/Sharma-Expanded-Gamut-Shoot-Out.pdf>
- [9] A. Sharma, J. Seymour, Evaluation of expanded gamut software solutions for spot color reproduction, volume 45 of Color. Research and Application, 2020, pp. 315–324. doi:10.1002/col.22471
- [10] N. Fdhal, M. Kyan, D. Androutsos, A. Sharma, Color space transformation from RGB to CIELAB using neural networks, in: P. Muneesawang, F. Wu, I. Kumazawa, A. Roeksabutr, M. Liao, X. Tang (Eds.), Advances in Multimedia Information Processing, volume 5879 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2009, pp. 1011–1017. doi:10.1007/978-3-642-10467-1_97.
- [11] H. Neugebauer, Die teoretischen Grundlagen des Mehrfarbendrucks [The theoretical basics of multicolor printing], Zeitschrift für wissenschaftliche Photographie, Photophysik und Photochemie, volume 36, (1937), 36–73.
- [12] E. M. Demichel, Le procédé 26 (1924), 17–21, 26–27.
- [13] Küppers' Theory of Color. URL: http://www.kuepperscolor.de/en/farbe_und_druckverfahren/mehrfarbendruck_und_druckbedingungen.html
- [14] V. Ostromoukhov, Chromaticity gamut enhancement by heptatone multi-colour printing. SPIE (1993; 1909) 139–151.
- [15] M. Shovheniuk, B. Kovalskiy, M. Semeniv, V. Semeniv, N. Zanko, Information technology of digital images processing with saving of material resources, in: Proceedings of the 15th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, volume I: Main Conference, Kherson, Ukraine, 2019, pp. 414–419. URL: <http://ceur-ws.org/Vol-2387/20190414.pdf>
- [16] B. Kovalskiy, M. Semeniv, M. Shovheniuk, Kompiuterna prohrama syntezy zobrazhennia na vidbytku dlia novoi informatsiinoi ta tradytsiinykh tekhnolohii kolorovoho druku [A computer program for the synthesis of an image on a print for new information and traditional color printing technologies], Science and Education a New Dimension, Natural and Technical Sciences 91 (2016) 72–78.
- [17] Kh. Kulchytska, M. Semeniv, B. Kovalskiy, N. Pysanchyn, Z. Selmenska, Influence of Hadamard Matrices Canonicity on Image, in: Z. Hu, S. Petoukhov, F. Yanovsky, M. He (Eds.), Advances in Computer Science for Engineering and Manufacturing, volume 463 of *Lecture Notes in Networks and Systems*, Springer, Cham, 2022. doi:10.1007/978-3-031-03877-8_29.
- [18] B. Durnyak, O. Tymchenko, O. Tymchenko, B. Havrysh, Applying the Neuronetchic Methodology to Text Images for Their Recognition, in: Proceedings of the 2nd International

- Conference on Data Stream Mining & Processing, DSMP'2018, pp. 584–589. doi:10.1109/DSMP.2018.8478482.
- [19] N. Patra, S. S. Nayak, Discrete Hartley transform and its applications - a review, volume 10 of J. Integr. Sci. Technol, 2022, pp. 173–179.
- [20] M. Ma et al., High-efficiency single-pixel imaging using discrete Hartley transform, vol. 11 of AIP Adv, 2021. doi:10.1063/5.0057119.
- [21] Allometric2. URL: <https://www.originlab.com/doc/Origin-Help/Allometric2-FitFunc#Function>
- [22] P. Boyce, Coefficient of Determination: Definition, Calculation & Examples, Econometrics, 2023. URL: <https://boycewire.com/coefficient-of-determination/>
- [23] Gamut volume evaluation URL: https://chromachecker.com/manuals/en/show/gamut_volume
- [24] Color Difference Calculator. URL: <http://www.brucelindbloom.com/>

Mathematical Modeling of COVID-19 Pandemic and its Impact on Economy

Konstantin Atoyev^{1,2}, Pavel Knopov^{1,2}

¹ V.M. Glushkov Institute of Cybernetics of NAS of Ukraine, 40 Glushkov ave., Kyiv, 03187, Ukraine

² State Institution "Center of Evaluation of Activity of Research Institutions and Scientific Supports of Regional Development of Ukraine NAS of Ukraine", 54 Volodymyrska str., Kyiv, 01601, Ukraine

Abstract

A model for the spread of the COVID-19 pandemic and its induced changes in individual sectors of the economy has been developed. An approach to studying the interconnection of the food production system, food transportation, and the socio-economic sphere during the pandemic conditions using a three-sector Lorenz model has been proposed. Research has been conducted on the impact of the pandemic on changes in the supply-demand balance in these sectors of the economy. A model of the T-cell immune response has been developed, considering the peculiarities of its course in COVID-19, when excessive inflammation leads to cytokine release syndrome (cytokine storm), causing damage to vital organs and disruption of the immune system. The model considers the influence of cellular energetics on the regulation of the immune response and the amplification of inflammation.

Keywords

Mathematical modeling, COVID-19, economic processes

1. Introduction

In the 2021 Davos Forum report [1], it was noted that infectious diseases rank first among the highest risks influencing the next decade. On April 7, 2024, a total of 774,699,366 people was infected worldwide, and 7,033,430 people died [2]. The global changes of the past decades have so closely intertwined various spheres of social organization that the global COVID-19 pandemic has significantly impacted many processes in the world economy. This has initiated a chain of changes leading to disruptions in socio-economic connections at the local, regional, and global levels [3].

In an effort to control the epidemic process, governments have implemented lockdowns and other restrictions on economic activity, which have closed many businesses, limited domestic travel, closed borders to the movement of labor and certain food products, and introduced requirements for social distancing and curfews. Anti-epidemic measures are becoming a heavy burden on the economy as they begin to affect a complex set of ecological, economic, political, and social processes, which have enormous consequences for the lives of individuals, societal well-being, economic activity, and food security. In formulating effective strategies to normalize the economic situation amidst the pandemic, many governments are faced with the difficult problem of how to provide support to the economy in the short term while avoiding unwanted inflationary consequences and risks that could threaten financial stability in the medium term.

The uniqueness of the COVID-19 virus is associated with the lack of a single viewpoint on the nature of its origin and dissemination pathways. This leads to a situation of uncertainty that requires special mathematical methods used to investigate improbable or unique events, where the stochastic nature of the object under study, or the incompleteness of the sample, reduces the effectiveness of traditional statistical methods. Therefore, the task of developing models that consider the impact of random disturbances becomes particularly relevant in forecasting the development of the pandemic, as these disturbances become additional factors increasing the level of structural disruptions in the epidemic system.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ konstantin_atoyev@yahoo.com (K. Atoyev); knopov1@yahoo.com (P. Knopov)

ORCID 0000-0001-9892-054X (K. Atoyev); 0000-0001-6550-2237 (P. Knopov)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

At the same time, it becomes clear that forecasting the development of pandemics is impossible without the development of mathematical models to study the negative impact of global epidemics on economic development. These models should consider both epidemic and economic factors and allow the determination of effective strategies to minimize the number of casualties and economic losses from the epidemic under various scenarios of its development. They should also enable a systematic analysis of responses to challenges for sustainable development in the face of new bio-threats. The aim of this work is to develop precisely such mathematical models for studying the negative impact of global pandemics on economic development and to conduct a system analysis of responses to challenges for sustainable development in the face of new bio-threats.

2. Modeling the impact of COVID-19 on economy

In [4], an approach to studying the interconnection of water, food, energy resources, food transportation, and medical consequences of a pandemic was proposed using a multisectoral Lorenz model. This model unifies sectors of the economy that are similarly described in a single structure, with each sector considered in terms of productivity levels, the number of jobs, and the level of structural disruptions. Through modeling, conditions for the emergence of deterministic chaos were identified, and trajectories of changes in socio-economic factors were calculated, allowing for the reduction of the number of structural disruptions. This reduction is achieved by altering the balance between supply and demand in creating jobs and production in relevant sectors of the economy. The study of the model allowed tracing how changes in the balance of supply and demand in interconnected systems lead to the emergence of a chaotic stable attractor.

The SIR epidemic model used frequently to study the spread of COVID-19. It describes the interaction of three population groups: the healthy (S), the infected (I), and those who recovered (R) [5]. To investigate the interrelationship between changes in the economy and the epidemic process, these two approaches are combined. This allows connecting the SIR model with parameters whose dynamics are determined using a model of interconnections in the food production, its transportation, and the socio-economic sphere. It is assumed that the level of disruptions is proportional to the level of disruptions in the operation of production systems in different sectors of the economy.

2.1. Mathematical model of COVID-19 spread

Let's consider three manufacturing systems (MS) related to food production, its transportation, and medical sector. We will use the Lorenz model to study interrelated sectors of the economy [4]. Let (X_i) , (Y_i) and (Z_i) be normalized levels of productivity, employment quantity, and level of structural disruptions respectively for the food production system ($i = 1$), food transportation ($i = 2$), and the socio-economic sphere (medical infrastructure) ($i = 3$). Let's assume that different sectors of the economy compete with each other for labor, while changes in climatic conditions, financial, and political instability introduce random disturbances $w_{ij}(t)$, becoming additional factors increasing the level of structural disruptions in the socioeconomic system [4]. Since processes in different sectors of the economy proceed at different rates, we will scale time in them by introducing parameters ε_i . The model has the following form:

$$\begin{aligned} \varepsilon_i \frac{dX_i}{dt} &= \sigma_i(Y_i - X_i) + \delta_i \dot{w}_{ij}, & \varepsilon_i \frac{dY_i}{dt} &= [r_i(X_1, X_2, X_3) - Z_i]X_i - Y_i + \delta_i \dot{w}_{ij}, \\ \varepsilon_i \frac{dZ_i}{dt} &= X_i Y_i - b_i Z_i + \delta_i \dot{w}_{ij}, \end{aligned} \quad (1)$$

where $r_i = r_{i0}(1 - \sum a_{ik} X_i)$; δ_i - perturbation intensities; σ_i, r_i, b_i - parameters of Lorenz model; a_{ik} - parameters that characterize competition in labor markets ($i \neq k$); $w_{ij}(t)$ - independent standard Wiener processes with parameters $E(w_{ij}(t) - w_{ij}(s)) = 0$, $E(w_{ij}(t) - w_{ij}(s))^2 = |t - s|$ [4].

Research on the canonical Lorenz model shows that increasing the parameters r_i leads to the emergence of turbulence in the model (1). Since in model (1) r_i are functions of the variables X_i , in fact we have a Lorenz model with variable parameters r_i . Threats to food resources, according to [1], are considered long-term, while threats to disruptions in food transportation and the socio-medical sphere are considered medium-term. Therefore, the following parameter values are chosen: $\varepsilon_1 = 1$, $\varepsilon_2 = \varepsilon_3 = 2.5$. The coefficients a_{ik} are selected considering the weight of threat factors. Their values are shown in Table 1.

Table 1
Ranking of threats factors

Threats	Rank of influence	Value of parameters
Crisis of food resources	43.9	$a_{12} = 0.012, a_{13} = 0.013$
Disruption of food supply chains	38.3	$a_{21} = a_{23} = 0.013$
Socio-economic risks	43.4	$a_{31} = 0.013, a_{32} = 0.012$

The results obtained in [4] allow us to relate the parameters σ_i, r_i, b_i with the characteristics of the sectors of the economy in the following way:

$$\sigma_i = (\alpha_{1i}\beta_{2i})/(\alpha_{2i}\gamma_{2i}), \quad r_i = (\beta_{1i}\gamma_{1i})/(\beta_{2i}\gamma_{2i}), \quad b_i = \zeta_i/(\alpha_{2i}\gamma_{2i}), \quad (2)$$

where α_{1i} and α_{2i} are parameters characterizing adaptive capabilities; β_{1i} are demands for the activity of the i -th MS, normalized per unit of the material production system, considering the workplace in the corresponding industry of production Y_i ; β_{2i} are supplies, normalized per unit of function of the i -th MS X_i ; γ_{1i} are demands for an increase in number of jobs, normalized per unit of X_i ; γ_{2i} are supplies of jobs involved in providing X_i , normalized per unit of Y_i ; ζ_i are the specific rate of growth in the number of disruptions. For the study of the epidemic process, we will use the modified SIR model, which has the following form:

$$\begin{aligned} \frac{dy_1}{dt} &= -\frac{R(Z_3)y_1y_2}{NT_{inf}} + \alpha(t)y_3 + \beta_1\dot{w}_1, & \frac{dy_2}{dt} &= \frac{R(Z_3)y_1y_2}{NT_{inf}} - \frac{U_1(X_3)y_2}{NT_m} + \beta_2\dot{w}_2, \\ \frac{dy_3}{dt} &= (1 - a_0)\frac{U_1(X_3)y_2}{NT_m} - \alpha(t)y_3 + \beta_3\dot{w}_3, & \frac{dy_4}{dt} &= a_0\frac{U_1(X_3)y_2}{NT_m} + \beta_4\dot{w}_3, \end{aligned} \quad (3)$$

where y_1 - susceptible individuals, y_2 - infected patients, y_3 - recovered patients, y_4 - deceased patients; β_k ($k=1,4$) - disruption intensity parameters; $w_k(t)$ has the same meaning as it had in model (1) and corresponds to the same conditions; a_0 - the fraction of infected individuals who died; T_{inf} - the active period during which the infected individual is contagious; T_m - average recovery period of an infected individual; N - total population. We will consider that the effectiveness of the medical system's operation is determined by the level of disruptions within it Z_3 , the dynamics of which is calculated using model (1), and the reproduction rate (average number of infections caused by one infected individual) depends on the effectiveness of the medical system's operation and is determined by the function $R(Z_3)$. We will also assume that the recovery rate depends on the productivity level of the medical system, which is determined by the function $U_1(X_3)$. The dynamics of variable X_3 is also calculated using model (1). Additionally, we will consider that immunity to the virus decreases over time for those who have recovered from COVID-19, i.e., a person can get sick again. The rate of this process is described by function $\alpha(t)$ in model (4). The negative impact of the epidemic on economic processes will be investigated by replacing the parameters r_i in (2) with function $r_i U_{2i}(y_2)$. An increase in the number of infected individuals will increase the parameters r_{10}, r_{20} and r_{30} in (2) and when they reach bifurcation values, this will lead to the emergence of stochastic regimes.

2.2. Investigation of the epidemic dynamics

Let us consider the initial wave of the epidemic, when there are no individuals with immunity to the virus. Let $\alpha(t) = 0$, and let the functions $R(Z_3)$, $U_1(X_3)$ and $U_2(y_2)$ have the following form:

$$R(Z_3) = R(1 + Z_3/k_1),$$

$$U_1(X_3) = 1 + k_2 X_3^2 / (k_3 + X_3^2), \quad U_{2i}(y_2) = 1 + l_i y_2 / N, \quad (4)$$

where l_i and k_i are parameters of the model ($i = \overline{1,3}$).

In figures 1-2, the results of simulating the initial wave of the epidemic are shown for various initial numbers of infected individuals and varying recovery rates k_2 .

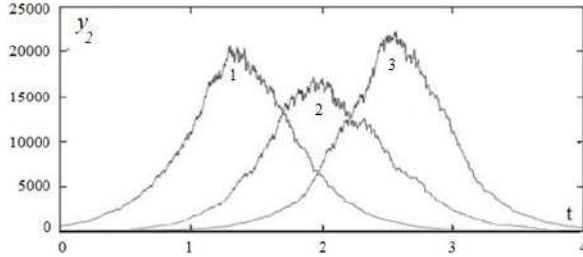


Figure 1: Dynamics of the number of infected individuals y_2 under different initial conditions: 1- $y_2(0) = 500$; 2- $y_2(0) = 50$; 3- $y_2(0) = 5$.

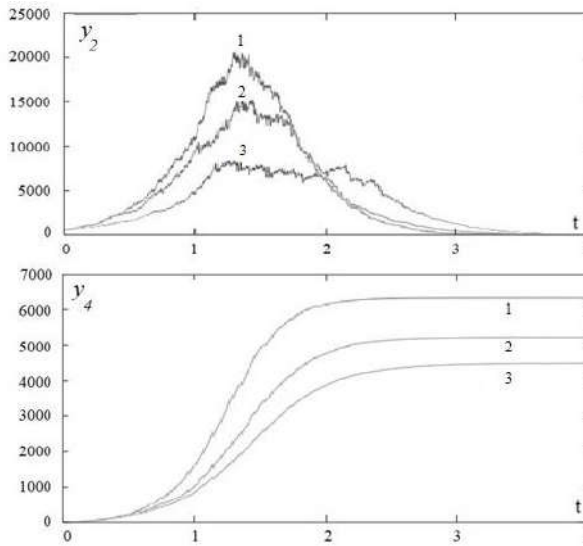


Figure 2: Dynamics of the number of y_2 and deceased individuals y_4 with variation in k_2 : 1- $k_2 = 27$; 2- $k_2 = 30$; 3- $k_2 = 34.5$

Figure 3 displays the results of modeling when a portion of the sick who died (a_0) is a function of the level of disruptions in the medical sector (Z_3) and is defined as follows:

$$a_0(Z_3) = a_0[1 + k_4 Z_3 / (k_5 + Z_3)].$$

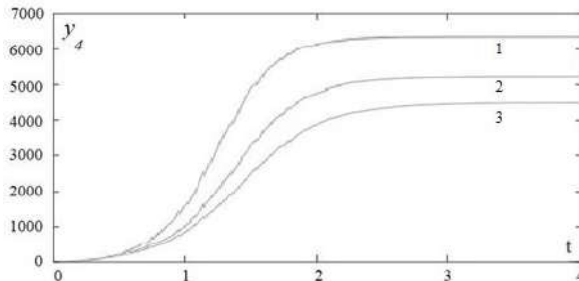


Figure 3: Dependency of the dynamics of deceased individuals y_4 on the number of disruptions Z_3 : 1- $k_4 = 0$; 2- $k_4 = 0.01$; 3- $k_4 = 0.02$; 4- $k_4 = 0.025$.

y_i , in all figures are measured by the number of people, Z_i are measured by the number of disruptions, time is measured in months.

In [4], the investigation focused on how changes in the parameter r affect the formation of system (1) regimes and the emergence of a stable strange attractor. Let's examine how the

emergence of a pandemic can alter the functioning of interrelated sectors of the economy from deterministic to chaotic regimes. The of the Lorenz model shows that increasing the parameter r is crucial for the onset of turbulence. The operating modes in model (1) change at the following bifurcation values of r : $r = 13.926$, $r = 24.06$, $r = 24.74$. When $r > 24.74$, the system transitions to a regime of metastable chaos - a strange attractor emerges [6]. For modeling purposes, the value of parameter $r = 16$ was chosen. The increase in the variable y_2 will raises r and lead to a change in the operating mode.

In figure 4, it is shown how the increase in the parameter l_3 , which characterizes the level of impact of the pandemic on the functioning of the medical sector, contributes to the emergence of stochastic regimes (time measured in months). In the absence of such influence ($l_3 = 0$), damped oscillations of the variable Z_3 are observed.

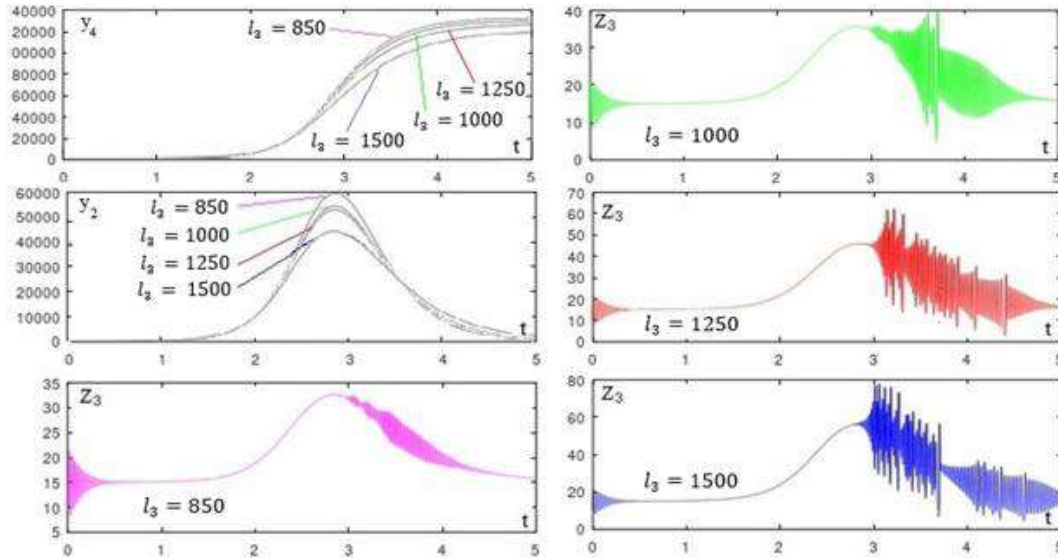


Figure 4: Modeling the impact of pandemic on structured disruptions of medical sector.

At $l_3 = 100$, the variable Z_3 reaches a steady level. Then it only deviates from it, depending on the changes in y_2 . At $l_3 = 850$ and $l_3 = 1000$, cyclic oscillations begin, which dampen when the level of infected individuals decreases to level that was before the beginning of the epidemic. At $l_3 = 1250$ and $l_3 = 1500$, stable chaotic oscillations arise.

In figure 5 the results of modeling changes in the levels of disruptions in the food production system Z_1 , food transportation Z_2 , and medical infrastructure Z_3 during the epidemic are shown. During the simulation, the total population was chosen to be 34 million people. With increasing values of the parameters r_i , the number of infected individuals also increases. Periodic trajectories transition to chaotic ones upon reaching the bifurcation value of this parameter.

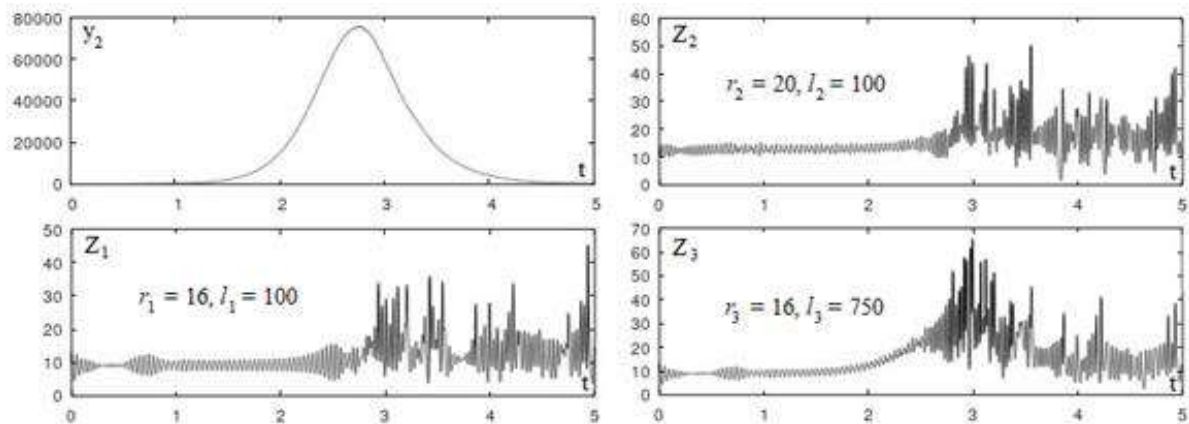


Figure 5: Modeling the emergency of chaotic operating regimes in various sectors of economy.

There exists a mutual negative influence. On one hand, the state of the economy affects the ability to implement effective measures against the pandemic. On the other hand, the infection rate influences the state of the economy. Figure 6 presents the results of modeling with variations in parameters b_i . The increase in these parameters simulates a decrease in job supply levels in sectors of the economy related to food production, its transportation, and medical infrastructure. We considered the case where these parameters are equal to each other. The growth of b_i leads to a significant change in the phase portrait of the system, resulting in a reduction in the levels of structural disruptions in various sectors of the economy Z_i and a decrease in the peak number of infected individuals y_2 in the chosen initial modeling interval. With b_i growing the overall number of infected individuals Σy_2 initially decreases and then begins to rise again.

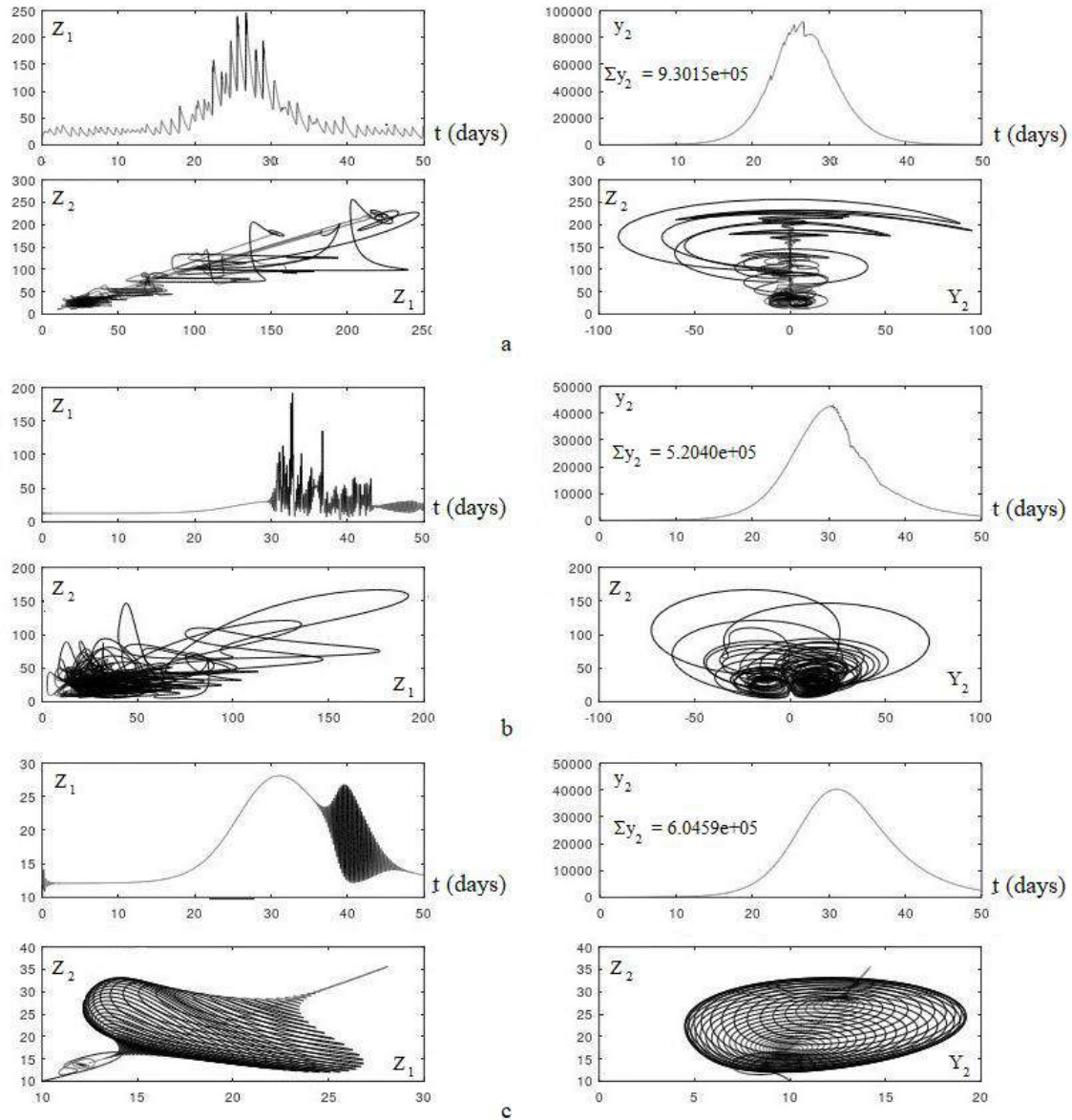


Figure 6: Changes in the level of infected individuals and structural disruption with parameters b_i variations: a) $b_i = 1$; b) $b_i = 15$; c) $b_i = 17$.

When modeling, we used a standard package for solving differential equations - the GNU Octave environment version 4.4.1, which is licensed under the GNU GPL and can run on Linux, macOS, BSD and Windows operating systems. Identification of model parameters was carried out using data [2]. According to these data, the peak of the epidemic in Ukraine was reached after 3 months, with the number of deaths reaching about 100,000 after 5 months when the first wave of the epidemic subsided. The total number of infected individuals during this time was over 3.5 million. These data align well with the results of modeling the first wave of the pandemic's development in Ukraine. To estimate the parameters of model (1), more detailed models [7, 8]

were used, developed as part of the joint project of the National Academy of Sciences of Ukraine and the International Institute for Applied Systems Analysis (Austria) "Complex modeling of the management of the safe use of food, water, and energy resources for sustainable social, economic, and environmental development."

3. Modeling the T-cell immune response in COVID-19

One of the most serious complications of COVID-19 is an excessive immune response, in which the level of pro-inflammatory cytokines responsible for regulating intercellular and inter-systemic interactions sharply increases (a "cytokine storm"). This leads to the disruption of the immune system, an increase in the level of free radicals, causing multiple damages to internal organs (lungs, heart, kidneys, blood vessels, liver, gastrointestinal tract, brain), severe multiorgan failure, and alters the balance of synthesis and expenditure of energy in cells, posing a lethal risk. Unfortunately, the specific mechanism of the cytokine storm remains undefined, making research on the impact of the immune system on the course of COVID-19 highly relevant.

Typically, when modeling the epidemic process, the key parameters determining the dynamics of epidemics include: 1) the average incubation period of the disease; 2) the average active period when the patient is contagious; 3) the average recovery period; 4) the average period until death. It is important to note that these constants in epidemiological models are functions of the state of the patient's immune system, which plays a crucial role in fighting infection. However, there are currently no epidemiological models that allow for the consideration of the influence of the immune system's state on the course of the disease and the selection of therapy. Therefore, the task arises to develop a mathematical model of the immune response in COVID-19 and analyze, through it, possible mechanisms of imbalance between different components of the immune system leading to a cytokine storm. This task will be addressed in the following section.

3.1. Mathematical model of immune system

In mathematical modeling of the immune system response to antigenic determinants of various natures (viruses, allergens, tumor antigens, etc.), typically, the components facilitating antigen presentation, its destruction, regulation of proliferative processes, and modulation of the immune response are considered. These include helper, killer, macrophage, and suppressor cells [9].

In each of these links, there is a group of precursor cells, including cytotoxic T-cells $-Y_{pK}$, helper T-cells $-Y_{pH}$, normal macrophages $-Y_{NM}$, suppressor T-cells $-Y_{pS}$; as well as a group of mature cells forming the immune response, including effector T-cells $-Y_K$, helper T-cells $-Y_H$, activated macrophages $-Y_{AM}$ and suppressor T-cells $-Y_S$. The listed cellular populations are responsible for the recognition and destruction virus-infected cells (VIC) $-Y_{VC}$. VIC debris $-Y_D$ contribute to the emergence of antigen-presenting cells (AK).

When infected with the coronavirus, the infection affects gene expression and protein synthesis. The virus imposes its protein synthesis algorithms on cells, necessary for its replication. Within hours, it neutralizes the cell's antiviral signaling, delaying and confusing the immune response. The virus reduces the ability of infected cells to translate genes into proteins, thereby decreasing overall protein synthesis. Additionally, it actively degrades cellular messenger RNA (mRNA), while its own mRNA remains protected. Finally, the virus can also prevent the export of mRNA from the cell nucleus, where they are synthesized. Based on the analysis of data [10, 11] on the peculiarities of the immune response in COVID-19, the following assumptions can be made, which should be considered in the model.

1. VIC accelerates the transition of precursor cytotoxic T-cells to a mature form and enhances the proliferation of effector cells. Moreover, they promote the production of lymphoid factors (F) by helper cells: interleukins, growth factors, and others that activate the proliferation of mature T-cells, the transition of normal macrophages into the activated form, and the inflammatory process.

2. Increased inflammation (I) enhances the proliferation rate of mature effector, helper, suppressor T cells, and activated macrophages, as well as accelerates the influx of all precursors.

3. *VIC* are recognized and destroyed by cytotoxic T-lymphocytes and activated macrophages.
 4. The proliferation of helper cells is activated after contact with antigen-presenting cells (*AK*).
 5. The increase in the level of effector cells activates the process of maturation of suppressors. Suppressor cells inhibit the processes of proliferation and maturation of helper and effector cells, as well as the processes of *VIC* destruction by effectors and macrophages.

6. There is a temporary hierarchy that allows identifying a group of fast variables that manage to reach a stationary state: antigen-presenting bed, lymphoid factors, and the inflammatory reaction.

7. The coronavirus can evade immune system recognition through suppression at the precursor cell level, thereby reducing the levels of killer and helper T-cells. Additionally, it may affect the efficiency of killer T-cells and activated macrophages in destroying cells infected with the coronavirus.

8. The coronavirus can also infect activated macrophages. This increases the level of *VIC* due to the action of virus-infected macrophages Y_{VM} .

9. One of the possible paths of COVID-19 immunopathogenesis is the induction of damage and death to vessel endothelium due to virus replication in the infection focus. Uncontrolled inflammatory reactions lead to significant tissue damage and the development of severe forms of the disease.

10. Tissue damage results in an energy imbalance, disrupting the interrelationships between energy function Y_{CF} and mitochondrial activity Y_{CM} for energy synthesis in the cell. A significant indicator of this imbalance, which increases with infection growth, is the number of structural disruptions Y_{CD} in the cellular energy system. An increase in Y_{CD} leads to a decrease in the activity of effector cells to destroy *VIC*. In [12], the Lorenz model was used to investigate these relationships. We will use these results to describe the energy block of the model for cells affected by the coronavirus. Taking these assumptions into account, the model has the following form:

$$\begin{aligned}
 \frac{dY_{pK}}{dt} &= a_1(1 + a_2I) - \frac{a_3Y_{pK}Y_{VC}}{(c_1 + c_2Y_S)(d_1 + d_2Y_{VC})} - a_4Y_{pK}, \\
 \frac{dY_{pH}}{dt} &= a_5(1 + a_6I) - \frac{a_7Y_{pH}AK}{(c_3 + c_4Y_S)(d_3 + d_4Y_{VC})} - a_8Y_{pH}, \\
 \frac{dY_{NM}}{dt} &= a_9(1 + a_{10}I) - a_{11}Y_{NM}F - a_{12}Y_{NM}, \\
 \frac{dY_{pS}}{dt} &= a_{13}(1 + a_{14}I) - a_{15}Y_KY_{pS} - a_{16}Y_{pS}, \\
 \frac{dY_K}{dt} &= \frac{a_3Y_{pK}Y_{VC}}{(c_1 + c_2Y_S)(d_1 + d_2Y_{VC})} - a_{17}Y_K + \frac{a_{18}Y_KI}{(c_5 + c_6Y_S)}, \\
 \frac{dY_H}{dt} &= \frac{a_7Y_{pH}AK}{(c_3 + c_4Y_S)(d_3 + d_4Y_{VC})} - a_{19}\frac{a_{20}Y_HF}{(c_7 + c_8Y_S)}, \\
 \frac{dY_{AM}}{dt} &= a_{11}Y_{NM}F - a_{21}Y_{AM} - a_{22}Y_{AM}Y_{VC}, \\
 \frac{dY_S}{dt} &= a_{15}Y_KY_{pS} - a_{23}Y_S + a_{24}Y_SI, \\
 \frac{dY_{VM}}{dt} &= a_{22}Y_{AM}Y_{VC} - a_{25}Y_{VM}, \\
 \frac{dY_{VC}}{dt} &= \frac{a_{26}Y_{VC}}{(c_9 + c_{10}Y_{VC})} - \frac{a_{27}Y_{VC}(Y_K + Y_{AM})}{c_{10} + d_5Y_{CD} + Y_{VC}} + \frac{a_{28}Y_{VM}Y_{VC}(1 + Y_{VC})^3}{L + (1 + Y_{VC})^4}, \\
 \frac{dY_D}{dt} &= \frac{a_{27}Y_{VC}(Y_K + Y_{AM})}{c_{10} + d_5Y_{CD} + Y_{VC}} - a_{29}Y_D, \\
 \frac{dY_{CF}}{dt} &= \sigma(Y_{CM} - Y_{CF}), \quad \frac{dY_{CM}}{dt} = Y_{CF}(r + d_6I - Y_{CD}) - Y_{CM} \quad \frac{dY_{CD}}{dt} = Y_{CF}Y_{CM} - bY_{CD},
 \end{aligned} \tag{5}$$

$$AK = \frac{Y_D(Y_{NM} + Y_{AM})}{c_{11} + Y_D}, \quad F = \frac{Y_H Y_{VC}}{c_{12} + Y_{VC}}, \quad I = \frac{c_{13} F}{c_{14} + F},$$

where a_i are constants that characterize: $i = 1, 5, 9, 13$ - inflows of $Y_{pK}, Y_{pH}, Y_{NM}, Y_{pS}$, respectively; $i = 2, 6, 10, 14$ - delay mode for activation of $Y_{pK}, Y_{pH}, Y_{NM}, Y_{pS}$ inflows, respectively; $i = 3, 7, 11, 15$ - speed of Y_K, Y_H, Y_M, Y_S , cells maturation, respectively; $i = 4, 8, 12, 16, 17, 19, 21, 23, 25$ - death rates of $Y_{pK}, Y_{pH}, Y_{NM}, Y_{pS}, Y_K, Y_H, Y_M, Y_S$, respectively; $i = 18, 20, 24, 26$ - population reproduction rates of Y_K, Y_H, Y_S, Y_{VC} , respectively; $i = 27, 29$ - destruction rates of Y_{CV} and Y_D , respectively; $i = 22$ - this rate quantifies how quickly viruses infect macrophages Y_{VM} ; $i = 28$ - this rate represent how infected macrophages stimulate the reproduction or replication of viruses; $c_k (k = \overline{1,14})$ - are constants that characterize the nonlinear effects of the interaction between different chains of the immune system; $d_l (l = \overline{1,4})$ - are constants characterizing immunosuppression from the coronavirus; d_5 - is a constant characterizing a decrease in the efficiency of virus destruction due to structural disruptions in the energy system Y_{CD} ; d_6 - is a constant characterizing the influence of inflammation on the energy system; L - is a constant characterizing the activity of stimulating virus reproduction from the side of macrophages Y_{VM} ; r, σ, b are parameters of the Lorenz model, which are functions of energy synthesis and expenditure processes in the cell [12].

3.2. Model investigation of immune system response in COVID-19

The mathematical formulation of the problem is as follows: given the model (4), the Cauchy problem needs to be solved over the time interval $[0, T]$ for the specified intervals of model parameters. Consider the immune response in conditions where there are no interconnections between the immune and energy systems ($d_1 = d_3 = 1$; $d_2 = d_4 = d_5 = d_6 = 0$), and there is no infection of macrophages ($a_{22} = 0$).

Figure 7 illustrates how the dynamics of the immune response change with an increase in the parameter c_{13} , which characterizes the influence of lymphoid factors on the level of inflammation. $Y_{VC}, Y_D, Y_{AM}, F, AK$ in all figures are measured by the number of cells, Y_{CD} is measured by the number of violations, I is dimensionless variable.

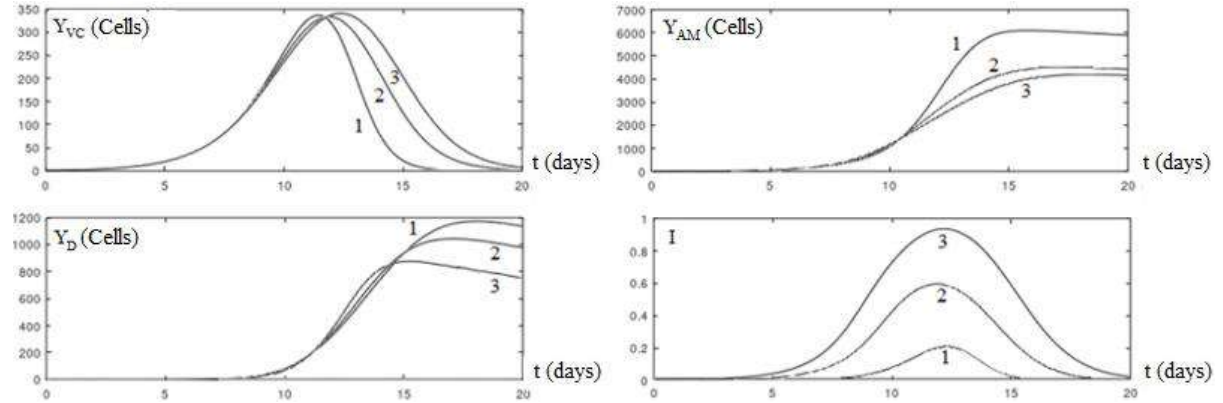


Figure 7: Changes in the dynamics of the immune response when parameter c_{13} increases: 1 - $c_{13} = 9$; 2 - $c_{13} = 50$; 3 - $c_{13} = 100$.

Figure 8 shows how the dynamics of the immune response changes when the initial number of Y_{VC} increases. As follows from the simulation results, at relatively high initial levels of Y_{VC} , the intensity of the immune response is sufficient for rapid destruction of Y_{VC} . However, with small initial values of viruses (curve 4), the immune response is delayed, so viruses have time for reproduction and their complete destruction requires more time.

Let's explore what changes in the immune response will result from considering the interactions between the immune and energy systems. In figure 9, the results of modeling are presented with variations in the parameter d_5 . This parameter characterizes the degree of inhibition of the effector function of killer cells and activated macrophages by the coronavirus.

The modeling results indicate that an increase in the degree of inhibition (increasing of d_5) leads to an increase in the time to reach the peak in lymphoid factors, inflammation, and Y_{VC} . The level of disruptions in the cellular energy system Y_{CD} also increases. (S - the area under the curve of disruptions).

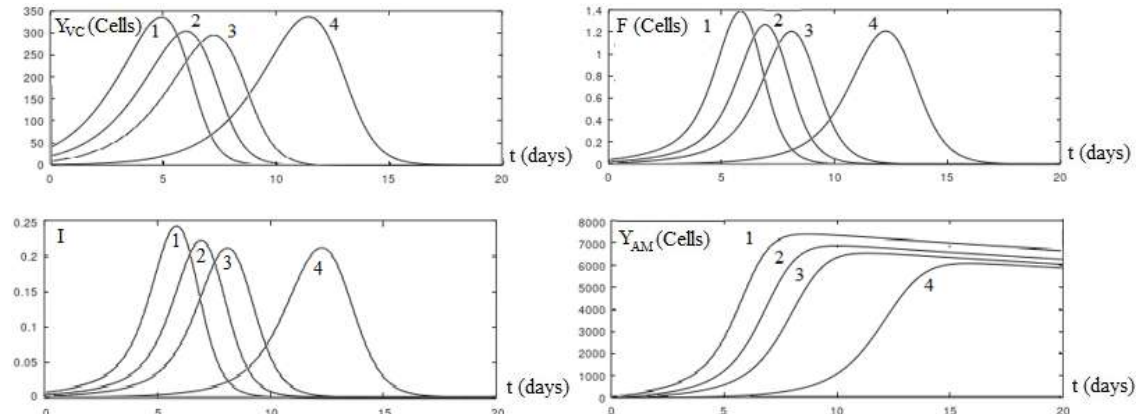


Figure 8: Changes in the dynamics of the immune response when initial level of Y_{VC} increases; 1 - $Y_{VC}(0) = 40$; 2 - $Y_{VC}(0) = 20$; 3 - $Y_{VC}(0) = 10$; 4 - $Y_{VC}(0) = 1$.

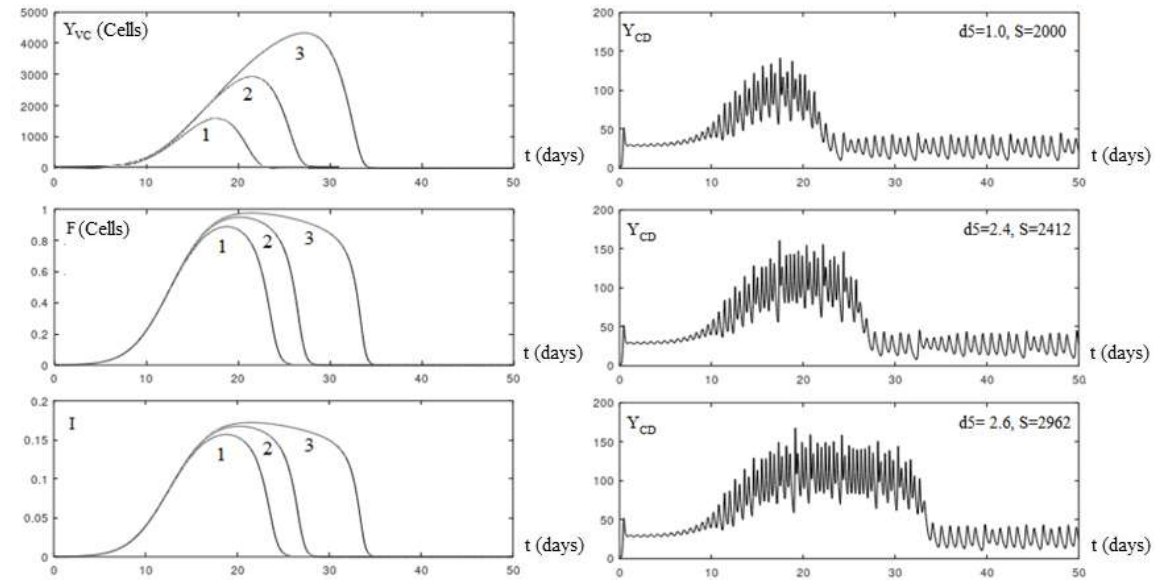


Figure 9: Dynamic of model variables with variation of parameter d_5 : 1 - $d_5 = 1$; 2 - $d_5 = 2.4$; 3 - $d_5 = 2.6$.

The results of the simulation for the case where the VIC infects macrophages are shown in figures 10 and 11. As evident from figure 10, when the value of parameter a_{28} reaches 0.07, the total amounts of the viruses, lymphoid factors, inflammation, and disruptions sharply increase, corresponding to the cytokine release syndrome.

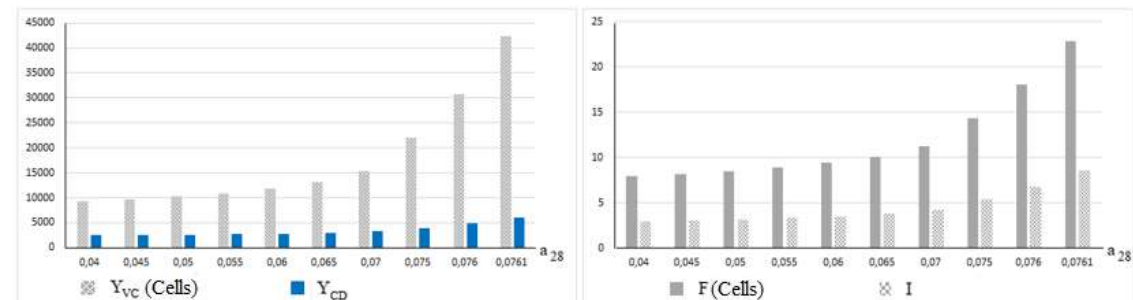


Figure 10: The dependency of the total quantity of VIC, lymphoid factors, inflammation, and disruptions on the parameter a_{28} .

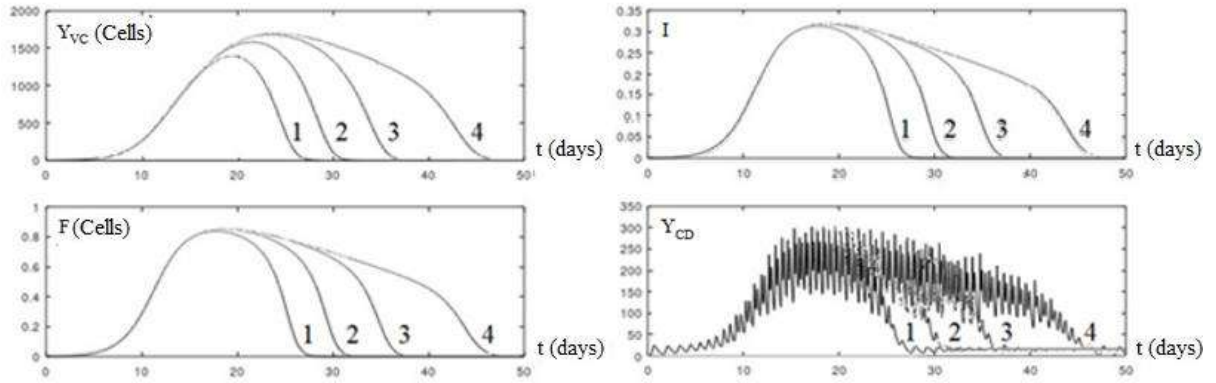


Figure 11: The dynamics of VIC, lymphoid factors, inflammation, and disturbances with an increase in the a_{28} : 1 – $a_{28} = 0.07$; 2 – $a_{28} = 0.075$; 3 – $a_{28} = 0.076$; 4 – $a_{28} = 0.0761$.

Further growth leads to uncontrolled reproduction of the virus, which is incompatible with life. As seen from this figure, the system is highly sensitive to small changes in the parameter a_{28} within the interval $[0.07, 0.0761]$.

If we consider the variables of model (5) as the mean values across a selected population, then the indicator of the number of structural disruptions Y_{CD} could characterize the collective immunity of this population. It is possible to examine the dependence of the parameters of model (3) on Y_{CD} . The relationships between (3) and (5) will be defined through the parameters a_0, T_{inf}, T_m . Parameters k_1 and k_2 from (4) also may be examined as functions of Y_{CD} . This will be considered in the further development of this work.

4. Conclusions

A model of the COVID-19 pandemic spread and associated changes in individual economic sectors has been developed. An approach to studying the interrelation within the food production system, food transportation, and the socio-economic sphere has been proposed using the three-sector Lorenz model. The model integrates economic sectors, uniformly described within a single framework, each of which is considered in terms of productivity, employment, and structural disruptions. The model studies were conducted with aim of: 1) assessing the influence of initial conditions and the state of economic sectors on the dynamics of the epidemic; 2) analyzing the impact of the epidemic process on interconnected sectors of the economy. The existence of deterministic and stochastic operational modes of interconnected economic sectors during a pandemic has been demonstrated. The transition from one mode to another is accompanied by an increase in the level of structural disruptions. The dynamics of these disruptions have been determined, along with the dependence of their total quantity on the epidemic process.

A mathematical model of T-cell immune response has been developed considering the peculiarities of its course in COVID-19, where excessive inflammation and increased levels of pro-inflammatory cytokines lead to cytokine release syndrome, causing damage to vital organs and disruption of immune system. The model considers the influence of cellular energetics, particularly mitochondria, on the regulation of the immune response and the amplification of inflammation. The simulation results have allowed investigation of the following: 1) how increases in the initial levels of virus-infected cells affect the average recovery period of severe symptomatic patients; 2) how inhibition of effector functions of killer cells and the level of inflammation affect the dynamics of the immune response; 3) how changes in the balance of energy synthesis and expenditure within the cell affect the occurrence of periodic and turbulent trajectories of the immune response.

The scientific novelty of the work lies in the following: 1) a proposed approach to the development of complex systems based on the use of multi-sector Lorenz models, allowing the exploration of interrelationships between the state of economy and the dynamics of epidemic; 2) the development of a mathematical model that accounts for the role of herd immunity in the dynamics of pandemic spread; 3) the developed models consider the superposition of two types

of random processes - "deterministic" Lorenz stochasticity and traditional Wiener stochasticity. Further development of the work will be associated with solving optimization tasks for immune response management aimed at minimizing virus-induced damage to the organism and substantiating various therapeutic regimens used in treatment.

Acknowledgment

The work was carried out as part of the scientific project PK № 0122III00552 "Comprehensive analysis of robust preventive and adaptive measures for managing food, energy, water, and the social sphere in conditions of systemic risks and consequences of COVID-19".

References

- [1] The Global Risks Report 2021(16th Edition). World Economic Forum, 2021. URL: https://www3.weforum.org/docs/WEF_The_Global_Risks_Report_2021.pdf.
- [2] WHO COVID-19 Dashboard. URL: <https://data.who.int/dashboards/covid19/cases>.
- [3] K. Schwab, T. Mallert, COVID-19: The Great Reset, 2020. URL: <http://reparti.free.fr/schwab2020.pdf>.
- [4] K. Atoiev, P. Knopov, Modeling of complex systems instability under risks and uncertainty, in: Proceedings of the 3rd. IEEE conference on System Analysis & Intelligent Computing (SAIC), Kyiv, Ukraine, 2022, pp. 1–4. doi:10.1109/SAIC57818.2022.9922972.
- [5] A. L. Bertozzi, E. Franco, G. Mohler M.B. Short, D. Sledge, The challenges of modeling and forecasting the spread of COVID-19. Proceedings of National Academy of Sciences of USA. (2020) 16732-16738. doi.org/10.1073/pnas.2006520117.
- [6] J. Yorke, E. Yorke, Metastable chaos: the translation to sustained chaotic behavior in the Lorentz model, J. Stat. Phys. (1979), 263–278.
- [7] V.A. Pepelyaev, A.N. Golodnikov, N.A Golodnikova, Reliability Optimization in plant production. Cybernetics and System Analysis (2022), 191–196. doi.org/10.1007/s10559-022-00450-5.
- [8] V.A. Pepelyaev, A.N. Golodnikov, N.A Golodnikova, Reviewing Climate Changes Modeling Methods, Cybernetics and System Analysis (2023), 398–406. doi: 10.1007/s10559-023-00574-2.
- [9] R. Eftimie, J.J. Gillard, D.A. Cantrel, Mathematical models for immunology: current state of the art and future research directions Bull Math Biol (2016), 2091–2134. doi 10.1007/s11538-016-0214-9
- [10] R. Channappanavar, S. Perlman, Pathogenic human coronavirus infections: causes and consequences of cytokine storm and immunopathology, Seminars in. Immunopathology (2017), 529–539. doi.org/10.1007/s00281-017-0629-x
- [11] Y. Fu, Y. Cheng, Y. Wu Understanding SARS-CoV2-mediated Inflammatory responses: from mechanisms to potential therapeutic tools, Virologica Sinica, (2020), 266–271. doi: 10.1007/s12250-020-00207-4.
- [12] K. Atoiev, P. Knopov, V. Pepeliaev P. Kisala, R. Romaniuk, M. Kalimoldayev, The mathematical problems of complex systems investigation under uncertainties, in: W. Wojcik, J. Sikora (Eds.). Recent advanced in information technology, London: CRC Press Taylor Francis Group, 2018, pp. 135–171. doi.org/10.1201/9781351243179-6

Intelligent System for Processing and Forecasting Financial Assets and Risks

Nickolay Rudnichenko¹, Vladimir Vychuzhanin¹, Tetiana Otradska¹ and Denys Shvedov¹

¹ Odessa Polytechnic National University, Shevchenko Avenue 1, Odessa, 65001, Ukraine

Abstract

The article contains a description of the results of development and research of intelligent system for processing, accounting and forecasting financial assets and risks. An analysis of current problems and technical aspects related to the assessment of financial flows and risks of loss of assets in the context of modern market development in the field of accounting and forecasting of financial time series was carried out. A justification is given for the feasibility and effectiveness of using machine learning methods to automate the solution of operational and analytical problems in the field of accounting for financial assets. The proposed concept of using machine learning within the framework of the developed application software is described, the key aspects of its design and implementation of the software and modular structure are considered. The results of a comparative analysis of the created ACF and PACF models, assessment of the quality metrics of robotic models in different conditions when assessing financial flows and risks are presented, an analysis of the results obtained is carried out and further ways of developing the approach proposed in the article to improve its efficiency are considered.

Keywords

Data analysis, finance risks, financial time series, machine learning, software development, automatic correction of trend and seasonality

1. Introduction

1.1. Research relevance

In the modern world, where data volumes are constantly growing, the importance of accurate accounting and forecasting of financial indicators becomes particularly significant. The relevance in this field primarily lies in the increasing need for precise and efficient tools capable of automating operational and analytical actions on data from financial flows, taking into account the risks of incorrect investments [1].

This not only allows optimizing investors' budgets but also provides client companies with means for more effective management of material resources in terms of planning potential revenues and various expenses, ensuring the principles of overall financial stability.

In the context of formalizing the process of processing data from financial flows or indicators, it should be noted that their nature can vary. Specifically, data from the outset may be well structured, semi-structured, or unstructured [2].

Structured financial data have a clearly defined format, often adhering to a relational database model, characterized by a standard hierarchical order.

They are highly organized, easily accessible, and directly utilized for analysis. In well-defined schemas and collections, structured data is predominantly stored in tabular formats. For instance, names, dates, addresses, credit card numbers, stock information, geolocation, etc., serve as examples of structured financial data [3]. Semi-structured financial data are often not stored in relational databases like structured data but possess certain organizational properties that facilitate their analysis.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ nickolay.rud@gmail.com (N. Rudnichenko); icst_nuop@ukr.net (V.Vychuzhanin); tv_61@ukr.net (T. Otradska); studylearnerstudy@gmail.com (D. Shvedov);

ORCID 0000-0002-7343-8076 (N. Rudnichenko); 0000-0002-6302-1832 (V.Vychuzhanin); 0000-0002-5808-5647 (T. Otradska); 0009-0002-4823-8782 (D. Shvedov)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Examples of such data include risk profiles, investment plans, and various financial accounting data. HTML, XML, JSON documents, NoSQL databases, etc., are the examples of semi-structured data types used in financial analysis.

For unstructured financial data, there is no predefined format or organization, significantly complicating their collection, processing, and analysis, as they mostly consist of textual and multimedia materials. Examples of unstructured financial data include dividend or investment reports, emails, blog posts, wikis, text documents, PDF files, images, presentations, web pages, and many other types of business documents containing valuable financial information [2, 4].

Within the framework of managing various types of data from financial assets of users or companies, the promising approach is the application of modern machine learning (ML) methods. This direction opens up a range of possibilities for automating the processing and analysis of large volumes of diverse financial data (executed transactions for the payment of goods, receipt of services, acquisition of securities, crypto assets, or other investment contributions). It allows uncovering hidden patterns and dependencies that may not be easily identified by humans in manual mode or through the use of specialized deterministic calculation programs [5].

The advantages of modern ML algorithms include their ability to automatically adapt to dynamic changes in financial profiles and investment risks of users, allowing an increase in the accuracy of data forecasting. ML algorithms can identify patterns and relationships in data that may be non-obvious to humans. This enables precise financial forecasts and informed investment decisions based on the analysis of diverse data.

Consequently, the application of ML in the field of financial risk assessment holds significant practical interest. Specifically, accounting and forecasting of financial indicators can be utilized in various sectors such as banking, financial consulting, asset management, and personal finance. The accuracy of such forecasts can greatly impact decision-making in these areas, particularly in risk management, budget planning, and investment planning [6].

Among the features of applying supervised ML algorithms, it is important to note that the models created upon these algorithms form a solution basis for tasks, relying mostly on information obtained from the input dataset. Another important characteristic of ML models is their ability to self-improve, meaning the formalization and utilization of acquired experience based on a data-driven approach. In fact, the more diverse and balanced data models analyze, the more accurate their predictions become. This is particularly relevant in sectors where high accuracy is required, such as finance, medicine, or security systems [7].

The primary scientific challenge in this context is achieving and instilling in the model a high level of generalization ability to effectively solve tasks on new, previously unknown data not used during training, with a minimal level of errors. This is why, when using ML in the field of financial analysis, it is necessary to integrate approaches from other applied and theoretical scientific directions, including mathematical statistics, optimization methods, and traditional mathematical domains. It is important to take into account the unique characteristics of model-building algorithms related to both non-functional aspects (computational efficiency or speed) and functional aspects (overfitting issues, sample balance, hyperparameter tuning) of ML utilization [8].

Various methodologies are used for forecasting financial data and creating client profiles, typically based on the analysis of historical data and influencing factors. This involves establishing statistical relationships between different characteristics and developing predictive models. Previously, the single-factor forecasting methods were dominant, relying on time series and regression analyses. However, not all of these approaches are sufficiently effective for assessing financial indicators and investment potential [8]. All of this makes the research topic relevant within the scope of this work.

1.2. Aspects of financial data processing issues

It is necessary to note that the specificity of processing and accounting financial data for their preliminary preparation to address forecasting tasks involves several substantive factors.

Specifically, forecasting in the financial domain is a non-trivial task as it entails analyzing and predicting changes in various financial flows, such as income, expenses, savings, and investments. Particularly in the risk analysis process, it is essential to consider the fact that personal finances have distinctive characteristics that vary in each specific case for different users. This is because they often rely on individual behavioral factors, which can be less predictable than market indicators. This process is crucial for individual financial planning as it helps individuals determine how best to allocate their resources to achieve financial goals [5].

In general, the process of accounting and forecasting finances is accompanied by a series of challenges, among which the following can be highlighted [9]:

- High unpredictability of the external environment. Chosen strategies of investment behavior and risk assessment can be extremely unpredictable due to various unforeseen events, such as sudden fluctuations in exchange rates, seasonal declines or surges in demand and supply, accidents and technological disasters, changes in population employment, political decisions and sanctions, or other types of crises. These events can significantly impact an individual's risk profile and complicate the creation of accurate long-term financial forecasts.
- Income variability. For many investors, income is not constant and may vary from month to month, especially for professionals or experts whose activities are largely commission-based. This variability complicates forecasting and requires a more flexible approach to budgeting and financial planning.
- Behavioral factors. Financial decisions often depend on behavioral factors such as personal beliefs, habits, emotions, and cognitive biases. For example, psychological barriers may hinder effective saving or investing, while impulsive purchases may lead to unplanned expenses.
- Economic and market conditions. External economic factors, such as inflation, interest rates, market fluctuations, and economic crises, can have a significant impact on personal finances. Predicting these conditions is challenging, but they must be taken into account when planning finances [10].

Thus, the specificity of forming and managing data structures regarding financial flows should be considered from the perspective of systematic intelligent analysis. Inaccuracies, data errors, outliers, and data incompleteness can significantly impact the quality of forecasts. In financial time series, there is a considerable amount of "noise" — random fluctuations that do not have an obvious explanation. Distinguishing the "signal" from the "noise" is one of the main challenges in financial modeling and forecasting, leading to various types of risks that are not always easily predictable and quantifiable with a high level of certainty. All of this underscores the relevance of this research and the fulfillment of the stated objective, which involves developing software for the accounting and forecasting of financial assets and risks based on the application of modern ML algorithms to automate the data analysis process.

1.3. Discussion with methods

Currently, there are various approaches to the analysis of heterogeneous data of time financial series of different orders and states (both stationary and non-stationary) in the context of integration with ML models. According to solving problem we should characterize: vector autoregression (VAR), generalized autoregressive conditional heteroskedasticity (GARCH) and automatic correction of trend and seasonality (ARIMA) [8,10].

VAR is a statistical method that allows us to model relationships between several time series simultaneously. A feature of the method is the expression of input variables in the form of linear combinations of past values that they took in the system. Var advantages are:

- taking into account the relationship between several time series, which can be used when analyzing heterogeneous financial data, for example, in the case when the prices of various assets or market indicators depend and influence each other stochastically;
- allows us to formalize the modeling of various emerging effects and delays in relationships in the data, which helps to identify key criteria for the financial series over time;

- there is potential application for assessing the impact of various non-obvious factors on financial risks.

Disadvantages:

- choosing the optimal number of lags and parameters is a labor-intensive and non-trivial process, requiring a large number of experiments on heterogeneous data;
- difficulty in interpreting results in the case of non-stationary processes;
- difficulties in software implementation for risk assessments in the context of their ranking into groups and formalization of the probabilistic factor.

GARCH is used to model and forecast the volatility of financial time series. GARCH allows us to take into account the nature of the occurrence of variability in data, which helps to predict future levels flows volatility, which is used in the securities valuation [9]. The volatility of a time series can change over time and depend on past values of this series; modeling is implemented through conditional heteroscedasticity based on previously obtained values (conditional variance depending on past squared errors). Advantages:

- taking into account variability in data to formalize the data structure, which can increase the accuracy of the forecast;
- adaptability for stationary time series by supporting modeling of dynamic volatility effects, including by clustering data according to internal financial market conditions.

Weakness:

- the limitation of the method in the context of use for big data due to the need to conduct a large number of heterogeneous experiments to obtain reliable results;
- high sensitivity to initial conditions, which can lead to unstable results if the parameters or model specification are incorrectly selected;
- are not effective when the nature of the dependencies of financial time series is initially uncertain (probabilistic).

ARIMA is a statistical method for analyzing and forecasting time series that allows us to model a financial time series as a combination of autoregressive, integrated and moving average components. The process of building an ARIMA model includes the following steps: identifying the necessary parameters based on the analysis of autocorrelations and partial autocorrelations of the time series, estimating the model parameters using the least squares or maximum likelihood method, checking the adequacy of the model, and analyzing the residuals [11]. ARIMA advantages:

- adaptability by supporting automatic correction of trend and seasonality, as well as searching for anomalies and patterns in the data.
- flexibility in accounting for integrated and moving averages through the use of complex components;
- support for forecasting horizon management capabilities, including short-term and long-term time periods, which allows expanding models in composite forecasting scenarios;
- automatic identification of model parameters based on data;
- extensibility of the model through the possibility of aggregated accounting of external factors, which is useful to increase forecasts accuracy in risks assessing.

Disadvantages:

- difficulty of application for short and noisy financial time series with a decrease in forecasting accuracy;
- reduced efficiency when used for high-frequency data (intra-day or online fed in multi-stream collection systems) due to the need for a large number of observations and a high degree of autocorrelation.

As a result of the comparative analysis of the considered methods within the framework of our task, the most appropriate is the use of ARIMA due to the greater degree of adaptability and software implementation possibility in the separate module form.

2. Project concept proposal

Within this research, in addition to developing the software logic for a web application for financial data accounting, the creation of hybrid polynomial regression ML models for data

forecasting will be conducted, particularly based on ARIMA. ARIMA represents a unique combination of autoregressive models and moving average models. The advantage of this approach is the ability to use such models for automating the forecasting of time series values (in our case, the values of financial assets) based on their past values, taking into account the obtained error estimates from previous forecasts, providing the models with a higher degree of adaptability [11].

During the data analysis process, it is proposed to perform several preliminary stages. This includes data import and preliminary processing, their structuring, determination of statistical indicators, and assessment of data normality. Thus, it is possible to better understand the nature of the considered data and identify the characteristics of financial risks [12].

In this context, an important factor is the identification of the characteristic trend of seasonality in transactions and investments because it can negatively impact the accuracy of forecasting within a defined horizon. Therefore, it is necessary to exclude this aspect before the forecasting process begins. It is also essential to analyze the variance to determine whether the data changes over time or if the variance is constant, which serves as a criterion for the balance of the data sample.

As it is impossible to empirically check the normality of a large volume of data visually, it is advisable to implement in the created web application a check for data normality using the Jarque-Bera test with the following expression

$$JB = n \frac{s^2}{6} + \frac{(k-3)^2}{24} , \quad (1)$$

where n is the sample size, s is the skewness coefficient, and k is the kurtosis coefficient. For normally distributed data, $s = 0$ and $k = 3$. In this case, the value of JB is equal to 0.

According to the null hypothesis that the data is normally distributed, the Jarque-Bera test has shown that the JB test asymptotically follows the Chi-square distribution in this equation [13].

The definition of the autocorrelation function (ACF) and the partial autocorrelation function (PACF) is also an important concept for determining the models we create. In time series analysis, the ACF shows the degree of linear statistical relationship between the values of a time series. Numerically, the ACF is a sequence of correlation coefficients between the original series and its copy, shifted by a given number of series intervals (this number is called the lag ACF). In general view proposed ACF is defined as follows

$$f(L) = \sum_{L=0}^m r_{t,t-L} , \quad (2)$$

where m - time series members number, r - correlation coefficient.

The autocorrelation function (ACF) measures the linear predictability of time series at time t , as it only uses lagged values. The partial autocorrelation function allows a stationary financial time series to have partial correlation with its own values. This is convenient in our case due to the support for regressing data values at short lags.

Since most our financial data is stationary, applying ARIMA models requires transforming variables. Therefore, ACF and PACF must be analyzed in combination to determine the order of the model, specifically to minimize forecast errors in model creation.

The complete definition of the ARIMA model involves choosing values for the parameters p , d , and q , which is a complex task. Parameter p represents the order of the autoregressive part, d is the degree of differencing the series, and q is the order of the moving average part. It's worth noting that within the proposed concept of financial data forecasting, a necessary condition is $p > 1$ [14].

The key idea within this concept, reflecting its scientific novelty, is the development and application of the designated ARIMA approach using ML within the framework of the analysis of stationary data on financial transactions based on the probabilistic approach (probability is ranked according to the Harrington desirability function). To reduce the degree of uncertainty when analyzing financial risks, a fuzzy logic model based on the Sugeno algorithm is used. All this forms a hybrid model with significant generalizing capabilities for predicting financial risks over various planning horizons, reducing (formalizing) the overall level of uncertainty in the impact of external factors, which is implemented in software system.

3. Intelligent system implementation

Let's describe the process of implementing the software project based on creating diagrams that formalize its logic and structure. When working with the web application, the user has the option to choose from several functions, some of which are related and have a certain dependency on each other:

- User transactions import. This functionality allows the user to import bank statements (currently supporting statements from monobank in CSV format). The system transforms these statements into internal entities called Transactions. These data are directly used for forecasting the user's income and expenses.
- Verification of imported transactions in a list with the ability to filter and sort.
- Forecasting. The system provides access to the forecasting module, where we can configure certain parameters such as: the type of forecasting (expenses or profits), the forecasting period, or the forecasting horizon, the start date of forecasting, i.e., the date of the last transaction on which the model will be trained and from which the forecast will be generated for the specified number of months.
- Visualization. The system provides the ability to view forecasting results for each type of forecast. The display is presented in the form of cumulative forecast sums and transactions for the forecast period. This will allow for the observation of trends in data changes and correlation between the forecast and actual data.
- Financial risk assessment. Implements functionality to obtain optimistic and pessimistic risk values from increasing expense levels and decreasing profit levels within the forecasting horizon based on the scenario method.

The developed scheme for detailing the stages of operation of the web application for accounting, processing, and forecasting user financial data is shown in Fig. 1. An important detail of the architecture of the created web application is the support for the re-creation of the model on different data fragments, its subsequent training, and use for generating user-created forecasts.

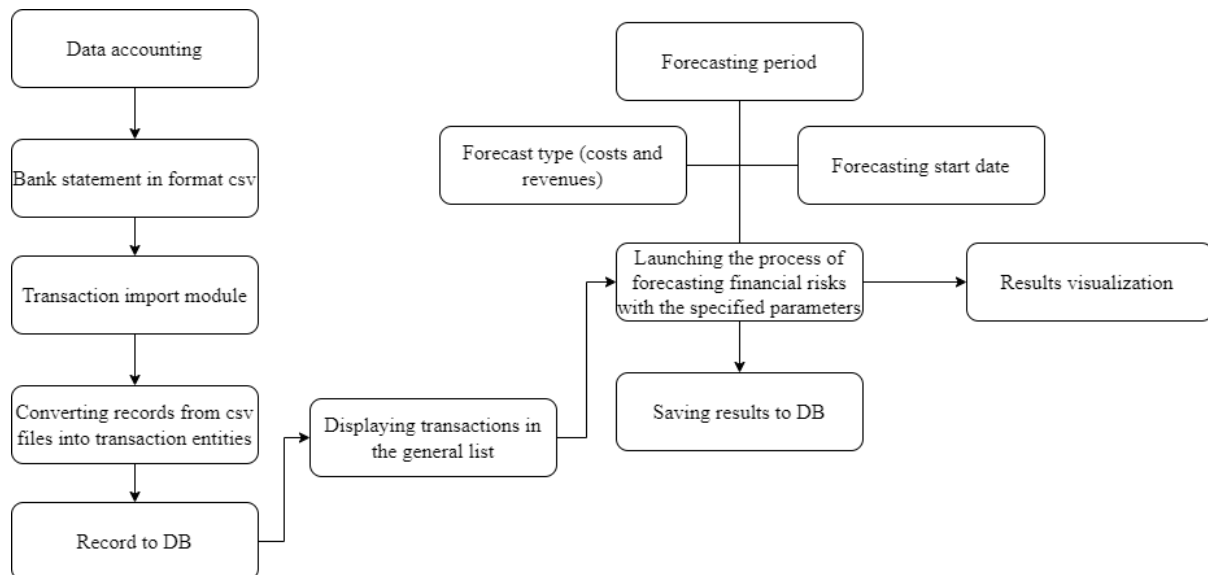


Figure 1: The detailed scheme of the web application's operational stages

The models used in the web application optionally have the ability for dynamic learning, which is necessitated by the need to use new portions of financial data for more detailed analysis and the formation of a more generalized forecast each time a model is generated. To overcome this problem, the "Forecast" entity has been created, which contains the forecasting results and the parameters used in forecasting.

From the perspective of object-oriented design efficiency, this allows for a reduction in computational resource costs. If the dataset and configurations remain unchanged, the forecast result will remain the same.

Therefore, the module will retrieve results from the database without performing additional operations.

To describe the usage process of all modules in the web application during its design, we will use the UML (Unified Modeling Language). The developed general sequence diagram of the application's operation is presented in Fig. 2.

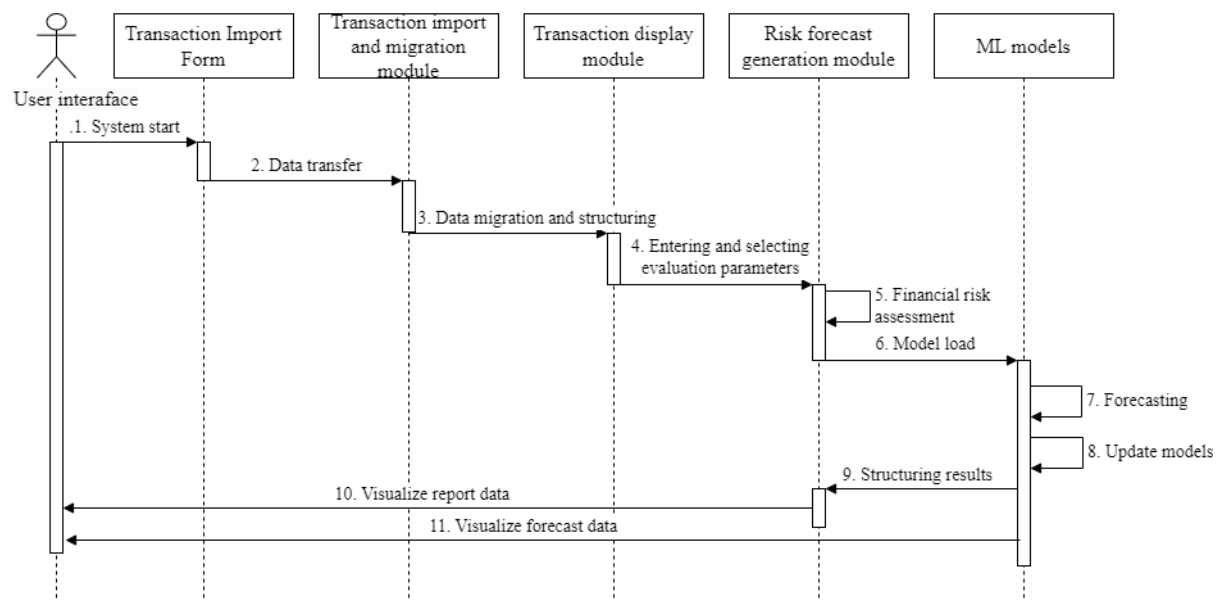


Figure 2: The sequence diagram of user actions in the software

After logging into the web application with their account, the user has the option to import transactions into the system through the transaction import form. The user can add a file or files in CSV format, generated by a bank or another financial institution with which they have relationships.

After migrating transactions into the system, the user can download a forecasting model to assess potential profits, expenses, and aggregated financial risks using the forecasting functionality. This module is programmatically linked to a form where the user can configure parameters for forecasting financial risks.

After confirming the parameters, they are passed to the ML control block, which deserializes the models, uses them, compares the parameters with the generated forecasts, and, if the data is correctly specified, forms a set of forecast values for each model.

The result is then sent to the user interface on the web page in the form of a report and graphical dependencies. In case the forecast is not reliable (error exceeds the set value) or the data is invalid, an informational message is displayed to the user in the form of a pop-up window.

From a software perspective, the application is a distributed microservices SPA based on JavaScript programming language, Nodejs platform, and React frontend framework, providing interactive user interaction capabilities. Due to the client-server architecture, the server-side backend and DB are separately launched and deployed.

The backend performs all functional tasks related to data accounting and processing, interacts with the DB for searching, filtering, creating, editing, reading, or deleting performed records.

Additionally, the Python programming language and a set of functional libraries for creating structures to manipulate data, including pandas and numpy, are employed.

Interaction between the backend and frontend parts of the web application is achieved through REST API. The backend interacts with MongoDB using mongoose. The library provides built-in validation for schemas to automate checking data compliance with specified rules and formats before they are saved or updated. The overall architecture of the software is shown in Fig. 3.

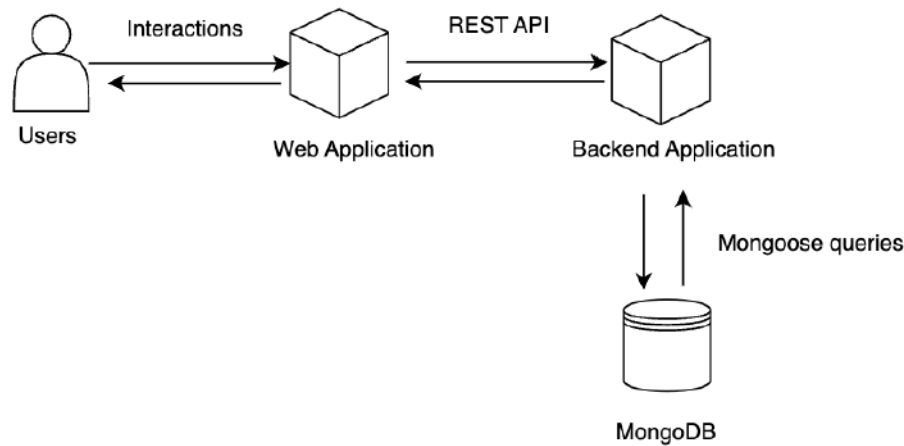


Figure 3: Architectural diagram of the components of the system in a general view

To perform a forecast for the selected model, data is required. In our case, this involves a test dataset of transactions over 3 years for the customer of monobank. Each transaction is described by the Transaction schema, and all transactions are stored in the system for further use in other modules. The Transaction entity retains information about each transaction in a structured format. The main modules that utilize the Transaction entity are the statistical processing and forecasting modules. Overall, the forecasting module performs only a few roles:

- formation of forecasted values for profits, expenses, and financial risks;
- a set of tools for interaction between the web application and a Python script, which is used for deserialization, description, training, and utilization of ML models;
- connection to the statistical module for aggregating transactions used to build graphical visualizations of the reporting results. With this data, the user can correlate real data with forecasted data, evaluate results with specified parameters, conduct additional computational experiments, and perform a consolidated analysis of financial risks, with the further use of data for developing their investment strategy.

The forecasting module uses the Forecast schema to record data about generated forecasts and the parameters used in the process. ForecastResult is a distinct software type, essentially a collection of data that describes a set of financial time series resulting from the model's forecast and has the following structure:

- `dateTime` [string], the date of one forecasted value of a financial asset, described in the form of an ISO string;
- `amount` [number], the amount of one forecasted value of a financial asset;
- `ForecastOptions`, a specialized object data type that describes the parameters used for generating a forecast or calculated for assessing the relevance of the forecast;
- `startTime` [string], the date from which the forecast creation will commence. It is also used for aggregating transactions; the model should not consider data beyond this date, as otherwise, the results would be inaccurate;
- `period` [1M, 2M, 3M], the period or forecast horizon. Currently, the software supports forecasting values for 1, 2, or 3 months, due to increasing errors and decreasing forecast reliability with longer periods;
- `nTransactions` [number], the number of transactions in the system, a parameter used to assess the relevance of the forecast;
- `modelVersion` [string], the version of the model, an internal parameter that essentially serves as metadata;
- `forecastType` [income, expenses], the type of financial asset for forecasting.

The user is provided with an authentication and registration system in the web application, so each user has their own personal record, and the User schema is used for this purpose.

The module responsible for importing transactions from the bank statement is the Mono module, which depends on the Transaction schema. Its main functionality involves parsing the bank statement structure in CSV format and migrating the data to the system. The

MonoController, which is an element responsible for managing endpoints for this module in the web application architecture, has only one API endpoint, /api/v1/mono/import. It is designed to receive the input configuration set for importing all transactions, including accountID, date (import start date), and the CSV file itself.

The imported transactions are utilized in the Analytics module, which is responsible for executing forecasts and aggregating transactions for correlation with the forecast. The module operates with the Forecast schema, used to store the forecast results in the DB. The module consists of AnalyticsController and AnalyticsService. AnalyticsController is responsible for the API and interaction with the main functionality in the service.

Below is a description of the code execution sequence for creating the ML model for forecasting:

1. Reading parameters from sys.argv, which are passed as requests when launching the Python process, namely period (forecast period) and forecast_type (forecast type). These two parameters are necessary for using factors in constructing the dataframe and determining parameters for the model.
2. Parsing transaction data from data.json using pandas.
3. Sorting and preparing data.
4. Determining model parameters based on the type of forecasting.
5. Normalizing data and creating additional factors.
6. Removing outliers for expense forecasting by introducing errors through increasing model indicators. This is part of the normalization process, where records in the dataframe with abnormally high or low values are removed.
7. Splitting data into test and training sets.
8. Creating an ML model.
9. Generating predictions for test data and determining model metrics.
10. Creating a forecast based on the period parameter.
11. Preparing output data and saving them to results.json.

The text indicates that illustrates the diagram of the software user interface (Fig. 4).

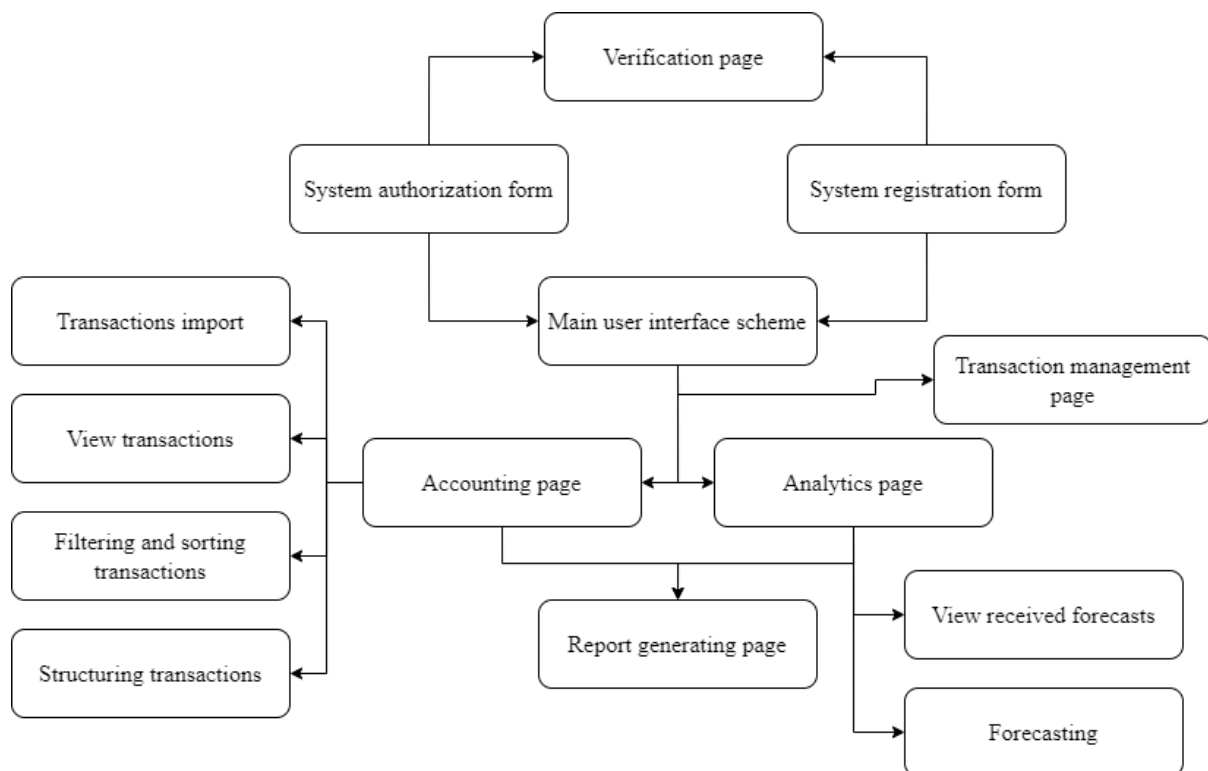


Figure 4: The pages of system's interface

For interactions with the transaction import and forecasting modules, a custom interface has been developed. The web application interface consists of several endpoints:

- Authorization, a page that greets the user and has a form for creating an account or logging in with pre-existing data, namely email and password.
- Transactions, a page consisting of a table that displays transactions created by the user or imported through the import module for accounting. The table provides the ability to sort and filter data based on certain available parameters. The user menu includes a button that opens a form for creating a transaction. The Import button, located in the upper right corner of the page, opens a form where the user can select specific parameters for import and a CSV file with a bank statement.
- Analytics, the page contains a button in the upper right corner, Forecast, which opens a form with settings for generating a forecast. After generating the forecast, graphs appear on the page displaying the forecasted results in the form of a line chart, as well as transactions for this period for a more detailed analysis and correlation between real and forecasted data.

After considering the main aspects of the system implementation, it is necessary to explore the possibilities of its application for financial data forecasting tasks.

4. ML models implementation and research

To conduct research using the created ML models, it is necessary to form training and testing datasets, which are formed based on a sample of 50,000 client transactions of Monobank over a 3-year period. The scikit-learn library's `train_test_split` method was used to create the datasets, allowing the dataset to be split using the `test_size` parameter.

The training dataset is provided to the hybrid model during its training along with corresponding evaluation results, enabling the model to establish a connection between input characteristics and accurate responses.

The testing dataset is used to assess the model's effectiveness. In this case, the model is not given the target characteristic; instead, it must predict it based on other characteristics. According to the obtained results, predictive values are compared with actual results.

The ACF and PACF plots based on (1) and (2) are shown in Fig. 5 depict the three parameters of the created hybrid ARIMA model: $p = 1, d = 1, q = 0$.

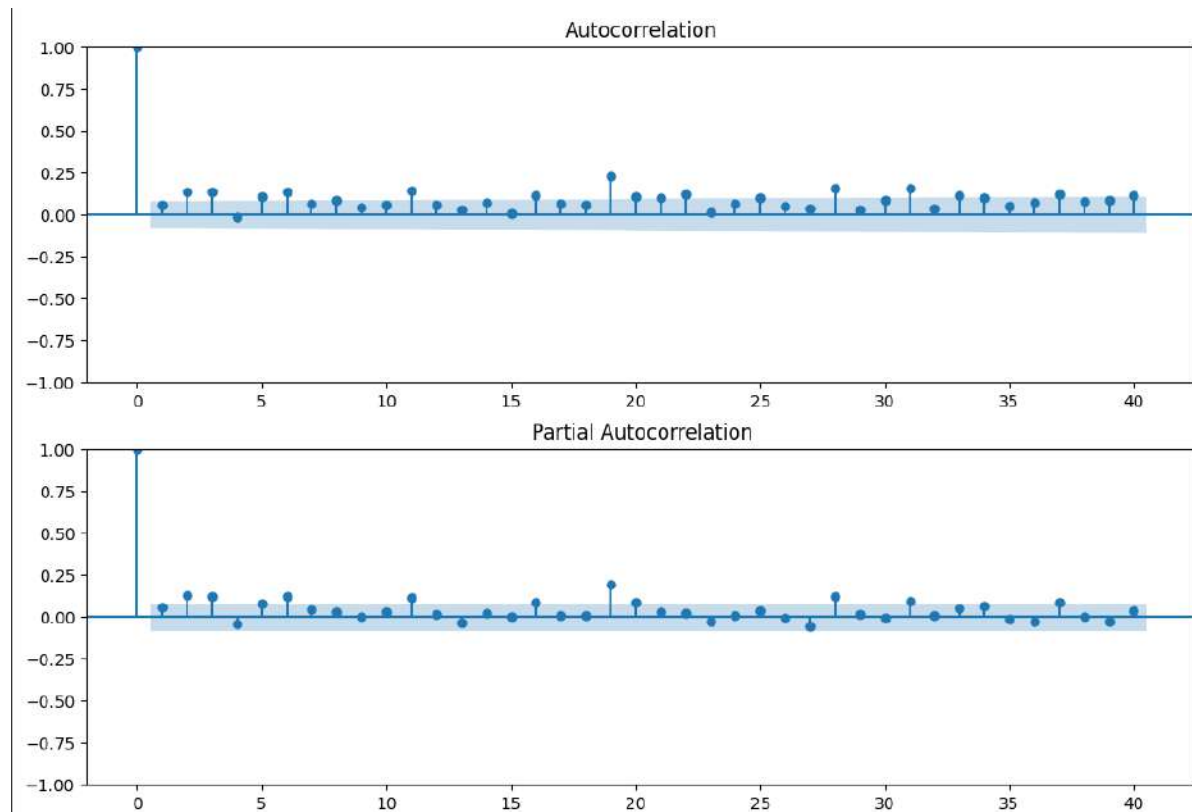


Figure 5: The ACF and PACF plots

Before starting the model training, it is important to establish criteria for evaluating its effectiveness to ensure the chosen algorithm's feasibility. One way to evaluate could be comparing the model's performance with results obtained through random generation of the target characteristic, i.e., a baseline model.

If the chosen algorithm shows worse results than simple random guessing, it may signal the need to reconsider the approach, possibly using alternative methods. In our case, for solving the financial risk regression task, it is appropriate to compare with the mean value of the training dataset. To compare the performance of the created hybrid model based on ARIMA, models such as Facebook Prophet and XGBoost were also chosen.

The results of measuring MAE for the constructed models are presented in Fig.6.

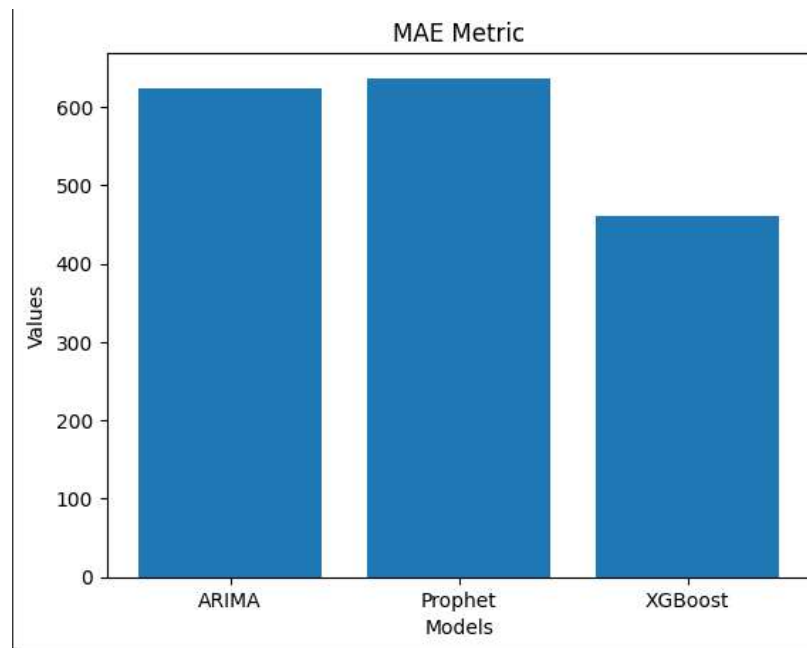


Figure 6: The results of measuring MAE for the constructed models

It should be noted that the overall nature of the distribution of the obtained evaluation results for the ACF and PACF models is quite balanced, with a spread not exceeding values of 0.25 on the positive side and 0.075 on the negative side.

The data sample is uniform, resulting in financial risk values ranging from 0.05 to 0.2. This indicates a fairly high accuracy in forecasting financial asset values and a moderate growth in calculation error.

As can be seen, the lowest MAE value, around 445, is achieved by XGBoost. However, the computational time of this model is nearly three times greater than the hybrid ARIMA-based model, which has an MAE of over 610. The Prophet model is the least accurate and fastest.

5. Conclusions

As a result of this study, software capable of automated processing, tracking, and forecasting financial assets and risks was developed, utilizing hybrid ML models with ARIMA.

The research involved exploring models for time series analysis, specifically the proposed hybrid ARIMA, Prophet, and XGBoost regression models within the context of predicting expenses and profits based on a prepared dataset of user transactions over a 3-year period. MAE metrics for each model are presented to aid in selecting the best-performing model for further performance enhancement.

Considering the chosen algorithm, it is advisable to further calculate the weight values of factors and dynamically adjust the hyperparameters of the created hybrid ARIMA model to increase its productivity and accuracy. An optimized model can be used for more precise forecasting of user spending and profit risks.

It's worth noting that obtaining predictive values of financial risks, taking into account user behavioral factors within their investment strategy, enables a clearer and more predictable assessment of profits. Predicting expenses for stable profits is less relevant.

A subsequent step in algorithm research involves testing the model on different datasets, such as data from different users. Adding new features to the dataset, such as holidays or weather data, may improve the model's effectiveness and prediction results, enhancing its generalization capability.

References

- [1] C. Peng, Digital Inclusive Finance Data Mining and Model-Driven Analysis of the Impact of Urban-Rural Income Gap, *Wireless Communications and Mobile Computing* 7 (2022) 1-8. DOI:10.1155/2022/5820145.
- [2] N. Rudnichenko, V. Vychuzhanin, I. Petrov, D. Shibaev, Decision Support System for the Machine Learning Methods Selection in Big Data Mining, in: *Proceedings of The Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020)*, CEUR-WS, 2608, 2020, pp. 872-885.
- [3] V. Vychuzhanin, N. Rudnichenko, Z. Sagova, M. Smieszek, V. Cherniavskiy, A. Golovan, Analysis and structuring diagnostic large volume data of technical condition of complex equipment in transport, in: *24th Slovak-Polish International Scientific Conference on Machine Modelling and Simulations - MMS 2019*, Liptovský Ján, Slovakia, 2019. DOI:10.1088/1757-899X/776/1/012049
- [4] J. Bertomeu, Machine learning improves accounting: discussion, implementation and research opportunities. *Review of Accounting Studies* 25 3 (2020) 1135-1155. DOI:10.1007/s11142-020-09554-9
- [5] Y. Baştanlar, M. Ozuysal, Introduction to machine learning. *Methods in Molecular Biology (Clifton, N.J.)* 1107 (2014) 105-128. DOI: 10.1007/978-1-62703-748-8_7
- [6] J. Bertomeu, E. Cheynel, E. Floyd, W. Pan, Using machine learning to detect misstatements. *Review of Accounting Studies* 26 2 (2019) 468-519. DOI: 10.1007/s11142-020-09563-8
- [7] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, A.J. Aljaaf, A systematic review on supervised and unsupervised machine learning algorithms for data. *Supervised and Unsupervised Learning for Data Science, Unsupervised and Semi-supervised Learning*, Springer International Publishing, Cham (2014) 3-21. DOI: 0.1007/978-3-030-22475-2_1
- [8] D. Alex, C. Timothy, A regression based approach to short term load forecasting. *IEEE transaction on power systems* 5 4 (1990) 1535-1550. DOI: 10.1109/59.99410
- [9] I. Moghram, S. Rahman, Analysis evaluation of five short term load forecasting techniques. *IEEE transaction on power systems* 4 4 (1989) 1484-1491. DOI: 10.1109/59.41700
- [10] X.L. Dong, T. Rekatsinas, Data integration and machine learning: a natural synergy, in *ACM SIGMOD International Conference on Management of Data*, Houston, 2018, pp. 1645-1650. DOI: 10.1145/3183713.3197387
- [11] I. Ali, N. Bilal, F. Marwa, Forecasting Stock Prices with an Integrated Approach Combining ARIMA and Machine Learning Techniques ARIMAML. *Journal of Computer and Communications*. 11 (2023) 58-70. DOI:10.4236/jcc.2023.118005.
- [12] M. Alhomsy, H. Ahmed, Forecasting of ExchangeRate: Autoregressive modelsvs. XGBoost : thesis, 2020.
- [13] V. Arumugam, V. Natarajan, Time Series Modeling and Forecasting Using Autoregressive Integrated Moving Average and Seasonal Autoregressive Integrated Moving Average Models. *Instrumentation Measure Metrologie* 4 (2023) 161-168.
- [14] W. Charles, Jr. Chase, ARIMA Models. *Demand-Driven Forecasting* 13 (2013) 203-237.

Exploratory survey of access control paradigms and policy management engines

Pavlo Hlushchenko¹, Valerii Dudykevych¹

¹ Lviv Polytechnic National University, S. Bandery str. 12, Lviv, 79013, Ukraine

Abstract

This article presents an in-depth exploration of access control paradigms and policy management engines, aiming to provide insights into their functionalities, strengths, and limitations. The study examines three prominent access control paradigms: Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), and Relationship-Based Access Control (ReBAC), elucidating their principles and real-world applications. Furthermore, the article evaluates major policy management engines, including Google Zanzibar, AWS Cedar, and Open Policy Agent (OPA), delineating their architectures, purposes, and strengths. Through this comprehensive analysis, key observations emerge regarding the evolving landscape of access control mechanisms and the critical role of policy management engines in orchestrating access control policies. The findings underscore the importance of aligning access control mechanisms with organizational requirements and use case scenarios, while also highlighting avenues for further research and experimentation in exploring novel approaches to access control and policy management with direct or complementary use of AI.

Keywords

Access control, rbac, abac, rebac, open policy agent, aws cedar, google zanzibar, policy engines

1. Introduction

The ever-expanding digital landscape necessitates robust mechanisms to safeguard access to sensitive information and resources. Access control paradigms and policy management engines emerge as indispensable tools in this context, enabling organizations to establish fine-grained control over access privileges while mitigating security vulnerabilities and upholding regulatory compliance.

This article embarks on a comprehensive exploration of these critical technologies. It delves into the diverse landscape of access control paradigms, their historical context, and introduces prominent models like Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), and Relationship-Based Access Control (RBAC). Subsequently, the article examines the critical role of policy management engines in facilitating the implementation of these paradigms. It delves into the functionality and significance of these engines, showcasing specific examples such as Google Zanzibar, Open Policy Agent (OPA), and AWS Cedar.

Through a high-level comparative analysis of these paradigms and engines, this article aims to equip readers with a foundational understanding of this crucial facet of cybersecurity and the current state of things in the field. This knowledge empowers organizations to make informed decisions regarding the selection and implementation of solutions that best align with their specific security needs and operational requirements.

Implementing access control, particularly within a zero trust architecture, serves as a proactive defense against a variety of cyber threats, notably ransomware attacks [1]. By adhering to the principle of least privilege, access control limits unauthorized access to sensitive data, mitigating the impact of ransomware incidents. Additionally, robust access control measures, such as multi-factor authentication, bolster defenses against ransomware by restricting unauthorized entry and reducing the attack surface. In a zero trust framework, where trust is never assumed, continuous verification of user identities and device integrity reinforces security measures, safeguarding critical assets against ransomware and other cyber threats [2].

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ pavlo.k.hlushchenko@lpnu.ua (P. Hlushchenko); valerii.b.dudykevych@lpnu.ua (V. Dudykevych)

ORCID 0000-0002-1262-5484 (P. Hlushchenko); 0000-0001-8827-9920 (V. Dudykevych)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Access Control Paradigms

Access control paradigms serve as foundational frameworks for managing and enforcing authorization policies within systems and applications. These paradigms delineate the rules and principles governing access to resources based on various attributes such as user roles, attributes, or contextual information.

The development of access control paradigms has evolved over time, reflecting the growing complexity of systems and escalating security concerns. Early efforts focused on simple mechanisms like password-based authentication, which later progressed to more sophisticated models offering granular control and flexibility.

Some of the prominent access control paradigms include:

- **Discretionary Access Control (DAC):** This traditional model allows the owner of a resource to grant access permissions to other users. Emerging in the early days of computing, DAC was well-suited for smaller, less complex systems. However, its reliance on individual users to manage access permissions can lead to scalability challenges and potential security vulnerabilities in larger environments
- **Mandatory Access Control (MAC):** In contrast to DAC, MAC enforces centrally defined access control policies that are pre-determined and not modifiable by individual users. This model, often found in military and government systems, prioritizes security over flexibility. However, its centralized management can be less adaptable in dynamic and complex environments
- **Role-Based Access Control (RBAC):** This widely adopted model assigns users to pre-defined roles, with every role containing specific individual permissions. By grouping users who have similar access requirements, this method streamlines the management of access. RBAC emerged as a response to the limitations of DAC in larger organizations requiring a more structured approach to authorization [3]
- **Attribute-Based Access Control (ABAC):** This dynamic model grants access based on a combination of attributes associated with the user, the resource, the environment, and the requested action. This flexibility allows for fine-grained control and policy enforcement based on various contextual factors. ABAC emerged as a response to the increasing need for more granular and adaptable access control in complex systems with diverse access requirements
- **Relationship-Based Access Control (ReBAC):** This emerging model focuses on the relationships between entities (users, resources, etc.) when making authorization decisions. Access rights are granted based on established relationships, such as ownership, membership in groups, or hierarchical structures. ReBAC offers a dynamic and context-aware approach to access control, particularly relevant in social network and collaborative environments

These paradigms have evolved in response to the changing landscape of technology and security requirements. RBAC, initially introduced in the 1990s to address the complexities of access management in large organizations, laid the groundwork for subsequent advancements in access control methodologies. Its flexibility allows it to implement MAC and DAC. ABAC emerged as a response to the need for more dynamic and context-aware access control mechanisms, accommodating the intricacies of modern computing environments. ReBAC represents a specialized approach tailored to organizations where access decisions are heavily influenced by inter-entity relationships and dependencies. While these approaches excel at their use cases, they can also be used together to complement each other. In the subsequent sections, we will delve deeper into each access control paradigm, examining their principles, strengths, and limitations, to provide a thorough understanding of their role in shaping access control strategies.

2.1. Role-Based Access Control (RBAC)

RBAC is a widely adopted access control paradigm that revolves around the concept of an organization and the roles defined in it. Access rights in RBAC are assigned to roles, and roles are then assigned to users based on their respective job functions and responsibilities. This hierarchical approach simplifies access management and enhances security by following the principle of least privilege, where users are granted only the permissions they need for their job. To make the access control management workflow easier and more consistent, it is a best practice to create user groups, add users to the respective groups and then assign roles to the groups. This way the onboarding/offboarding process is straightforward as well as movement of a user between different teams or projects inside an organization.

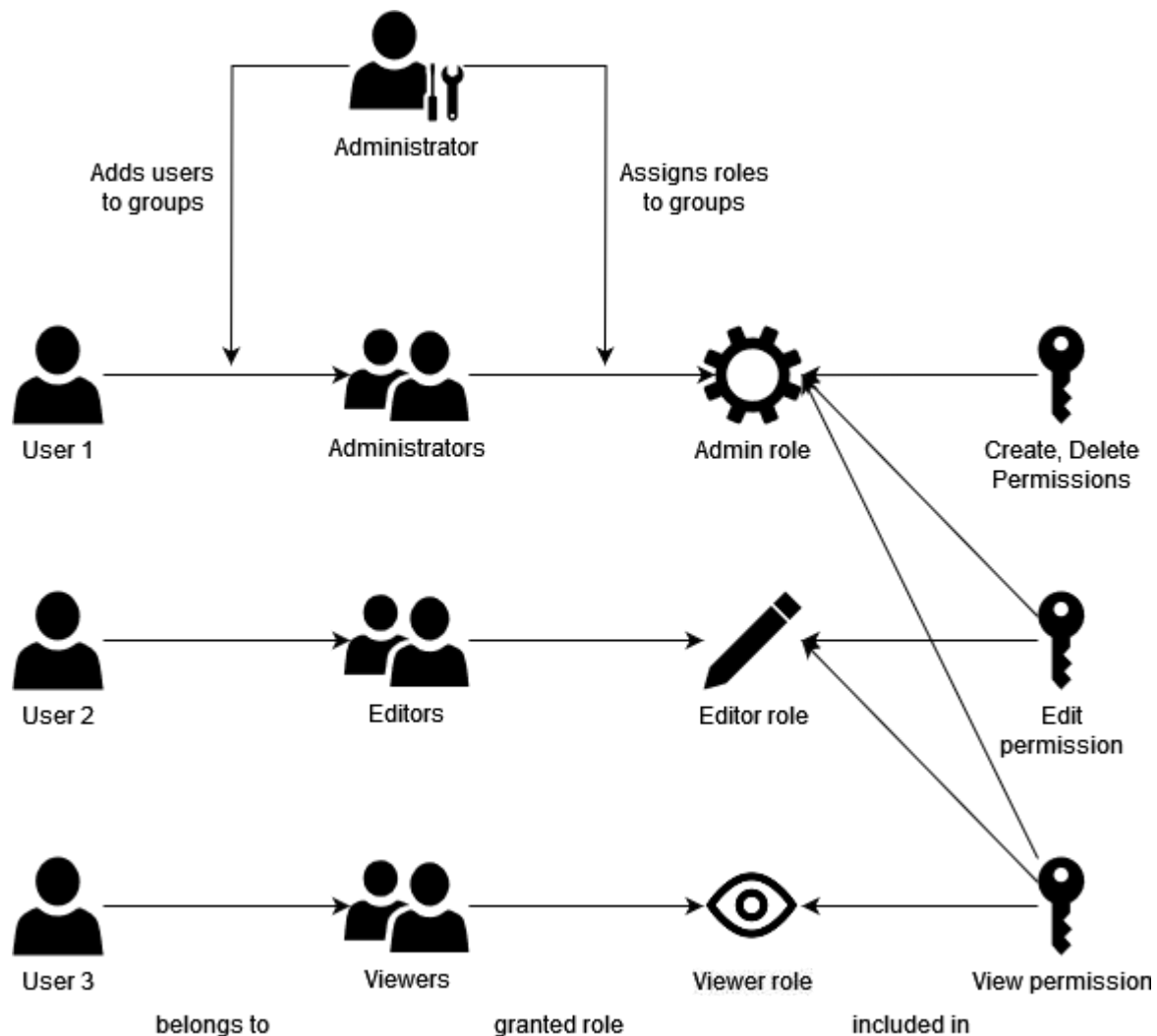


Figure 1: RBAC architecture

Key concepts in RBAC are:

- **Roles:** Represent a collection of permissions associated with a specific job function or responsibility within an organization. Examples include Administrator, Editor, Viewer, etc.
- **Users:** Individuals who are assigned to one or more roles based on their job responsibilities and access requirements.
- **Permissions:** Define the specific operations a user can perform on a resource, such as create, view, edit, delete, etc.
- **Resources:** Represent entities within the system that require access control, such as files, data, applications, services, etc.

RBAC model has several upsides that make it a preferred choice for access management within organizations. Firstly, its simplicity streamlines access control by organizing permissions into roles, significantly reducing administrative overhead and complexity. This structured approach also enhances scalability, allowing for seamless adaptation to organizational growth. With RBAC, new users can effortlessly be added to existing groups with pre-assigned roles, while the creation of new roles can be easily accommodated as organizational needs evolve. Furthermore, RBAC promotes consistency in access control policies throughout the organization, ensuring uniformity and minimizing the likelihood of errors or discrepancies. Lastly, RBAC enhances security by enforcing the principle of least privilege, thereby reducing the risk of unauthorized access and potential security breaches.

While RBAC offers significant benefits, it also presents several limitations that organizations should consider. The practical implementation of RBAC in real-world scenarios presents several challenges, notably the issue of role explosion. This occurs when the number of roles becomes excessively large and unmanageable, complicating the administration and efficiency of the access control system. It makes managing a multitude of roles challenging and can lead to a significant administrative overhead.

Aaron Elliott and S. Knight (2010) [4] highlight this problem and propose new approaches based on real-world practices to mitigate it. Secondly, role maintenance, including tasks such as role creation, modification, and deletion, can be time-consuming and resource-intensive, particularly in environments with high dynamics. Additionally, RBAC may exhibit limited flexibility in accommodating dynamic access control requirements or complex authorization scenarios involving contextual attributes or relationships. Lastly, there is a potential for abuse, as improperly defined roles or assigning users to inappropriate roles can introduce security vulnerabilities, underscoring the importance of careful role management practices.

Another significant challenge in RBAC implementation is adapting the system to context-sensitive environments, such as cloud and fog networks, where dynamic access control is crucial. A.S.M. Kayes et al. (2020) [5] discuss these context-aware access control mechanisms, noting the difficulties in applying traditional RBAC systems effectively in such rapidly changing settings. They suggest enhancements to address these issues, providing a detailed taxonomy of solutions based on real-world application case studies. These studies illustrate both the potential and the complexities of RBAC systems, emphasizing the need for continuous adaptation and innovation to overcome practical barriers in diverse operational contexts.

RBAC's simplicity, scalability, and security benefits have cemented its position as a foundational access control paradigm in many organizations and cloud environments. However, as computing environments evolve, organizations may supplement RBAC with more dynamic and context-aware access control mechanisms to address emerging challenges, requirements.

RBAC simplifies permission management by assigning permissions to roles rather than individuals, yet it faces specific challenges and potential vulnerabilities. The phenomenon of role explosion, where the number of roles grows unwieldy, can complicate the management and increase the risk of errors and security gaps. RBAC's static nature does not account for contextual factors, potentially leading to over-privileged accounts and security risks. Furthermore, RBAC may struggle with enforcing complex segregation of duties (SoD) policies, increasing the risk of fraud or misuse if roles are not meticulously defined. Administrative overhead is another concern, as managing a large number of roles and permissions can become labor-intensive and prone to delays, impacting both security and operational efficiency. Finally, scalability issues can arise as the system expands, potentially leading to inconsistencies in enforcing security policies across an organization.

Mitigating these vulnerabilities in RBAC involves regular audits of roles and permissions, implementing role hierarchies to streamline management, utilizing automated tools to reduce errors, and potentially integrating RBAC with dynamic models like ABAC or ReBAC to enhance flexibility. Such strategies help maintain security and efficiency in RBAC systems, even as organizational complexity grows.

2.2. Attribute-Based Access Control (ABAC)

Attribute-Based Access Control (ABAC) is a flexible, extensible and dynamic access control paradigm that grants or denies access based on a combination of attributes associated with various entities and the requested action. These attributes can include:

1. Subject attributes: Characteristics of the user, such as job title, department, location, security clearance, etc.
2. Object attributes: Characteristics of the resource being accessed, such as data classification level, creation date, location, etc.
3. Action attributes: Characteristics of the operation being requested, such as read, write, delete, etc.
4. Environment attributes: Contextual factors like time of day, network location, device type, etc.

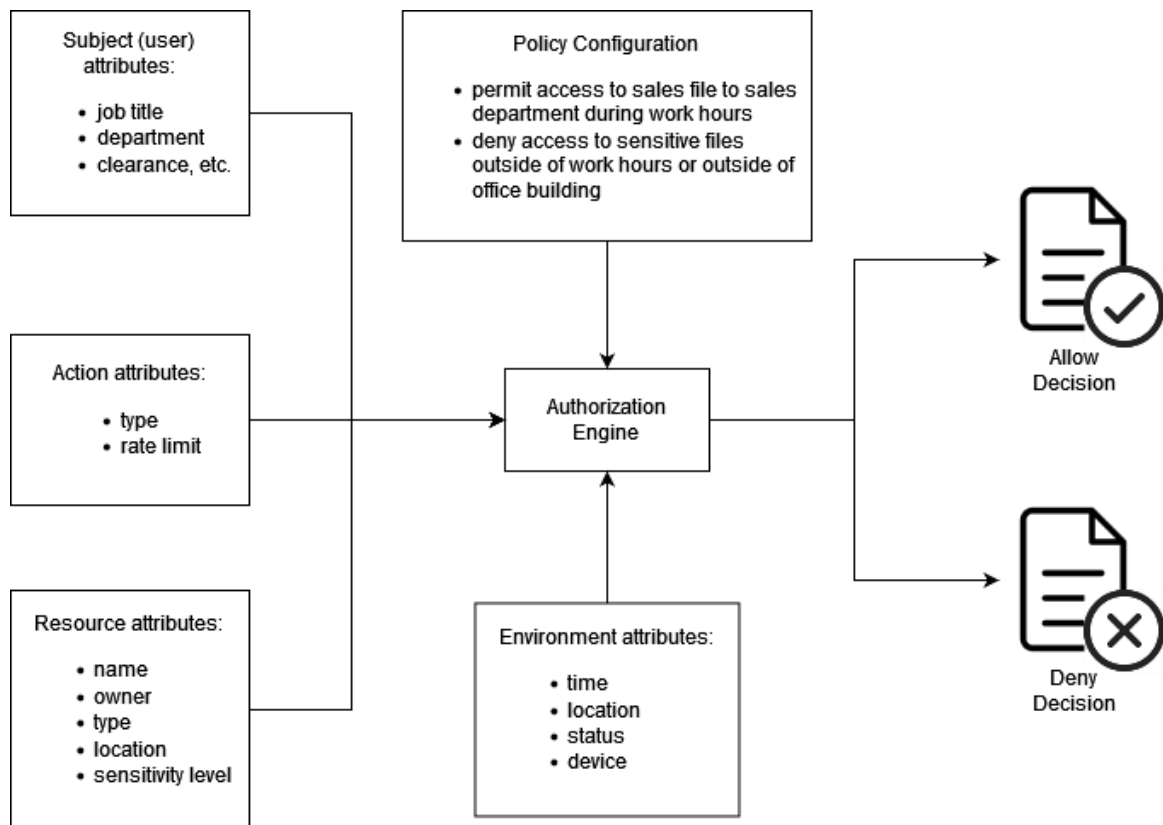


Figure 2: ABAC architecture

While the concept of ABAC existed for many years, it was endorsed by the Federal Chief Information Officers Council in 2011 as a model to adopt from considering growing complexity of systems and the need for more granular and adaptable access control [6] compared to traditional models like RBAC. While RBAC simplifies access management, it can be less flexible in dynamic environments where user and resource attributes are constantly changing as shown in Figure 2. The main concepts in ABAC model are:

- **Attributes:** Represent characteristics associated with users, resources, actions, and the environment.
- **Policies:** Define the access control rules based on the combination of relevant attributes. These policies dictate which combinations of attributes grant or deny access for specific actions on specific resources.
- **Policy Engine:** Analyzes the attributes of the user, resource, action, and environment to evaluate the applicable policies and make an access decision (grant or deny).

ABAC offers several strengths that make it a compelling approach to access control within organizations. Firstly, ABAC facilitates fine-grained control by allowing for highly granular access control based on various contextual factors, which are represented as attributes. This enables organizations to tailor access policies with precision, accommodating diverse scenarios and requirements. Secondly, the dynamic nature of ABAC enhances flexibility, as it enables the system to adapt to changing access requirements without the need to modify pre-defined roles. This adaptability simplifies the management of access control policies, particularly in dynamic environments where access needs may evolve rapidly. Lastly, ABAC can contribute to improved security by enabling the enforcement of more complex and nuanced access control policies. By considering multiple attributes and factors, ABAC can mitigate risks more effectively compared to simpler access control models, thereby enhancing overall security posture.

While ABAC offers significant advantages, it also presents certain concerns that organizations should consider. Firstly, managing and enforcing ABAC policies can be more complex compared to RBAC, primarily due to the dynamic nature of attributes and the requirement for robust policy engines. The dynamic nature of attributes introduces complexity in policy definition and enforcement, potentially leading to increased administrative overhead depending on the implementation. Secondly, evaluating complex ABAC policies can incur performance overhead compared to simpler access control models. The need to consider multiple attributes and factors during policy evaluation may result

in additional processing time, impacting system performance, particularly in high-throughput environments. Therefore, organizations implementing ABAC should carefully consider these potential challenges and ensure adequate resources and processes are in place to address them effectively.

The practical implementation of Attribute-Based Access Control (ABAC) in various real-world scenarios reveals several challenges, primarily associated with the complexity and dynamism of modern environments. ABAC systems, while flexible and powerful, often struggle with the nuanced requirements of fine-grained access control, particularly in complex infrastructures like cloud and fog networks, as well as IoT systems. A notable issue highlighted in studies such as by A.S.M. Kayes et al. [5] and S.C. Hubli and G.P. Potdar [7], is the management of attributes that dynamically change with context, complicating the enforcement of access policies across different scenarios.

Moreover, the integration of ABAC in environments like smart homes or military applications, as discussed by G. Goyal, P. Liu, and S. Sural [8], demonstrates challenges related to scalability and performance, where the systems must efficiently manage a vast number of attributes without compromising on security or operational speed. These challenges are compounded in settings that require robust security measures against evolving threats and complex, frequently changing access rules. The research by S. Parkinson and S. Khan [9] further illuminates these issues, pointing out that real-world applications of ABAC still face significant hurdles in terms of policy management and the real-time processing of attributes, emphasizing the need for ongoing development to enhance the practicality and effectiveness of ABAC systems.

ABAC offers flexibility and granularity in managing access based on attributes, but it also comes with potential vulnerabilities. The complexity of managing a large number of detailed policies can lead to misconfigurations and security loopholes, while the performance impact of dynamic attribute evaluation can affect system efficiency and scalability. Additionally, the integrity of the attributes is crucial; any compromise can lead to incorrect access decisions, potentially allowing unauthorized access and data breaches. ABAC systems are also susceptible to insider threats, as knowledgeable insiders could manipulate attributes or exploit policy gaps. Furthermore, ensuring compliance with regulatory requirements can be complex due to the dynamic nature of ABAC policies, posing additional challenges in proving consistent compliance across all scenarios.

To mitigate these vulnerabilities, organizations should develop robust policies, perform rigorous testing, ensure accurate and secure attribute management and verification, optimize performance, and regularly monitor and review ABAC systems. These measures help maintain the system's integrity and compliance while harnessing ABAC's potential for dynamic and flexible access control.

Overall, ABAC offers a powerful and versatile approach to access control, especially in complex environments and scenarios requiring fine-grained control and dynamic rules [10]. However, the increased complexity requires careful consideration of its implementation and ongoing management.

2.3. Relationship-Based Access Control (ReBAC)

Relationship-Based Access Control (ReBAC) is an access control paradigm that considers the relationships between entities within an organization when making access control decisions. Unlike traditional access control models that primarily focus on user roles or attributes, ReBAC takes into account the relationships between users, resources, and other entities to determine access permissions [11]. Key concepts and components of ReBAC include:

- **Relationship Definition:** ReBAC begins with the identification and definition of relationships between entities within an organization. These relationships can include hierarchical relationships (e.g., supervisor-subordinate), team-based relationships (e.g., project team membership), or custom-defined relationships based on organizational structure or business processes.
- **Relationship-Based Policies:** Access control policies in ReBAC are defined based on the relationships between entities. These policies specify the conditions under which access is granted or denied based on the nature and characteristics of the relationships between users, resources, and other entities.
- **Contextual Access Control:** ReBAC incorporates contextual information and relationship dynamics into access control decisions. Access permissions may vary based on the specific relationships between entities, the nature of the access request, and the contextual environment in which the request is made.
- **Fine-Grained Control:** ReBAC offers fine-grained control over access permissions based on relationship attributes and characteristics. Organizations can define complex access control policies

that reflect the nuances of relationships within the organization, enabling precise control over access to resources.

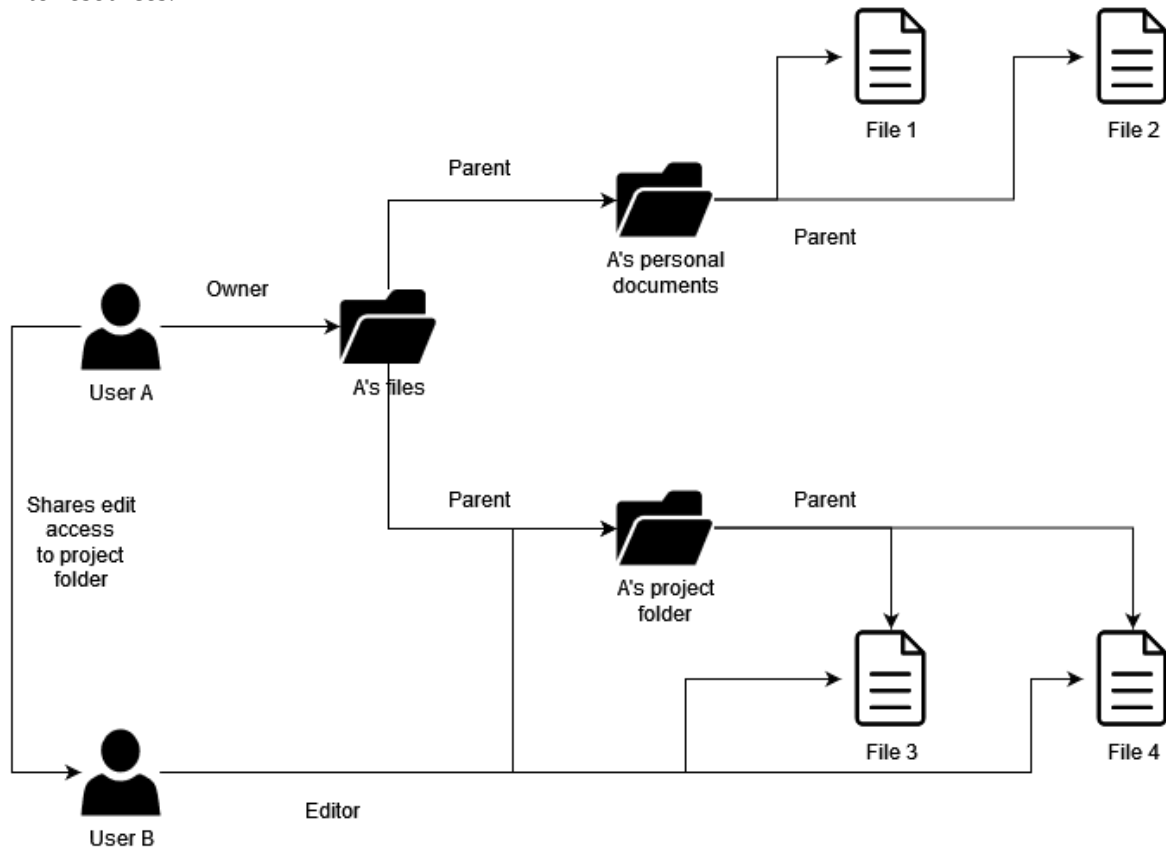


Figure 3: ReBAC architecture

ReBAC boasts several strengths that make it a formidable approach to access control within organizations. Firstly, it offers granular and comprehensive access control by providing precise control over access permissions based on the relationships between entities. This capability enables organizations to enforce tailored access control policies aligned with specific relationship dynamics, enhancing security and compliance. Secondly, ReBAC incorporates context awareness into access control decisions, allowing organizations to consider relationship dynamics, organizational structure, and business processes when granting or denying access. This contextual understanding enhances the accuracy and relevance of access control decisions, bolstering overall security posture. Additionally, ReBAC exhibits adaptability to changing organizational structures and relationship dynamics, enabling organizations to adjust access control policies dynamically as relationships evolve over time. This flexibility ensures that access control remains aligned with organizational needs and evolving business requirements. Moreover, ReBAC excels in environments where relationships play a significant role in access control, such as social media platforms or collaborative workspaces, providing improved control and security in such contexts. Lastly, ReBAC policies offer flexibility in capturing various types of relationships, allowing organizations to tailor access control based on specific contexts and requirements, further enhancing precision and effectiveness in access control enforcement.

Despite the notable strengths and a completely different approach to access control, ReBAC also presents several limitations that organizations should consider. Firstly, implementing ReBAC can be complex, especially in organizations with intricate relationship structures and dynamic relationship dynamics. Sophisticated policy management and enforcement mechanisms may be necessary to effectively implement ReBAC, adding complexity to the access control framework. Secondly, improperly defined relationships or policies can lead to unforeseen access control outcomes, potentially compromising security and compliance. Additionally, ReBAC requires capturing and representing various types of relationships, which may pose challenges in accurately modeling complex relationship dynamics. Furthermore, relationship management, including capturing and managing relationship data and ensuring consistency across systems, can be challenging, particularly in distributed or heterogeneous environments. Lastly, the dynamic evaluation of relationship-based policies in real-time may introduce performance overhead, particularly in high-throughput environments or systems with

large numbers of relationships, impacting system performance and responsiveness. Therefore, organizations considering the adoption of ReBAC should carefully assess these limitations and ensure appropriate strategies are in place to address them effectively.

The practical implementation of Relationship-Based Access Control (ReBAC) in real-world scenarios encounters several significant challenges, primarily arising from the complexity and dynamic nature of the relationships it needs to manage. One of the primary difficulties is the policy definition complexity, as highlighted by C. Arora (2022) [12]. This study discusses the challenges such as solving homomorphism problems that emerge when defining and enforcing relationships within access control frameworks, especially in settings where relationships can dynamically change, thus complicating the access control process.

Moreover, the scalability of ReBAC in distributed systems like IoT, blockchain, and cloud environments presents another layer of complexity. L. Golightly et al. (2023) [13] explore these challenges in the context of IoT, emphasizing issues related to policy management, enforcement efficiency, and the system's ability to handle a vast network of dynamic relationships without performance degradation. Similarly, ASM Kayes et al. (2020) [5] point out the difficulty in expressing and managing access control policies that effectively capture the fluid nature of user relationships and roles within decentralized networks such as cloud and fog computing environments. These environments require a more flexible and scalable approach to access control, which current ReBAC implementations struggle to provide.

These studies collectively indicate that while ReBAC is theoretically robust in utilizing relationships for access control, its practical application is often hindered by issues related to complexity in policy formulation, scalability, and the dynamic management of relationships in high-demand environments.

Managing complex and dynamic relationships within large organizations can become cumbersome and error-prone, leading to potential misconfigurations and security breaches. Additionally, ReBAC systems face scalability challenges as they must update access rights in real-time based on changing relationships, which can strain system resources and lead to latency issues.

The specification and enforcement of ReBAC policies are complex due to the intricate, multi-layered nature of relationships and their context dependency. Any inaccuracies or manipulations in relationship data can compromise access control decisions, leading to unauthorized access and increasing the risk of insider threats. Furthermore, the fluidity of ReBAC makes auditing and compliance particularly challenging, as demonstrating consistent application of access controls in a continually adapting environment can be difficult. To mitigate these vulnerabilities, organizations should implement robust management tools for relationships, sophisticated policy evaluation engines, and extensive auditing features to maintain continuous integrity and compliance in the ReBAC system.

As shown in Figure 3, User A is assigned the role Owner of Files folder, which means he is also the owner of every file and folder for which Files is the parent folder. ReBAC's focus on relationships between entities within an organization offers a unique perspective on access control, enabling organizations to enforce fine-grained access control policies that reflect the nuances of relationship dynamics. By considering relationships alongside user roles and attributes, ReBAC enhances security and enables organizations to enforce more context-aware and adaptable access control policies.

3. Overview of Major Policy Management Engines

In the realm of access control and authorization, policy management engines play a crucial role in enforcing access control policies and facilitating the implementation of various access control paradigms. A policy management engine can be defined as a software component or system that is responsible for the administration, enforcement, and evaluation of access control policies within an organization's IT infrastructure.

The primary purpose of policy management engines is to provide a centralized platform for defining, managing, and enforcing access control policies across diverse computing environments. These engines enable organizations to translate high-level security requirements and access control policies into actionable rules and mechanisms that govern access to resources.

Policy management engines facilitate the implementation of access control paradigms such as RBAC, ABAC, and ReBAC by providing the necessary infrastructure for policy definition, evaluation, and enforcement.

Policy management engines have evolved in response to the growing complexity of IT environments, the proliferation of access control paradigms, and the increasing demands for fine-grained access control. Initially, access control policies were often implemented through ad-hoc mechanisms embedded within individual applications or systems, leading to fragmented and inconsistent access control practices.

As organizations recognized the need for centralized policy management and enforcement, dedicated policy management engines began to emerge. These engines provided standardized interfaces and tools for defining, managing, and enforcing access control policies across heterogeneous IT environments.

Over time, policy management engines have evolved to incorporate features such as policy modeling languages, policy evaluation engines, and integration with identity and access management (IAM) systems. Modern policy management engines leverage technologies such as machine learning and artificial intelligence to enhance policy enforcement and adaptability to changing access control requirements.

Policy management engines are utilized across various industries and sectors, including healthcare, finance, government, and e-commerce, to enforce access control policies and ensure compliance with regulatory requirements. These engines are often integrated with existing IT infrastructure, including directory services, authentication systems, and application servers, to provide seamless access control enforcement across the organization.

The future of policy management engines is likely to be shaped by advancements in areas such as cloud computing, AI, Internet of Things (IoT), and edge computing. As organizations embrace distributed and hybrid IT environments, policy management engines will need to evolve to support dynamic and scalable access control mechanisms that span across disparate computing platforms and environments.

Additionally, the convergence of access control, identity management, and security analytics is expected to drive the development of more intelligent and context-aware policy management engines that can adapt to evolving threat landscapes and user behavior patterns. As organizations grow, get more complex and dynamic and operate at ever-increasing scale, AI is going to be a key part of such systems in the future.

3.1. Open Policy Agent (OPA)

Open Policy Agent (OPA) emerged as a notable solution for policy-based authorization in the context of modern, cloud-native architectures and microservices. Developed against the backdrop of evolving cloud-native technologies and the challenges inherent in managing access control in distributed systems, OPA embodies a departure from traditional access control mechanisms towards a more declarative and flexible approach. [14]

The genesis of OPA lies in the recognition of the limitations of existing access control mechanisms in addressing the complexities of cloud-native environments. As organizations embraced microservice architectures and containerization, the need for a unified and adaptable policy enforcement solution became apparent. OPA's development represents a response to this growing demand for a policy engine capable of addressing the dynamic and heterogeneous nature of modern computing environments.

Advantages of leveraging OPA include its utilization of Policy-as-Code (PAC), where policies are defined through Rego code, providing all the benefits associated with code, such as versioning, reviews, and auditing. OPA also supports fine-grained permissions, accommodating RBAC, ABAC, and ReBAC (with certain adjustments), thus enabling granular permissions management. Furthermore, OPA offers a high degree of flexibility and extensibility, empowering organizations to tailor access control policies to their specific requirements and use cases. With a robust open-source community and widespread adoption across various industries, OPA benefits from continuous development and contributions, ensuring ongoing enhancements and improvements. Moreover, OPA's architecture is engineered for scalability and performance, leveraging in-memory caching to scale efficiently and deliver optimal performance.

Using OPA may also entail certain disadvantages. Firstly, there is a learning curve associated with mastering the Rego policy language and understanding OPA's functionalities, which may be challenging for users unfamiliar with declarative policy languages and policy-based authorization concepts. Secondly, operational complexity may arise when managing and maintaining OPA instances in production environments, especially in distributed or heterogeneous environments with multiple

OPA instances and policy repositories. This complexity can pose challenges in ensuring consistency and efficiency across OPA deployments, requiring careful planning and management to mitigate potential issues.

Additional factors to consider include the complexity associated with real-time policy updates, the reliance on multiple data sources in rules, which often requires bundling not easily achieved, and the synchronization challenges of multiple OPA instances when operating more than one concurrently.

While OPA alone does not directly address these challenges, they can be effectively resolved by leveraging OPAL [15] in conjunction with OPA. OPAL serves as an administrative layer atop OPA, facilitating real-time updates and offering solutions to address synchronization issues.

In conclusion, OPA is a well-established and broadly used solution that provides substantial benefits including versatility, separation of code and policy, ability to integrate with various systems, and policy evaluation done in an efficient manner. When enhanced with OPAL, OPA becomes even stronger, successfully tackling further complexities found in intricate access control settings.

3.2. AWS Cedar

Cedar, recently released by AWS in open-source, is a policy-as-code language that is designed to enhance management of IAM and access control processes. It offers a well-organized and scalable approach to permissions management, particularly impactful for application-level permissions management.

Cedar presents several advantages that contribute to its effectiveness in access control management within organizations. Firstly, akin to OPA, Cedar adopts a policy-as-code paradigm, employing a policy language for defining access control policies. This approach facilitates the treatment of policies as code, enabling versioning, auditing, and transparent review processes, thereby enhancing policy management practices. Secondly, Cedar prioritizes readability in its design, ensuring accessibility for both technical and non-technical team members. This emphasis on readability fosters collaboration and understanding across diverse teams within an organization. Additionally, Cedar adopts a structured and safety-oriented design approach [16], emphasizing safety by default. This commitment to security and correctness provides confidence through development practices guided by verification. Furthermore, Cedar is specifically designed for application-level authorization, catering to the unique access control requirements within applications [17]. Its design ensures effective enforcement of permissions at the application level, thereby enhancing security and compliance. Lastly, Cedar supports a variety of access control paradigms, including RBAC, ABAC, and ReBAC, enabling granular permissions management to accommodate diverse use cases and access control requirements effectively.

Although Cedar boasts significant benefits, it also entails certain drawbacks that organizations need to take into account. Firstly, Cedar currently lacks an extensive ecosystem of tools and modules compared to more established policy languages. This constraint could require extra work or custom solutions for particular tasks, which may affect the efficiency of policy management workflows. Additionally, given that it is a newly introduced policy language, Cedar boasts an active yet smaller community compared to, for instance, OPA. This smaller community might lead to less documentation, resources, and support available from community channels, which could present challenges for organizations needing help with Cedar-related initiatives. Therefore, organizations considering the adoption of Cedar should carefully assess these limitations and ensure appropriate strategies are in place to address them effectively.

3.3. Google Zanzibar

Google Zanzibar represents a pivotal advancement in access control systems, developed by Google to address the complex access control requirements of its distributed services and platforms. Originating from Google's internal needs for a scalable, centralized access control solution, Zanzibar was designed to meet the challenges posed by Google's vast and diverse ecosystem of services, users, and resources.

Zanzibar possibly draws its name from the historic island known for its significance as a trading hub, reflecting its role as a central hub for managing access control policies across Google's services and infrastructure. Historically, Google utilized a decentralized approach to access control, where each service managed its own access policies. However, as the company's services and user base grew exponentially, this decentralized model became increasingly challenging to manage and scale effectively.

Google Zanzibar represents a departure from this decentralized model, offering a centralized, globally-distributed access control system capable of managing access policies for millions of resources and users across Google's infrastructure. By consolidating access control policies into a unified system, Zanzibar enables Google to enforce consistent, fine-grained access control across its services, while also providing scalability, fault tolerance, and adaptability to changing access requirements [18].

Google Zanzibar has several advantages that contribute to its effectiveness in access control management within large-scale distributed environments. Firstly, it offers centralized access control, providing a unified platform for managing access control policies. This centralized approach enables consistent enforcement of access policies across distributed services and platforms, enhancing security and compliance. Secondly, Zanzibar is designed to scale seamlessly to handle millions of policy evaluations per second, catering to Google's vast ecosystem of services and users. This scalability ensures that access control remains efficient and responsive even as the organization grows. Additionally, Zanzibar employs a globally-distributed architecture with built-in fault tolerance mechanisms, ensuring high availability and reliability of access control services. This fault tolerance enhances system resilience and minimizes the risk of service disruptions. Lastly, Zanzibar enables fine-grained access control, allowing Google to define and enforce access policies based on diverse attributes, relationships and conditions. This granularity enables precise control over access permissions, enhancing security and enabling organizations to tailor access policies to specific requirements and use cases.

On the other hand, Zanzibar has several drawbacks for consideration. Firstly, Zanzibar is a proprietary system developed by Google for internal use, which limits its accessibility and interoperability with non-Google systems. This proprietary nature may hinder organizations seeking to integrate Zanzibar into their own infrastructure or collaborate with external partners. Secondly, external documentation and insights into Zanzibar's architecture and implementation details are limited, making it challenging for organizations to gain a comprehensive understanding of the system and its capabilities. This limited documentation may impede broader adoption outside of Google's ecosystem. Lastly, organizations leveraging Zanzibar may face vendor lock-in, as the system is tightly integrated with Google's internal infrastructure and services. This dependency on Google's ecosystem may restrict organizations' flexibility and ability to migrate to alternative solutions in the future. Therefore, organizations considering the adoption of Zanzibar should carefully evaluate these limitations and weigh them against the system's advantages.

Implementing a graph based on Google Zanzibar into the cloud environment adds complexity, frequently depending on hosted services. This reliance can raise concerns regarding latency and scalability, as dependencies on external services may impact system performance and introduce additional points of failure. Moreover, graph-based systems such as Zanzibar might encounter deployment challenges at the edge owing to their large size and non-local characteristics. These limitations could hinder their effectiveness in edge computing environments, where low latency and efficient access control are essential.

Compared to other policy management engines, Google Zanzibar distinguishes itself through its centralized, globally-distributed architecture tailored for large-scale, distributed services. While OPA offers flexibility and extensibility through its open-source nature, and AWS Cedar provides seamless integration with AWS services, Zanzibar excels in scalability, fault tolerance, and fine-grained access control tailored for Google's unique infrastructure and requirements.

Google Zanzibar represents a groundbreaking approach to access control, addressing the scalability and complexity challenges inherent in managing access policies for large-scale, distributed systems. While proprietary in nature, Zanzibar's design principles and architectural innovations offer valuable insights and lessons for the broader community in developing scalable and resilient access control solutions for modern computing environments. Many other authorization engines were inspired by Zanzibar, such as SpiceDB [19].

3.4. Integration with existing infrastructure

Integrating RBAC, ABAC, and ReBAC within existing IT infrastructures involves adapting these access control models to meet the complex requirements of modern technological environments. The study by M. Penelova (2021) [20] discusses how these models can be applied to enhance security in enterprise software systems. RBAC's role-based restrictions, ABAC's attribute-based flexibility, and ReBAC's relationship-oriented permissions each offer unique advantages depending on the specific

needs of an organization. The integration often requires careful planning to align with existing security policies and IT architecture while ensuring scalability and adaptability to future changes. Integration involves mapping organizational roles to access permissions, ensuring that these roles reflect actual job functions and access needs within the IT infrastructure. RBAC should be implemented with a central administration point that manages role assignments and permissions, facilitating updates and maintenance. For ABAC, the focus is on defining attributes (such as department, time of access, and type of data accessed) that control access decisions. Integration requires setting up a policy enforcement point that dynamically evaluates attributes against policies before granting access. This might involve modifying existing data schemas to include necessary attribute data. ReBAC requires identifying and managing relationships between entities (like user-to-user or user-to-data relationships) within the IT system. Integration can be complex, involving the creation of a relationship graph that must be maintained and referenced by the access control system. Existing applications might need to be extended to support these access control models or, in case it is not possible, a middleware application can be implemented to handle access control.

The survey by L. Golightly et al. (2023) [13] extends the discussion to distributed systems such as cloud services, blockchain, IoT, and Software Defined Networks (SDN). These environments pose additional challenges due to their decentralized nature and the dynamic interactions between their components. The study highlights how RBAC, ABAC, and ReBAC need to be adapted to manage access control efficiently in these settings, focusing on the need for models that can dynamically adjust to changes in user roles, attributes, and relationships without compromising the security or performance of the system. This could involve automated tools to monitor changes in the operational environment and adjust access controls accordingly. Additionally, in distributed systems, central management of access controls might not be feasible. Instead, a distributed approach where each node or service manages its own access controls based on synchronized policies could be more effective.

Furthermore, J. Park et al. (2021) [21] explore the application of these access control models in smart environments, where the integration of IoT and cloud computing introduces new dimensions of complexity. Their work discusses the principles of next-generation access control systems that not only secure but also enhance the functionality and user collaboration in smart infrastructures. The integration of RBAC, ABAC, and ReBAC in these systems is shown to be critical in managing diverse and dynamically changing access requirements while maintaining a high level of security. Access control systems need to be integrated with cloud-based services and IoT devices. This includes the ability to handle the high volume of access requests typical in these environments and ensuring that access control decisions are made swiftly to avoid delays in service. In smart environments, access control systems need to manage not only user access but also device interactions. This requires policies that can handle complex scenarios where devices act on behalf of users or interact with other devices autonomously.

3.5. Summary

After examining three prominent policy management engines, namely Google Zanzibar, Open Policy Agent (OPA), and AWS Cedar, each offers distinct features and functionalities tailored to specific use cases and organizational requirements.

Google Zanzibar stands out for its centralized access control system, designed for large-scale, distributed services. Its strengths lie in scalability and fault tolerance, making it ideal for organizations operating expansive, complex ecosystems where centralized access control is paramount. However, its proprietary nature and limited external documentation may pose challenges for organizations seeking interoperability and comprehensive understanding outside of Google's infrastructure.

Open Policy Agent (OPA) offers unparalleled flexibility and extensibility, empowering organizations to define and enforce access control policies across diverse cloud-native environments. Its open-source nature, coupled with a vibrant community, makes it suitable for organizations prioritizing flexibility and community-driven development. OPA excels in scenarios where fine-grained control and policy-as-code capabilities are essential, such as Kubernetes-based environments and microservices architectures.

AWS Cedar, recently released as an open-source solution by AWS, introduces a structured approach to IAM management and access control. It offers a seamless integration with AWS services and prioritizes readability and safety in policy enforcement. Cedar is well-suited for organizations heavily invested in the AWS ecosystem, particularly those requiring application-level authorization and compliance management within AWS environments. Having said that, it is not entirely AWS specific.

The project is currently available in open source as a collection of repositories and can be self-hosted using cedar-agent.

In summary, the choice of policy management engine should align with organizational requirements, system architecture, and cloud platform preferences as described in Table 1. Google Zanzibar is recommended for organizations with extensive, distributed infrastructures requiring centralized access control. OPA is suitable for organizations seeking flexibility, extensibility, and community support in managing access control policies across diverse cloud-native environments. AWS Cedar is ideal for organizations heavily reliant on AWS services, prioritizing readability and structured IAM management within AWS environments. Careful consideration of each engine's strengths and limitations is essential to selecting the most suitable solution for specific use cases and organizational needs.

Table 1
Policy management engines comparison table

Key Areas	Open Policy Agent	AWS Cedar	Google Zanzibar
Open Source	Yes	Yes	No (but alternative implementations are available)
Ability to self-host	Yes	Yes	No
Supported access control paradigms	RBAC, ABAC, ReBAC (with additional coding)	RBAC, ABAC, ReBAC	ReBAC
Language	Rego (Go-based)	Cedar (Rust-based)	ZCL
Authorization type	Policy as code (PAC)	Policy as code (PAC)	Graph-based
Scale	Scalable (with OPAL)	Scalable	Huge
Data volume	Lower	Lower	Higher
Performance	Higher	Higher	Lower compared to policy as code
Policy complexity	Higher	Higher	Lower
Deployment at edge	Supported	Supported	Infeasible due to size
Advantages	Policy as code, performance, adoption	Policy as code, application level policies, readability	Best fit for hierarchies and nested relationships, scale
Disadvantages	Learning curve, hard to use in production without OPAL	Limited tooling, community size, adoption	Cannot be deployed locally due to size, ReBAC only
Ease of Updates	Highly flexible (with OPAL)	Highly flexible	Less flexible
Deployment	Standalone	Standalone or as part of AWS services	Not available as a standalone deployment or service, although open source implementations exist

4. AI Applications in access control space

The application of Artificial Intelligence in the access control space holds significant promise for revolutionizing how organizations manage and enforce access policies within their digital ecosystems. Traditional access control mechanisms, while effective, often struggle to adapt to the increasingly dynamic and complex nature of modern computing environments. As organizations embrace cloud computing, microservices architectures, and distributed systems, the need for adaptive, intelligent access control solutions becomes more pronounced. AI can be leveraged in various ways for access control and policy management.

Some potential use cases include:

1. **User and Entity Behavior Analytics (UEBA):** AI can be used to establish baseline behavior patterns for users, devices, and applications within an organization. By continuously monitoring and analyzing activity logs, AI algorithms can detect anomalous behavior that deviates from the established baselines, potentially indicating unauthorized access attempts or policy violations. This can help identify insider threats, compromised accounts, or malicious activities.
2. **Automated Policy Management:** AI can assist in the creation, review, and maintenance of access control policies. It can analyze existing policies, identify inconsistencies, redundancies, or gaps, and suggest improvements. AI can also be used to automatically generate and update policies based on organizational requirements, industry regulations, and best practices.
3. **Intelligent Access Governance:** AI can support access governance by continuously evaluating user access privileges and entitlements against defined policies. It can identify excessive or unnecessary permissions, recommend access revocations or modifications, and automate the review and certification processes for user access rights.
4. **Risk Analysis and Mitigation:** AI can assess the potential risks associated with granting or denying access to specific resources or systems. By analyzing various factors such as user roles, resource sensitivity, threat intelligence, and historical data, AI can provide risk scores and recommendations to help organizations make informed access control decisions.
5. **DLBAC (Deep Learning Based Access Control)** implementation can augment, complement and in future possibly even replace the currently used access control approaches in large scale, complex and dynamic environments [22]. It involves a neural network that adapts to the environment and context and reduces the effort spent on policy engineering, changes and other related maintenance activities.
6. **Continuous Monitoring and Adaptation:** AI can continuously monitor changes in the organization's infrastructure, applications, and user population. It can proactively identify potential policy violations or access control issues and suggest appropriate remediation measures or policy updates to ensure ongoing compliance and security.
7. **Intelligent Multi-Factor Authentication (MFA):** AI can be used to enhance MFA mechanisms by analyzing contextual factors such as user behavior, device characteristics, location, and risk indicators. It can dynamically adjust the authentication requirements or challenge types based on the perceived risk level, providing an additional layer of adaptive access control.
8. **Natural Language Processing (NLP) for Policy Interpretation:** AI-powered NLP can assist in interpreting and understanding access control policies written in natural language. It can translate policies into machine-readable formats, identify ambiguities or inconsistencies, and facilitate better understanding and enforcement of policies across the organization.

The integration of AI technologies into access control systems has the potential to enhance security, streamline access management processes, and improve user experience in today's increasingly interconnected and dynamic digital landscape. By addressing challenges such as dynamic policy management, anomaly detection, and adaptive access control, AI-driven access control solutions can help organizations stay ahead of evolving security threats and compliance requirements.

5. Conclusions

In this exploratory survey, we delved into various access control paradigms and policy management engines, aiming to provide insights into their functionalities, strengths, and limitations. Through the examination of Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), and Relationship-Based Access Control (ReBAC), alongside major policy management engines such as Open Policy Agent (OPA), AWS Cedar, and Google Zanzibar, several key observations and considerations have emerged.

Firstly, it is evident that access control paradigms have evolved significantly to address the diverse and dynamic requirements of modern computing environments. While RBAC offers simplicity and scalability, ABAC provides flexibility and granularity, and ReBAC introduces a nuanced perspective by considering relationships between entities within an organization. Each paradigm has its own set of strengths and limitations, necessitating careful consideration of organizational needs and use case requirements in selecting the most suitable approach.

Furthermore, policy management engines play a pivotal role in facilitating the implementation and enforcement of access control policies across distributed and heterogeneous IT environments. Open

Policy Agent (OPA), AWS Cedar, and Google Zanzibar offer distinct approaches to policy management, catering to different deployment scenarios and integration requirements. While OPA provides an open-source, flexible solution with community support, AWS Cedar and Google Zanzibar offer managed services tailored for integration with their respective cloud platforms.

In the realm of authorization systems, there exists universal solution to cater to the diverse range of scenarios. The decision to employ graph-based systems, such as Google Zanzibar or SpiceDB, or policy-as-code systems like Open Policy Agent or AWS Cedar, is not a matter of selecting a technologically superior approach. Instead, it hinges on aligning the chosen system with the specific requirements, scale, and complexity of the environment in question.

The choice of access control approach and policy management engine should be guided by factors such as organizational requirements, system architecture, and compliance considerations. Organizations must carefully assess the trade-offs between simplicity, flexibility, scalability, and vendor lock-in when selecting access control mechanisms and policy management solutions.

Graph-based systems demonstrate proficiency in managing voluminous data sets and effectively representing intricate relationships within large-scale applications. Conversely, policy-as-code systems offer a high degree of flexibility, facilitating seamless policy updates and exhibiting adeptness in handling complex policy structures. Each paradigm possesses its unique strengths and limitations, necessitating a judicious evaluation of system requirements and a careful weighing of trade-offs.

Moreover, it is imperative to acknowledge the fact that these systems can complement each other. By combining their respective capabilities, a hybrid system can be crafted, capitalizing on the strengths of both approaches and thereby addressing the diverse authorization needs that may arise. Ultimately, the overarching objective remains the effective control of access to services, ensuring data privacy, and upholding system integrity, irrespective of the chosen authorization strategy.

Areas for further research and experimentation include exploring the intersection of access control paradigms with emerging technologies such as blockchain, AI, and zero-trust security architectures, investigating the impact of dynamic policy enforcement mechanisms on system performance and scalability, and exploring novel approaches to policy modeling and enforcement in distributed and decentralized computing environments.

Future research could focus on the development of AI-driven models for predictive access control, where machine learning algorithms could predict and prevent unauthorized access based on behavior patterns and anomaly detection. This approach could significantly improve security in real-time applications and large-scale systems by allowing more proactive management of access permissions. Investigating the integration of AI to automate and optimize the management of complex access control policies would also be valuable, particularly in environments where access requirements are continually changing.

Blockchain technology could be explored further for its potential to facilitate decentralized access control mechanisms, particularly in distributed environments like the Internet of Things (IoT) and cloud computing. The immutable and transparent nature of blockchain could help in creating more secure and resistant access control systems that are less susceptible to tampering and insider threats. Research could also look into hybrid models that integrate blockchain with traditional access control systems to enhance trust and security in multi-party transactions or operations.

There is also a compelling case for advancing policy management engines to better handle the complexity and scale of modern IT environments. Enhancing the capability of these systems to dynamically adjust policies based on real-time data and contextual information could address many of the current limitations in scalability and flexibility. Additionally, exploring ways to simplify policy management while maintaining robust security controls in highly dynamic environments would be beneficial.

In conclusion, as the landscape of access control continues to evolve in response to emerging technologies and evolving threat landscapes, ongoing research and experimentation are essential to develop robust, adaptable, and scalable access control mechanisms that meet the evolving needs of organizations in an increasingly interconnected and dynamic digital ecosystem.

References

- [1] D. Zhuravchak, T. Ustyianovych, V. Dudykevych, B. Venny and K. Ruda, Ransomware Prevention System Design based on File Symbolic Linking Honeypots, in: Proceedings of the 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced

- Computing Systems: Technology and Applications (IDAACS), IEEE, Cracow, Poland, 2021, pp. 284-287, doi: 10.1109/IDAACS53288.2021.9660913.
- [2] D. Zhuravchak, P. Hlushchenko, M. Opanovych, V. Dudykevych, A. Piskozub, Zero Trust Concept For Active Directory Protection To Detect Ransomware, *Cybersecurity: Education, Science, Technique*, vol.2, no.22, pp.179-190, 2023.
 - [3] R. S. Sandhu, Role-based Access Control, in: M. Zelkowitz (Ed.), *Advances in Computers*, volume 46, Elsevier, College Park, Maryland, 1998, pp. 237-286. doi:10.1016/S0065-2458(08)60206-5.
 - [4] A. Elliott, S. Knight, Role explosion: Acknowledging the problem, in: *Proceedings of Software Engineering research and practice, WORLDCOMP '10a*, Las Vegas, NV, 2010, pp. 349-355.
 - [5] A. S. M. Kayes, R. Kalaria, I. H. Sarker, M. S. Islam, P. A. Watters, A. Ng, I. Kumara, A survey of context-aware access control mechanisms for cloud and fog networks: Taxonomy and open research issues. *Sensors* 20 (2020). doi: 10.3390/s20092464
 - [6] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, J. Voas, Attribute-Based Access Control, *Computer* 48(2) (2015) 85-88. doi: 10.1109/MC.2015.33.
 - [7] Hubli, S. C., Potdar, G. P. Security evolution in object-oriented software systems: shaping the future landscape through innovative patterns, *IRJMETs* 6(1) (2024). doi: 0.56726/IRJMETs48793.
 - [8] G. Goyal, P. Liu, S. Sural, Securing Smart Home IoT Systems with Attribute-Based Access Control, in: *Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems, Sat-CPS '22*, Association for Computing Machinery, New York, NY, 2022. pp. 37–46. doi:10.1145/3510547.3517920
 - [9] S. Parkinson, S. Khan, A Survey on Empirical Security Analysis of Access-control Systems: A Real-world Perspective, *ACM Computing Surveys* 55 (2022) 1-28. doi:10.1145/3533703.
 - [10] J. A. Khan, Role-Based access Control (RBAC) and Attribute-Based Access Control (ABAC), in: P. Goel, H. Pandey, A. Singhal, & S. Agarwal (Eds.), *Improving Security, Privacy, and Trust in Cloud Computing*, 1st ed, IGI Global, Hershey, PA, pp. 113-126. doi:10.4018/979-8-3693-1431-9.ch005
 - [11] P.W.L. Fong, Relationship-based access control: protection model and policy language, in: *Proceedings of the first ACM conference on Data and application security and privacy, CODASPY '11*, Association for Computing Machinery, New York, NY, 191–202. doi:10.1145/1943513.1943539
 - [12] C. Arora, Higher-Order (Temporal) Relationship-Based Access Control, Master's thesis, University of Calgary, Calgary, Canada, 2022.
 - [13] L. Golightly, P. Modesti, R. Garcia, V. Chang, Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN, *Cyber Security and Applications* (2023). doi:10.1016/j.csa.2023.100015
 - [14] Daniel Bass, An intro to Open Policy Agent (OPA), 2022. URL: <https://www.permit.io/blog/introduction-to-opa>
 - [15] Daniel Bass, Real-time dynamic authorization - an introduction to OPAL, 2022. URL: <https://www.permit.io/blog/introduction-to-opal>
 - [16] J. Cutler, et al. Cedar: A New Language for Expressive, Fast, Safe, and Analyzable Authorization (Extended Version), arXiv preprint, 2024. doi:10.48550/arXiv.2403.04651.
 - [17] Mike Hicks, How we built Cedar with automated reasoning and differential testing, 2023. URL: <https://www.amazon.science/blog/how-we-built-cedar-with-automated-reasoning-and-differential-testing>
 - [18] R. Pang, R. Caceres, M. Burrows, Z. Chen, P. Dave, N. Germer, A. Golynski, K. Graney, N. Kang, L. Kissner, J. L. Korn, A. Parmar, C. D. Richards, M. Wang, Zanzibar: Google's Consistent, Global Authorization System, in: *Proceedings of the 2019 USENIX Annual Technical Conference, USENIX ATC 19*, USENIX Association, Renton, WA, 2019, pp. 33-46. isbn:978-1-939133-03-8
 - [19] Jake Moshenko, Google Zanzibar Open Source: The Architecture of SpiceDB, 2021. URL: <https://authzed.com/blog/spicedb-architecture>
 - [20] M. Penelova, Access Control Models, *Cybernetics and Information Technologies* 21(4) (2021) 77-104. doi:10.2478/cait-2021-0044
 - [21] J. Park, R. Sandhu, M. Gupta and S. Bhatt, Activity Control Design Principles: Next Generation Access Control for Smart and Collaborative Systems, *IEEE Access* 9 (2021), 151004-151022, doi: 10.1109/ACCESS.2021.3126201.

- [22] M. N. Nobi, R. Krishnan, Y. Huang, M. Shakarami, R. Sandhu, Toward Deep Learning Based Access Control, in: Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy, CODASPY '22, Association for Computing Machinery, New York, NY, 2022, pp. 143–154. doi:10.1145/3508398.3511497.

The implementation of Montgomery modular reduction to speed up of modular exponentiation

Ihor Prots'ko¹, Oleksandr Gryshchuk²

¹ Lviv Polytechnic National University, S.Bandery, 12, Lviv, 79013, Ukraine

² LtdC "SoftServe", Sadova, 2d, Lviv, 79021, Ukraine

Abstract

Modular exponentiation over large integers involves multiple modular multiplications, which is very computationally expensive. Many processing systems use the Montgomery modular multiplication method, which reduces the latency of software and hardware implementations. The main directions of software development and outlines of the parts of Montgomery modular multiplication for the implementation are presented. The class Montgomery Arithmetic over large integers is implemented using four methods for Montgomery modular multiplication. We present the computation of modular exponentiation using the right-to-left binary exponentiation method for a fixed basis with a developed pre-computation of a reduced set of remainders using modular Montgomery multiplication.

A comparison of the runtimes of three variants of functions for computing the modular exponentiation over large integers is performed. The algorithm with pre-computation of residues for fixed base provides a faster computation of modular exponentiation using Montgomery modular multiplication compared to the functions of modular exponentiation of the MPIR, OpenSSL libraries for large number more than 1K bits.

Keywords

Montgomery modular multiplication, modular exponentiation, multithreading, large numbers

1. Introduction

Modular reduction is the computation of $x \bmod m$. A basic operation in processing systems is computations in Z_m integers modulo m , where m is a large positive integer, which may or may not be a prime. Modular reductions are normally used to create finite groups, rings, or fields. The most common usage for performance-driven modular reductions is in modular exponentiation algorithms. An efficient implementation of the modular reduction $x \bmod m$ of large numbers is the key to high performance.

The classical algorithm of modular reduction has no restriction on the size of x , m and can easily be adapted to a division algorithm with quotient and remainder. The formalization consists of estimating the quotient digit as accurately as possible. This is justified by the fact that using multiplication and division are the most time-consuming operations in the inner loops of algorithms, especially when calculating Modular reduction over multi-bit numerical data.

Among the modular reduction algorithms: classical, Barrett, and Montgomery's, the Montgomery reduction is relatively simple and very efficient [1]. The baseline Montgomery reduction algorithm will produce the residue for any size input. Montgomery reduction is a common algorithm used for modulus reduction. The unique property of this algorithm is that it does not compute the modulus directly, but instead, the modulus multiplied by a constant.

The further development using Montgomery reduction for computing modular multiplication is much faster and does not require any division by m . This method is referred as Montgomery

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ ihor.o.protsko@lpnu.ua (I. Prots'ko); ocr@ukr.net (O. Gryshchuk)

🆔 0000-0002-3514-9265 (I. Prots'ko); 0000-0001-8744-4242 (O. Gryshchuk)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

modular multiplication and combines Montgomery reduction and multiple-precision multiplication.

The scientific problem of speeding up modular reduction for processing systems is relevant for the present stage of the development of information and computer technologies. The software implementations of modular multiplication over large integers on general-purpose processors are an important target for optimization. The further increase in the speed of the computational implementation of the modular reduction operation and then the full multiplication part can be achieved only by using the multithreading of multi-core processor architectures.

The paper is structured as follows: after the Introduction in Section 1 is described Montgomery Reduction as a common algorithm used for modulus reduction, and outlines the parts and basic stages of the Montgomery modular multiplication algorithm. Section 2 describes the developed software implementation of efficient Montgomery multiplication over large integers using the Multiple Precision Integers and Rationals library. For performance analysis in Section 3, the experiments and discussion of the software implementation of Montgomery modular multiplication for the computation of developed modular exponentiation are presented. As a result, the developed software implementation provides a faster computation of modular exponentiation using Montgomery modular multiplication compared to the general-purpose functions of modular exponentiation of the MPIR and OpenSSL libraries over large integers.

2. Literature review

There are different contemporary variations of Barrett and Montgomery algorithms, which have advantages and mines. Barrett reduction is a reduction algorithm proposed in 1986 by P.D. Barrett [2], designed to optimize the integers modulo m operation assuming m is constant and, divisions are replaced by multiplications. P. Barrett offered the idea of estimating the quotient $x \div m$ with operations that are less expensive in time than a classical multi-precision division by m . The only pre-computation $[2^{2n}/m]$ required for successful modular reduction use of Barrett's algorithm, where $2n$ is a number of bits. The computation of modular exponentiation based on Barrett's algorithm is better than the other known ones for small numerical values.

Montgomery reduction uses on the changing of the original reduction modulo by some other convenient modulo. By representing the residue classes modulo Montgomery's algorithm [3] replaces a division by m with a multiplication followed by a division by a power of radix r . In computer applications, b is usually defined as the power of 2, when $m = 2k$, k – the processor's word-size, this operation is very easy and inexpensive. The idea developed by P. Montgomery's method suggests that the operations of addition and subtraction are practically unchanged, but multiplication changes slightly in a simple procedure without using reductions modulo m . Montgomery's algorithm (only for modulo m for which $\gcd(m, r) = 1$) is faster than both the classical and Barrett's one and as fast as multiplication almost.

There are different implementation algorithms of Montgomery reduction, which are improving to simpler and higher regularity. The paper [4] proposes new residue number system Montgomery reduction algorithms, which achieve less number of unit multiplications. Traditional Montgomery approaches are combined with multiply-reduce methods at the bit-level in hardware implementations or based on the processor's word-size level for software implementations [5]. The parallel execution of modular operations "square and multiplications" based on Montgomery algorithm are described in the papers [6]. The implementation of the Montgomery algorithm has been improved over the years, both at the software and hardware levels [7].

3. Montgomery Reduction and Modular Multiplication

The Montgomery reduction of number T is defined as

$$TR^{-1} \bmod m, \quad (1)$$

where m is a positive integer, T and R are integers such that $R > m$, $\gcd(m, R) = 1$, and $0 \leq T < mR$.

The formula (1) is called a Montgomery reduction of number T modulo m with respect to R . Using Montgomery reduction easy to carry out modular reduction in the residue number system. The residue number system is a method for representing an integer as an n -tuple of its residues with respect to a given base. Montgomery Reduction $i R^{-1} \bmod m$ is a one-to-one mapping defined from Z/m_Z to Z/m_Z , for $0 \leq i < m$.

To compute the Montgomery reduction, it is necessary to determine the value of R^{-1} that meets the condition $R \cdot R^{-1} \bmod m = 1$.

To find the inverse modulo, you can use the extended Euclidean algorithm. Indeed, if $\gcd(m, R) = 1$, then the following integers will be found u and v , that

$$Rv + mv = 1. \quad (2)$$

If we pass to congruence modulo m in the last equality, then we obtain $Ru \equiv 1 \pmod{m}$, which gives $(R^{-1}) \equiv u \pmod{m}$.

For working with large numbers, where Montgomery multiplication is implemented, is common to write the Montgomery radix R as

$$R = r^k = 2^k, \quad (3)$$

where k is the word-size of the computer architecture. Higher radices may be used but the radix-2 provides a simple algorithmic and hardware implementation.

The algorithm to compute Montgomery constant $\mu = -m^{-1} \bmod R$ for odd values m and $R = 2^k$ is presented in the Fig. 1.

Algorithm. Compute Montgomery constant $\mu = -m^{-1} \bmod R$
 Input : Odd integer m and $R = 2^k$
 Output : $\mu = -m^{-1} \bmod R$

```

 $y \leftarrow 1$ 
for  $i = 2$  to  $k$  do
  if  $(m y \bmod 2^i) \neq 1$  then
     $y \leftarrow y + 2^{i-1}$ 
  end if
end for
return  $\mu \leftarrow R - y$ ;
```

Figure 1: Algorithm of Computation of the Montgomery constant $-m^{-1} \bmod R$

There are different fast Modular Reduction Methods to implementing Montgomery modular reduction. The algorithm Montgomery Reduction for radix 2, which does not require some pre-computation is presented in Fig. 2.

Algorithm Montgomery Reduction $X R^{-1} \bmod m$
 Input : X, m and $R = 2^k$
 Output : $X 2^{-k} \bmod m$

```

 $x = X$ 
for  $i = 1$  to  $k$  do
  if  $x$  is odd then
     $x = x + m$ ;
     $x = x/2$ ;
  end if
end for
return  $x$ ;
```

Figure 2: Algorithm of Computation of the Montgomery Reduction for radix 2

This algorithm is based on scanning the bit of a large number X from the right (the least significant bit) to the left (the most significant bit).

In the paper [8] is described the efficiently computes Montgomery reduction. Let $m' = -m^{-1} \bmod R$, if $U = Tm' \bmod R$, $m * m^{-1} \bmod R = 1$, then

$$T R^{-1} \bmod m \equiv (T + U m) / R. \quad (4)$$

Taking the remainder modulo m was replaced by division by R , and also taking the remainder modulo R in the numerator of the formula (4). As a result, we can choose such R that truncation can be used instead of division. If we have long arithmetic with some radix r , then the degree of this radix r_i . That is, modulo residues and divisions will turn into shifts and throw out extra numbers. In the chapter 14.3.2 Montgomery reduction [8] are presented the algorithms and examples of Montgomery reduction based on formula (4). The algorithm does not require $m' = -m^{-1} \bmod R$, but rather $m' = -m^{-1} \bmod r$.

Most processing systems are implemented by repetition of a modular multiplication with a large modulus m , that is,

$$z = x \cdot y \bmod m. \quad (5)$$

where m is usually a large prime or a product of two large primes $x = (x_{n-1} \dots x_1 x_0)_r$ and $y = (y_{n-1} \dots y_1 y_0)_r$, which are non-negative integers in a radix r representation such that $x < m$ and $y < m$.

Let us represent x' and y' of a number x and y in the Montgomery space as follows

$$x' = x R \bmod m \text{ and } y' = y R \bmod m.$$

The Montgomery reduction of multiplication $x'y'$ is:

$$x' \cdot y' R^{-1} \bmod m \equiv (x R \bmod m * y R \bmod m) / R^{-1} \bmod m = x \cdot y R \bmod m. \quad (6)$$

This means that, after doing the multiplication of two numbers in the Montgomery space, we need to reduce the result by multiplying it by R^{-1} and taking of modulo m . There is an efficient way to use Montgomery reduction. This operation called the Montgomery modular multiplication. Montgomery modular multiplication itself is fast, but it requires some pre-computation. Montgomery multiplication algorithm involves three basic stages:

1. The conversion of operands from integer domain to Montgomery space;
2. The multiplication of operands in the Montgomery space;
3. The conversion of operands back from Montgomery space to integer domain.

The Montgomery multiplication needs to convert x and y into Montgomery space and their product out of Montgomery space (Fig. 3). In this method the costly division operation usually needed to perform modular reduction is replaced by simple shift operations by conversing the operands into the reduced number system domain before the operation and re-conversing the result after the operation. Montgomery modular multiplication involves: first conversion of operands into the Montgomery space, multiplication and then after the result is re-converted into the Montgomery space.

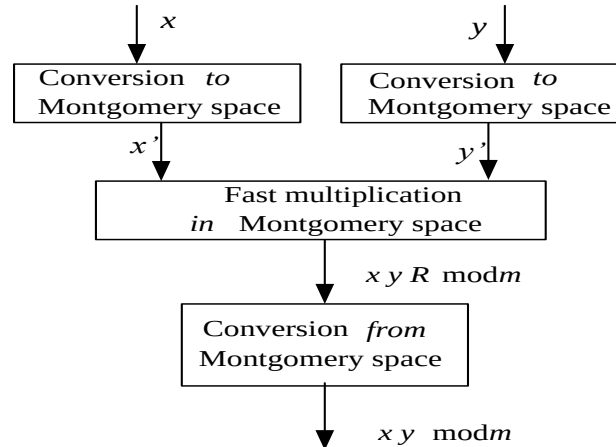


Figure 3: Computation of modular reduction using Montgomery modular multiplication

For practical (Fig. 4) interest the $R=r^n$ will suffice when there can be a power of 2 and $R=2^n$ [9]. The condition $R > m$ is clearly satisfied, but $\gcd(m, R) = 1$ needs to be relatively prime i.e. must not have any common non-trivial divisors which will hold only if $\gcd(m, r) = 1$.

Montgomery modular multiplication algorithm $XY(R^{-1}) \bmod m$
 Input : X, Y, m and $R=2^k$,
 Output : $XY 2^{-k} \bmod m$
 S_0 : LSB of $S, x_i \in (x_{n-1} \dots x_1 x_0)_2$

```

S=0
for i= 0 to n do
    S = S + xi Y;
    S = S + S0 m;
    S = S/2;
if S ≥ m then S = S-m;
return S;

```

Figure 4: Algorithm of computation of the Montgomery modular multiplication

There are different implementations of Montgomery modular multiplication: the digit-serial architectures [10], special purpose circuits, what perform multiplication and reduction simultaneously [11], and parallel execution of modular multiplication [12]. In practice at the software and hardware levels, Montgomery multiplication is the most efficient method when is used a very regular structure, which speeds up the implementation [13, 14].

The software implementations of modular multiplication over large integers on general-purpose processors are an important target and has been improved over the years. In the next Section, we describe the software implementation of efficient Montgomery multiplication over large integers using the Multiple Precision Integers and Rationals library.

4. The software implementation of Montgomery reduction to modular multiplication

The software implementations of Montgomery modular multiplication on the general purpose processors are an important target for optimization. Important focus is on the software implementation of the full multiplication parts including the efficient reduction. Many works improve the performance of a Montgomery Multiplication [15, 16]. Almost all the implementations of modular multiplication in many processing systems are performed in assembly languages to take advantage of the specific architectural properties of the processor [17].

In this section, we describe software implementations of modular multiplication on the basis of the realization of Montgomery modular multiplication, which includes the efficient modular reduction and multiplication parts.

The modular multiplication is implemented in C++ language. The developed *class MontgomeryArithmetic* (Fig. 5) implements the Montgomery modular multiplication and reduction using the Multiple Precision Integers and Rationals library (MPIR) [18].

```

Class MontgomeryArithmetic
private:
    const mpz_class mod_;
    size_t mod_size_;
    mpz_class inv_;
    const size_t limbs_;
    const size_t bits_;
public:
    explicit MontgomeryArithmetic(const mpz_class& mod);
    mpz_class init(const mpz_class& x) const;
    void multiply(mpz_class& a, const mpz_class& b) const;
    void reduce(mpz_class& x) const;

```

Figure 5: The *MontgomeryArithmetic* class

According to the markings in Fig. 5, the member variables of the *Class MontgomeryArithmetic* are: *size_t mod_size_* is a divisor size in MPIR limbs (64-bit integers); *mpz_class inv_* is a pre-computed inverse factor for the Montgomery reduction; *const size_t limbs_* is the same as *size_*, but a more convenient name;

*const size_t bits_* is a bit count for the modular arithmetic.

The parameters of the methods are:

mod is a divisor for modular arithmetic;

x is a number for the conversion;

a and *b* are the first and second numbers converted to the Montgomery space.

The constructor *MontgomeryArithmetic(const mpz_class& mod)* computes a modular inverse factor for the Montgomery reduction and initializes other member variables, where the argument *mod* is a divisor for modular arithmetic.

For computing the inverse factor $m' = m^{-1} \bmod R$ efficiently, we can use the mathematical dependence, which is inspired by Newton's method. The algorithm for calculating the inverse factor is described and proved in [19] :

$$m \cdot x \equiv 1 \bmod 2^k \rightarrow m \cdot x \cdot (2 - m \cdot x) \equiv 1 \bmod 2^{2k}. \quad (7)$$

This means we can start with $x = 1$, as the inverse of m modulo 2^1 , apply the trick of power times and in each iteration we double the number. This algorithm uses only shifts, subtractions and multiplication of large numbers in each iteration and has the same computational complexity as the algorithm, which is shown in Fig. 1.

The method *init mpz_class init(const mpz_class& x) const* converts a number to the Montgomery space. It is required to convert all numbers before applying the Montgomery multiplication. The algorithm for the conversion is described in [19], where the relation is used

$$x \cdot R \bmod m = x R^2 / R = x \cdot R^2, \quad (8)$$

where x is a number for the conversion. Converting the number into the space is just a multiplication inside the space of the number with R^2 . Therefore, we can pre-compute $R^2 \bmod m$ and just perform a multiplication instead of shifting the number. This algorithm uses the shifts and the subtractions and multiplications of large numbers in each *bits_* iteration.

The method returns the converted value, which can be used for the Montgomery multiplication. The method *void multiply(mpz_class& a, const mpz_class& b) const* multiplies two numbers, where a, b are the numbers converted to the Montgomery space. The method returns the result via first argument in place and then performs the Montgomery reduction. It modifies the first argument in place to improve efficiency and avoid copying. For multiplication, it uses regular multiplication provided by the MPIR library, which is optimized using AVX2 SIMD instructions.

The method *void reduce(mpz_class& x) const*, where argument x is a number for the reduction in place, computes the Montgomery reduction in place. Any number from the Montgomery space can be converted back using this method. This is one of the most performance-critical methods. The MPIR library [18] offers a few low-level implementations, which can be further optimized for specific use cases. This method calls the *mpn_redc1()* function provided by MPIR to compute the Montgomery reduction in place.

The methods and initialized member variables in the developed *class MontgomeryArithmetic* provide an implementation of Montgomery modular multiplication corresponding to Fig. 3. The operations of multiplication and division by $R = 2^k$ are very fast in the methods of class, as they are just bit shifts. Thus, Montgomery's algorithm is faster than the usual $(a \cdot b) \bmod m$, which contains division by m . However, the computation R^{-1} , m^{-1} and conversion of numbers to the remains and vice versa are time-intensive operations, as a result, of which it is inefficient to use the product for a single computation. Montgomery reduction is the fastest in computing a reasonably long series of modular reductions, for instance in computing exponential function. This algorithm is a time critical step in the computation of the modular exponentiation operation.

5. Experiments and discussions of the software implementation of Montgomery modular multiplication for the computation of modular exponentiation

Modular exponentiation over large integers involves multiple modular multiplications, which is very computationally expensive. Modular exponentiation of large numbers is extremely

necessary for providing high crypto capability of information data, for finding the discrete logarithm, in number-theoretic transforms and many other applications.

Considerable attention is paid to the development of effective methods of modular exponentiation aimed at effective computation and reduction of the execution time of the modular exponentiation operations [20, 21]. One of the ways to speed up computations of modular exponentiation is parallelization of computations using modern software technologies for universal computer systems or creation of specialized computing tools. The software implementation of the Montgomery multiplication and modular exponentiation computation is included in the software libraries Crypto++, OpenSSL, PARI/GP, MPIR designed for working with large numbers.

The production-grade software library and full-featured toolkit popular on Linux and other systems is OpenSSL library. OpenSSL library contains a set of tools that implements the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) [22]. The functions *BN_mod_mul_montgomery*, *BN_MONT_CTX_new* of OpenSSL library implement Montgomery multiplication. The library includes three functions to calculate the modular exponentiation using Montgomery multiplication: *BN_mod_exp_mont()*, which calculates A to the power of x modulo m , and *BN_mod_exp_mont_consttime()*, *BN_mod_exp_mont_consttime_x2()*.

Let's compare the use of Montgomery modular multiplication with the usual modular multiplication operation on the example of an efficient computation of modular exponentiation of large numbers. Consider the basic iterative algorithm using pre-computation to form a shortened sequence of residues of the fixed base A for computing the modular exponentiation

$$y = A^x \bmod m. \quad (9)$$

The central idea to calculate $A^x \bmod m$ is to use the binary representation of the exponent x . For a fixed-base A of the modular exponentiation (9), which is equal to the product of the residuals $r_0, r_1, \dots, r_{k-1}$ of the exponent $(A^{2^i}) \bmod m$, ($i = 0, 1, 2, \dots, k-1$). Modular exponentiation is implemented using the development of the right-to-left binary exponentiation method for a fixed base with pre-computation of a reduced set of residuals. That can speed up the process of computing the modular exponentiation by pre-computing (Fig. 6) the sequence of residuals, and repetitions with the period T' after the offset u in the unit Precomputation u, T' [23].

The scheme (Fig. 6) for computing the modular exponentiation consists of the denotations:

- A is the input of the base number; m is the input of the module;
- x is the input of an exponent with binary digits $x_{(k-1)}, x_{(k-2)}, \dots, x_2, x_1, x_0$;
- $(A^{2^i})_m$ are blocks of computation of the integer exponent of exponent 2^i of the number A by the module m , $i = 0, 1, 2, \dots, (k-1)$;
- $r_0, r_1, \dots, r_{k-1}$ are residuals $A^{2^i} \bmod m$, ($i = 0, 1, 2, \dots, k-1$),
- $(X) \bmod m$ is the block of modular multiplication;
- y is the output of the modular exponentiation.

Thus, applying the parallel execution of the computation of modular exponentiation with the pre-computation, threads are created during the software execution of the modular multiplication of residual values r_i , where $i \leq T'$, in the block of modular multiplication. These residual values r_i are determined in the process of computing of residual exponents $(A^{2^i}) \bmod m$, ($i = 0, 1, 2, \dots, k-1$). The only difficulty in organizing computations with such threads is the need to synchronize the streams and the unit of Precomputation u, T' to ensure the correct computation of the final value y of modular exponentiation.

To implement the algorithm for computing the integer power of a number A^x by modulus m , the MPIR library is used, which is written in C and assembler and provides the ability to compile its functions in Visual Studio C++. Accordingly, in the MPIR library, the data type *mpz_t* represents large numbers of arbitrary length, which are chosen for the power of the number base and mod with the number of bits from 256 to 2048 bits for testing. However, using the function *mpn_redc1()* implement Montgomery multiplication is not efficient enough in the process of modular exponentiation.

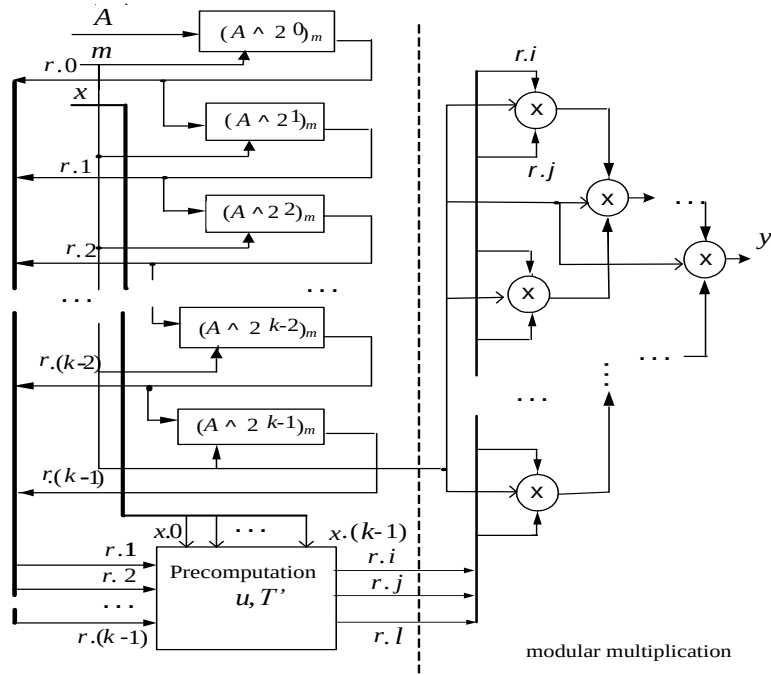


Figure 6: The scheme for computation of modular exponentiation $y = A^x \bmod m$ with pre-computation

The algorithm consists of *precompute()* and *precompute_parallel()* functions. The *precompute()* function determines the sequence of a reduced set of residues. The *precompute()* function calculates the sequence of remainders for fixed numbers *base* and *mod* for $exp = 2^i$ ($i = 0, 1, 2, \dots$) and analyzes the periodicity with the appearance of each defined remainder $r.i$, which are calculated by the *find_remainders()* function. The pre-computation has been made in a separate *find_remainders()* function to optimize multiple remainder searches $(A^{2^i}) \bmod m$. The function *update_remainders()* reduces the length of the sequence of remainders as a result of fixing the periodicity T' , taking into account the offset u .

The *precompute_parallel()* function aim to compare the performance execution with the use of Montgomery modular multiplication and usual modular multiplication operation. To implement the algorithm, the *mpz_init_set (mul, base)*, *mpz_sizeinbase (exp, 2)*, *mpz_tstbit (exp, i)*, *mpz_mul (r, r, mul)* functions from the MPIR library are used, the parameters of which can be multi-bit data limited to bit size 2048 bits. To organize efficient multithreading computation of modular exponentiation according to the *precompute_parallel()* function, the *thread_function()* and *parallel()* are implemented. The developed *precompute_parallel()* function uses multiple threads for the computation of the modular exponentiation. The method *run()* runs parallel exponentiation using multiple threads. It has the following steps:

- 1) creates a collection of the active exponent bits;
- 2) splits the exponent bits among the defined number of threads;
- 3) waits for every thread execution.
- 4) calculates the final result by multiplying partial results calculated by the threads.

The final result of the function is written to the variable *s_thread_result*, and the computation time is fixed and averaged to output.

We compare the time of calculating the modular exponent using the usual modular multiplication with the Montgomery modular multiplication based on the developed functions *precompute_modulo()*, *precompute_parallel_modulo()* and *precompute_montgomery()*, *precompute_parallel_montgomery()*, respectively.

Testing of the calculation of modular exponentiation were carried out on a computer system with a multi-core microprocessor Intel Core i9-10980XE (18 cores, 36 threads, 3.0GHz) with shared memory in a 64-bit Windows. According to hyper-threading technology, each physical core of 18 consists of two virtual 36 ones. The numerical results are presented in Figure 7, which contains the values of average execution time (μs microseconds) for 500 and 250 trials of

computing the modular exponentiation for pseudo-random data *base*, *exp*, *mod* for 1024 bits and 2048 bits.

```

Командный рядок
Enter number of threads (2..64) > 12
===== bits=1024 trials=500 =====
base = 12721060630754409625292260503576596349857635561196288808652696561975202158929122632961118436394053817448876189632
07920057135985340347211711431162815981800688806439320000660228338078534392042255435215892772898818014960680315173136660
33797426823103562740550577334927448756210422810699954369405284794763585910
exp = 142835209786489997170873255507946573556448214084628524605421188224099687322984673543614176852071053547415698252622
573845683075452982474922517841367278609089136988583447756479483918441795733215771335093892746851604252279730368411597397
577626119447392355080344631875081748708394736819710836176156337925349995164
mod = 368223265393616765838289047484087924724083351214856164756403136919936573865892058431345272076782139089436981490798
503001806033596893273007523252013138190688398066467075600577730915050017234560082947101924548982647158267357750118464079
6332529642734561465318932673353360118318330216661780501404921220381528643
mpz_powm average time = 301 microseconds.
precompute_modulo average time = 302 microseconds.
precompute_montgomery average time = 180 microseconds.
precompute_parallel_modulo average time = 153 microseconds.
precompute_parallel_montgomery average time = 71 microseconds.
===== bits=2048 trials=250 =====
base = 20498029031211213516565822840229761961320835651346614663317794240083339030983123010071164303284509646555769270863
912393542850240253530970605341865669423311669663064208489768715890569302112265638341318004988435595314134625325628311364
830215315733074763251664649909209703082695735960838862474347161067863717087046061641109513208507216683290127954925328495
260842559986644911036492884727353319775222498757195420699142721986578630749268470630599143554615293794640765714058225304
80569236918545081943298334955457185046022092761597333468733853070540541037500298179827364715521180234116278941177627452
28342173791692628840738
exp = 206971186674602942893291319268428623712608587189616225423903941285779658834620654647264762656215005844221010903248
805904788942050645268534371871257651724912857624853913319544665818453970774205805936276513235167804757330671171360229336
910074202766840819523964084411554460264006668359867024199348668244418903647742281408991589590100597574944920059811530558
721874424954824117451346461205848045559295541835156979224291886237220605698948712689724650080897444104535423282766208959
38777704297176988032776203315585798048230323891888933926985203629914823135222855353547016859173882001780159272297308256
1458560545884722373680
mod = 39379307836430888989982185920314902063610143414215425345540627854542150881576612854892278971719513827185241896934
84043298053212367558745592744633067379665068316786711862281193766926085121967533895669525512487774111648354763670474879
583602698230326785874988566339640136473488173875558706171182342716634896316172735583709359139773798460819278277075831296
56866254202071251226329658682484634354372671824932447221319815753300084769255407490226142154721378943229480820161164938
614827865775795822192901667030446623011805703635693587741515918942940244348835864709049275298972146812975233297111844366
6005795214405971895267
mpz_powm average time = 2088 microseconds.
precompute_modulo average time = 1290 microseconds.
precompute_montgomery average time = 1050 microseconds.
precompute_parallel_modulo average time = 411 microseconds.
precompute_parallel_montgomery average time = 173 microseconds.

```

Figure 7: The results of testing the functions of computing the modular exponentiation on a computer system with an Intel Core i9-10980XE processor with a chosen number of threads of 12

The pre-computation time to determine the sequence of a reduced set of residues is taken into account, therefore the total average time for computing the modular exponentiation *modexp()* is equal to:

1) for the usual modular multiplication operation:

$$\text{modexp}() = \text{precompute\_modulo}() \text{ time} + \text{precompute\_parallel\_modulo}() \text{ time.}$$

In accordance with the result of testing (Fig. 7) average time are equal to:

$$\text{modexp}() = (301 + 153) \mu\text{s} = 454 \mu\text{s};$$

$$\text{modexp}() = (1290 + 411) \mu\text{s} = 1701 \mu\text{s};$$

for pseudo-random data the base, exp, mod of 1024 bits and 2048 bits respectively.

2) for the Montgomery modular multiplication:

$$\text{modexp}() = \text{precompute\_montgomery}() \text{ time} + \text{precompute\_parallel\_montgomery}() \text{ time.}$$

In accordance with the result of testing average time (Fig. 7) are equal to:

$$\text{modexp}() = (1290 + 71) \mu\text{s} = 251 \mu\text{s};$$

$$\text{modexp}() = (1050 + 173) \mu\text{s} = 1223 \mu\text{s};$$

for pseudo-random data the base, exp, mod of 1024 bits and 2048 bits respectively.

Therefore, the implementation of the Montgomery modular multiplication is based on the developed *class MontgomeryArithmetic* for computing the modular exponentiation speed up $454 \mu\text{s} / 251 \mu\text{s} = 1,8$ and $1701 \mu\text{s} / 1223 \mu\text{s} = 1,4$ times for pseudo-random data the base, exp, mod of 1024 bits and 2048 bits.

A highly optimized modification of the well-known GMP or GNU Multiple Precision Arithmetic Library the MPIR library [18] contains the function *mpz_powm()* to realize the computation of modular exponentiation. The MPIR library uses an optimized version a floating-window algorithm of the modular exponentiation with Montgomery multiplication/reduction, which reduces the average number of multiplication operations. The function of the MPIR library

mpz_powm(expected_result, base, exp, mod) better performs modular exponentiation than function *BN_mod_exp_mont()* of the OpenSSL and Crypto++ libraries in accordance with the results received in [24], therefore we chose the function *mpz_powm()* for comparison (Fig. 5). At testing results, the total average time of *mpz_powm()* function for computing the modular exponentiation *modexp()* is 301 μ s and 2088 μ s and is greater than the average time for computing the modular exponentiation the Montgomery modular multiplication 251 μ s and 1223 μ s for pseudo-random data the base, exp, mod of 1024 bits and 2048 bits respectively.

The closest scientific work for comparing research results is work [25], where an approach that uses vector SIMD instructions for parallel computation of multiple Montgomery multiplications is applied. This work [25] describes the fact of the comparison of a parallel version of Montgomery multiplication using vector SIMD instructions to the implementation of the function of modular exponentiation in the OpenSSL library. The parallel version of Montgomery multiplication using vector SIMD instructions performance increases by more than a factor of 1.5 compared to the implementation in the OpenSSL library in the classical arithmetic logic unit on the Atom platform for 2048-bit moduli. Our implementation of the modular Montgomery multiplication to compute the modular exponentiation has factors 1.8 and 1.4 for the pseudorandom data the base, exp, mod of 1024 bits and 2048 bits compared to the sequential implementation in MPIR library. According to the obtained results of modular exponentiation [24], the MPIR library is faster for large numbers than OpenSSL.

The values of an average execution time of modular exponentiation depend on the computing capabilities in universal computer systems. Testing results was received on two computer systems with different computing capabilities with processors an Intel Core i9-10980XE (18 cores, 36 threads, 3.0GHz) and Intel Core i9-13900K (24 cores, 32 threads, 3.0GHz). The results are presented in Table 1, which contains the values of average execution time (μ s microseconds) for 500 trials of the functions *modexp()* and *montgomery_modexp()* using developed Montgomery modular multiplication for computing the modular exponentiation with pseudo-random data of 1024 bits.

Table 1
The average execution time (μ s) of the functions of computing the modular exponentiation

Release/x86	Intel Core i9-10980XE	Intel Core i9-13900K
Data bits / trials	1024 / 500	1024 / 500
precompute_parallel modulo <i>modexp()</i>	554	225
precompute_parallel <i>montgomery_modexp()</i>	255	153

The optimal number of threads is 12...16 for fast computation of modular exponentiation for universal computer systems [24].

Therefore, based on the developed Montgomery modular multiplication software the further implementation of the computation of modular exponentiation using multithreaded technologies will provide an opportunity for the efficient computation of modular exponentiation with a fixed base.

Conclusions

In the work is compared and analysed the developed software implementation of the class *MontgomeryArithmetic* in modular exponentiation function. The main directions of software development and outline of the parts of Montgomery modular multiplication for the implementation are presented. Modular exponentiation with a fixed base is implemented using the development of the right-to-left binary exponentiation method with pre-computation of a reduced set of residuals with the use of Montgomery modular multiplication or the usual modular multiplication. The average run time of the computation on multi-core microprocessors of

universal computer systems have been defined. As a result, an algorithm with pre-computation of residues for fixed base provides faster computation in average 1,5 times of modular exponentiation using Montgomery modular multiplication compared to the functions of modular exponentiation using the usual modular multiplication.

The scientific novelty of obtained results lies in the implementation of parallelism using multithreading in the function of computing the modular exponentiation based on Montgomery modular multiplication, which is the best among the known modular exponentiation functions of Crypto++, OpenSSL and MPIR libraries for large numbers more than 1K bits.

The practical significance of the work lies in the fact that the obtained results can be successfully applied in modern asymmetric cryptography, for efficient computation of number-theoretic transforms and other computational problems.

Prospects for further research are the parallel implementation of Montgomery Modular Multipliers in the developed function of the modular exponentiation for large numbers using the computation on the video cards.

Acknowledgements

The authors are grateful to Roman Rykmas of the team leader of Uniservice LtdC for participation in testing and discussing the results obtained by the functions of computing the Montgomery modular multiplication.

The work was carried out within the framework of the state-budget scientific-research work of DB "Neuroruh" of the Lviv Polytechnic National University.

References

- [1] M. J. Ferrao, K. Kiran V. G, N., M. Megha, Implementation of Modular Reduction and Modular Multiplication Algorithms. IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), 8(6), Ver. I (2018). doi: 10.9790/4200-0806013438.
- [2] Barrett reduction, 2023. URL: https://handwiki.org/wiki/Barrett_reduction.
- [3] Montgomery Reduction Scheme Functions, 2022. URL: <https://www.intel.com/content/www/us/en/docs/ipp-crypto/developer-reference/2022-2/montgomery-reduction-scheme-functions.html>.
- [4] S. Kawamura, Y. Komano, H. Shimizu, T. Yonemura, RNS Montgomery reduction algorithms using quadratic residuosity. Journal of Cryptographic Engineering, 9 (2019). doi: 10.1007/s13389-018-0195-8.
- [5] Bing Li, Jinley Wang, Guocheng Ding, Haisheng Fu, Bingjie Lei, Haitao Yang, Jiangang Bi, Shaochong Lei. "A high-performance and low-cost Montgomery modular multiplication based on redundant binary representation." IEEE Trans. Circuits Syst. II Express Briefs 68.7(2021): 2660-2664. doi:10.1109/tcsii.2021.3053630.
- [6] I. Boiarshyn, O. Markovskiy, B. Ostrovska, Organization of parallel execution of modular multiplication to speed up the computational implementation of public-key cryptography, Information, Computing and Intelligent systems 3 (2022). doi:10.20535/2708-4930.3.2022.265418.
- [7] J. Bos, S. Friedberger, Faster modular arithmetic for isogeny-based crypto on embedded devices, Journal of Cryptographic Engineering, 10.2 (2020). doi: 10.1007/s13389-019-00214-6.
- [8] A. Menezes, J. Oorschot, A. Vanstone, Handbook of Applied Cryptography, Taylor & Francis excl. spl reprint, India, 2018.
- [9] R. K. Venkata, S. C. Simranjeet, V. Desalphine, D. Selvakumar, A Low Latency Montgomery Modular Exponentiation, Procedia Computer Science, 171 (2020). doi:10.1016/j.procs.2020.04.087.
- [10] S. Fatemi, M. Zare, A. Khavari, M. Maymandi-Nejad, Efficient implementation of digit-serial Montgomery modular multiplier architecture, IET Circuits Devices Syst., 13.7 (2019). doi:10.1049/iet-cds.2018.5182.
- [11] S. Srinitha, S. Niveda, S. Rangeetha, V. Kiruthika, A High Speed Montgomery Multiplier used in Security Applications. In: Proceedings of the 3rd International Conference on Signal

Processing and Communication (ICPSC), Coimbatore, India, 2021, pp. 299–303, URL: https://www.proceedings.com/content/059/059276_webtoc.pdf.

[12] B. Buhrow, B. Gilbert, C. Haider, Parallel modular multiplication using 512-bit advanced vector instructions: RSA fault-injection countermeasure via interleaved parallel multiplication, *Journal of Cryptographic Engineering* 2 (2021). doi: 10.1007/s13389-021-00256-10.

[13] Jinan Ding, Shuguo Li. “A low-latency and low-cost Montgomery modular multiplier based on NLP multiplication.” *IEEE Trans. Circuits Syst. II Exp. Briefs*, 67.7 (2020): 1319-1323. doi:10.1109/TCSII.2019.2932328.

[14] Z. Zhang, P. Zhang, A Scalable Montgomery Modular Multiplication Architecture with Low Area-Time Product Based on Redundant Binary Representation, *Electronics* 11.3712 (2022). doi: 10.3390/electronics11223712.

[15] M. Issad, B. Boudraa, M. Anane, A. M. Bellemou, Efficient PSoC Implementation of Modular Multiplication and Exponentiation Based on Serial-Parallel Combination, *Journal of Circuits, Systems and Computers* 28.13 (2019). doi: 10.1142/s0218126619502293.

[16] D. Mukhopadhyay, Improvement over Montgomery Modular Multiplication. *Information Systems Security*. In: *Proceedings of 17th International Conference (ICISS 2021)*, Patna, India, 2021, pp. 212–217. doi: 10.1007/978-3-030-92571-0_14N.

[17] Drucker, S. Gueron, Fast modular squaring with AVX512IFMA, Report 2018/335, *Cryptology ePrint Archive*, Preprint. MINOR revision (2018). URL: <http://eprint.iacr.org/2018/335>.

[18] MPPIR: Multiple Precision Integers and Rationals. 2021. URL: <http://mpir.org/>.

[19] Montgomery Multiplication, 2022. URL: https://cp-algorithms.com/algebra/montgomery_multiplication.html#implementation.

[20] J. Robert, C. Negre, T. Plantard, Efficient Fixed Base Exponentiation and Scalar Multiplication based on a Multiplicative Splitting Exponent Recoding, *Journal of Cryptographic Engineering*, 9.2 (2019). doi: 10.1007/s13389-018-0196-7.

[21] N. Emmart, F. Zhenget, C. Weems, Faster modular exponentiation using double precision floating point arithmetic on the GPU. In: *Proceedings IEEE 25th Symposium on Computer Arithmetic (ARITH)*, Amherst, MA, USA, 2018, pp. 130–137. doi: 10.1109/ARITH.2018.8464792.

[22] OpenSSL . Cryptography and SSL/TLS Toolkit, 2024. URL: <http://www.openssl.org/>.

[23] I. Prots'ko O. Gryshchuk, The Modular Exponentiation with precomputation of reduced set of residues for fixed-base, *Radio Electronics, Computer Science, Control* 1 (2022). doi: 10.15588/1607-3274-2022-1-7.

[24] I. Prots'ko, O. Gryshchuk, V. Riznyk, Efficient Multithreading Computation of Modular Exponentiation with Pre-computation of Residues for Fixed-base. In: *Proceedings of Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS 2023)*, Zaporizhzhia, Ukraine, 2023, pp.224-234. doi:10.32782/cmisis/3392-19.

[25] J. Bos, P. Montgomery, Montgomery arithmetic from a software perspective, in: J. W. Bos, A. K. Lenstra, (Ed.), *Topics in Computational Number Theory Inspired by Peter L. Montgomery*, 1st. ed., Cambridge University Press, 2017, pp.10-39. doi: 10.1017/9781316271575.

Enhancement of Land Cover Classification by Geospatial Data Cube Optimization

Artem Andreiev¹, Anna Kozlova¹, Leonid Artiushyn² and Peter Sedlacek³

¹ Scientific Centre for Aerospace Research of the Earth, Institute of Geological Sciences, National Academy of Sciences of Ukraine, Olesia Honchara str., 55-b, Kyiv, 01054, Ukraine

² State Research Institute of Aviation, Hryhoriia Andriuschenka Str., 6-V, Kyiv, 01135, Ukraine

³ University of Zilina, Univerzitna, 8215/1, Zilina, 01026, Slovakia

Abstract

This paper presents the optimization technique to reduce the geospatial data cube size and enhance the land cover classification. The technique is based on training sample separability. Accordingly, the Separability Index of the Training Sample (SITS) was developed and used as the object function for the optimization. In order to test the effectiveness of the optimization technique, the experiment was conducted. It implied the land cover classification of the highly heterogeneous natural landscapes in the case of the Shatsky National Natural Park, where the prevailing landscape is wetlands. After the optimization of the input geospatial data cube, classification enhancement was evidenced by increasing indicators such as overall accuracy by 0.04 from 0.9 to 0.94 and the kappa coefficient by 0.06 from 0.86 to 0.92. In addition, the data cube size was reduced by 5.55 times from 222 to 40 layers

Keywords

Remote sensing, land cover classification, supervised classification, training sample separability, geospatial data cube, data optimization

1. Introduction

Land cover classification is a critical process in remote sensing, providing spatially explicit information at different scales for numerous environmental applications [1]. Such information is widely applied to issues that require practical geospatial solutions like land cover change detection [2], environmental monitoring [3,4], fossil fuel exploration [5], and landmine detection [6]. Land classification techniques, likewise, play a crucial role in the integration of Earth observation data into comprehensive interdisciplinary issues on sustainable development goals achieving [7, 8], in particular, combat climate change and its impacts [9], reverse land degradation [10] and halt biodiversity loss [11], protect water-related ecosystem for safety water supply [12], and provide support for food security and sustainable agriculture [13, 14].

Today, most classification methods are mainly divided into supervised and unsupervised [15]. However, in remote sensing, the supervised classification methods are the most appropriate for the majority of the thematic tasks because applying these methods can establish the characteristics of the output classes, unlike unsupervised ones. A training sample set is used to set the characteristics of classes in supervised classification methods. Such a set contains the signatures of features of each class.

The input data for classification is heterogeneous geospatial data, which can be represented in the form of raster layers. To combine such layers into a single array, it is customary to form a geospatial data cube [16]. From ordinary datasets, data cubes differ by integrating different data types into a coherent and interoperable structure [17, 18]. After the cube's formation, the training sample's signatures must be determined in each layer. Hence, each layer is a feature of the training sample.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ artem.a.andreev@gmail.com (A. Andreiev); ak.koann@gmail.com (A. Kozlova); artleonid2017@gmail.com (L. Artiushyn); peter.sedlacek@fri.uniza.sk (P.Sedlacek)

🆔 0000-0002-6485-449X (A. Andreiev); 0000-0001-5336-237X (A. Kozlova); 0000-0002-7488-7244 (L. Artiushyn); 0000-0002-7481-6905 (P.Sedlacek)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

A geospatial data cube for specific classification tasks can comprise numerous layers. Including multitemporal data, a data cube aims to distinguish dynamic objects that change significantly during a specific period or vary much from each other at different stages of their development, e.g. vegetation cover, wetlands in particular [12]. Different physical aspects of multisource data, e.g. optical and radar, highlight diverse object traits and variations in land cover types [19]. Multiple ancillary data, e.g. data on geomorphology, hydrology, or phenology, help to differentiate land cover types due to their context [20].

However, the redundancy of the geospatial data cube causes two significant problems [21]. Firstly, the processing of such a data cube has high computational complexity. Secondly, since the signatures of the training sample are defined in each layer of the geospatial data cube, the separability of the training sample can be reduced if the layers are either incorrectly created or irrelevant to the selected thematic task. In turn, the low separability of the training sample leads to a decrease in classification accuracy [22].

In light of the above, optimization of the geospatial data cube is seen as a solution to the mentioned problems [23, 24]. Among approaches to reduce input data, the Principal Component Analysis (PCA) [25] and the Minimum Noise Fraction (MNF) [26] are the most widely used. There are also similar methods, for example, Noise-Adjusted Principal Components (NAPC) [27], Independent Component Analysis (ICA) [28], Non-Negative Matrix Factorization (NMF) [29] and Spatio-Spectral Decomposition (SSD) [30]. However, a common disadvantage of the considered approaches is that they do not consider the training sample's structure (in particular separability) and the selected classifier's specificity.

The presented study aims to enhance land cover classification by selecting the cube layers, the training sample separability of which will be the highest among other options. For this purpose, the optimization technique of the geospatial data cube was developed. It has two goals: enhancement of land cover classification and reduction of geospatial data cube size.

Hence, in the relevant sections of this article, the separability assessment of the training sample, the optimization technique of the geospatial data cube, and the experiment conducted to demonstrate the effectiveness of the developed technique are described.

2. Methods

This section presents the optimization technique of a geospatial data cube. Since this optimization is based on training sample separability, the objection function is the developed separability index of the training sample (SITS). Thus, the training sample separability assessment is also presented below as an algorithm for SITS calculation.

2.1. Assessment of the training sample separability

Separability is one of the training sample characteristics that affect classification accuracy. This characteristic shows the extent to which signatures representing different classes do not overlap. A low degree of separability is inherent in a high level of training sample mixing. In turn, this leads to a significant number of misclassified objects in the classification. Thus, the training sample separability is directly proportional to the classification accuracy.

The algorithm depicted in the flowchart(Figure 1) describes the separability assessment of the training sample.

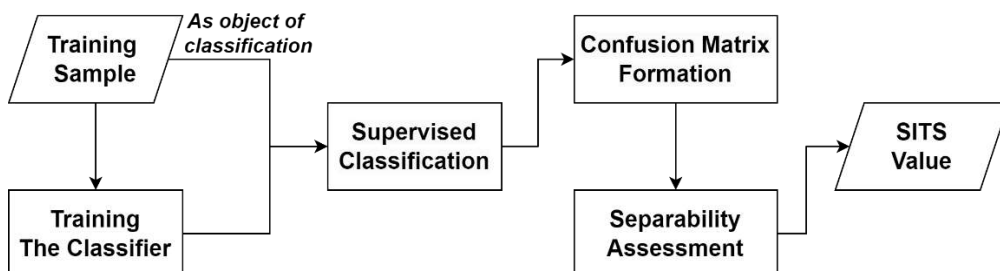


Figure 1: Algorithm of the separability assessment of the training sample

The first step implies classifier training by the training sample. Importantly, the supervised classification method must be the same as the one selected to classify the geospatial data cube further. Moreover, due to the proposed, separability depends on its structure (i.e., the set of layers) and the selected supervised classification method.

In the second step, the classifier is used to classify each signature from the training sample set.

The third step is the formation of the confusion matrix [31] for the classification obtained in the previous step.

The fourth and final step is calculating the SITS. This index quantifies the separability of training samples by measuring the ratio of correctly classified training samples to the total number of training samples. In other words, SITS equals the overall accuracy [31] based on the confusion matrix obtained in the previous step. The calculation of the SITS is shown in the following formula:

$$SITS = \frac{\sum_{i=1}^K x_{ii}}{N}, \quad (1)$$

where K is the number of classes, N is the total number of training sample signatures, x_{ii} is the number of class i signatures classified as class i (i.e. diagonal elements of the obtained confusion matrix that correspond to correctly classified signatures).

The values of the considered index range from 0 to 1. In this case, the value 0 shows that the training sample is entirely mixed (minimum separability), and the value 1 corresponds to the training sample, which is entirely separable (maximum separability).

2.2. Optimization technique

This technique is an optimization procedure that aims to reduce the number of layers of the geospatial data cube and increase the separability of the training sample, the signatures of which are defined in each layer of this cube. The objective function implies using the SITS. Thus, geospatial data cube optimization can be described as a search of the minor number of cube layers for which the training sample has the highest SITS value among all other sets of cube layers.

The flowchart of the technique algorithm is shown in Figure 2.

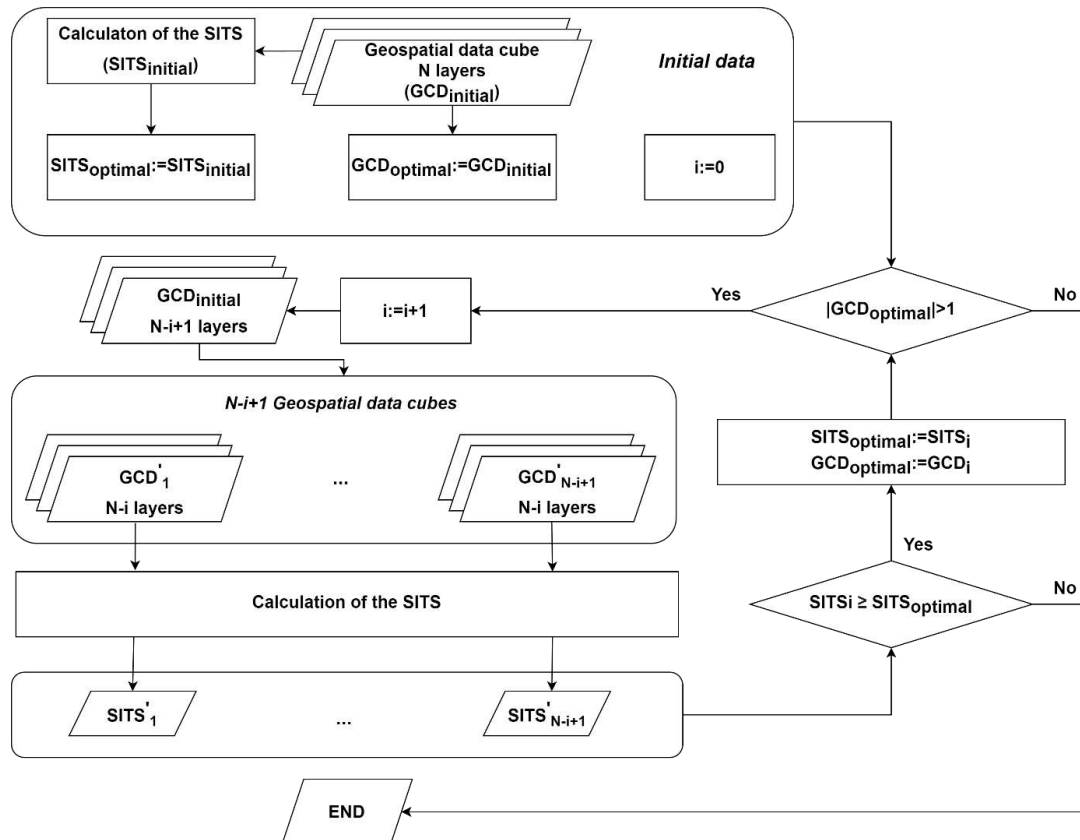


Figure 2: Algorithm of the optimization technique

The initial data and their characteristics will be introduced below for a detailed description of the technique algorithm.

Let the initial geospatial data cube have the following form:

$$GCD_{initial} = \{L_1, L_2, \dots, L_N\},$$

where L_i is the layer i of the geospatial data cube, and N is the total number of layers included in the initial geospatial data cube.

Then, as shown in Figure 2, the SITS value of the initial training sample, which has the signatures defined in each layer of the initial geospatial data cube, is first calculated. Let this value be $SITS_{initial}$.

Let us assign the values obtained above to the variables corresponding to the optimal set of layers of the geospatial data cube and the corresponding value of the SITS. Thus, we have:

$$GCD_{optimal} := GCD_{initial},$$

$$SITS_{optimal} := SITS_{initial}.$$

Next, an iterative procedure follows, in which the following steps are performed at each iteration.

Step 1. At the iteration i , the current geospatial data cube $GCD_{optimal}$, consisting of $N-(i-1)$ layers, is decomposed into $N-(i-1)$ cubes. Each newly formed cube is obtained by discarding one of the layers from the current cube. Then, each of the newly created geospatial data cubes will have the following form:

$$GCD'_1 = GCD - \{L_1\} = \{L_2, L_3, \dots, L_{N-(i-1)}\},$$

$$GCD'_2 = GCD - \{L_2\} = \{L_1, L_3, \dots, L_{N-(i-1)}\},$$

...

$$GCD'_{N-(i-1)} = GCD - \{L_{N-(i-1)}\} = \{L_1, \dots, L_{N-(i-1)-1}\}.$$

Since the number of layers decreases by one at each iteration, the obtained cubes will contain $N-i$ layers. Therefore, the following is valid:

$$|GCD'_1| = |GCD'_2| = \dots = |GCD'_{N-(i-1)}| = N - i.$$

Thus, the generated cubes can be written in the form of the following set:

$$C = \{GCD'_1, GCD'_2, \dots, GCD'_{N-(i-1)}\}.$$

Step 2. For each newly formed cube, the SITS value is calculated for the training sample, the signatures of which are defined in each cube layer. Then, the value of SITS for a particular cube GCD'_t will be denoted as $SITS'_t$. Thus, a set containing the value of the SITS for each newly formed cube will be obtained:

$$S = \{SITS'_1, SITS'_2, \dots, SITS'_{N-(i-1)}\}.$$

Step 3. Among the obtained cubes, the one with the highest value of the SITS is selected. Such a cube will be denoted as GCD_i . The selected cube can be expressed as follows:

$$GCD_i = \{GCD'_t \in C | SITS'_t = \max\{S\}\}.$$

$SITS_i$ denotes the value of the SITS of the geospatial data cube GCD_i .

Step 4. The values of the variables $SITS_i$ and $SITS_{optimal}$ are compared, and two options are considered:

1) if $SITS_i < SITS_{optimal}$, then the execution of the optimization algorithm will be interrupted, and further steps will be ignored. The optimal geospatial data cube will be the one obtained in the previous iteration, namely $GCD_{optimal}$. Accordingly, the SITS of the training sample of the corresponding cube has the value $SITS_{optimal}$.

2) if $SITS_i \geq SITS_{optimal}$, then the variable $SITS_{optimal}$ is assigned the value of the variable $SITS_i$, i.e.:

$$SITS_i := SITS_{optimal}.$$

Step 5. At this step, as at the previous one, two options are considered:

1) if the number of layers of the obtained cube $GCD_{optimal}$ is 1, i.e.:

$$|GCD_{optimal}| = 1,$$

then the execution of the optimization algorithm will be interrupted. The optimal geospatial data cube will be the one obtained at the current iteration, namely – $GCD_{optimal}$. Accordingly, the SITS value of the training sample of the corresponding cube is – $SITS_{optimal}$.

2) if the number of layers of the obtained geospatial data cube $GCD_{optimal}$ is greater than 1, i.e.:

$$|GCD_{optimal}| > 1,$$

then a new iteration will start, and the actions specified in **step 1** will be performed again.

The result of the described optimization procedure is the geospatial data cube with the set of layers that achieves the highest value of the SITS among all other considered sets. This geospatial data cube will be used for further land cover classification.

3. Experiment

The experiment was conducted to test the effectiveness of the developed technique. It consisted of carrying out a classification of the selected study area. For this purpose, an initial geospatial data cube with an excessive number of layers was formed. Then, the optimization technique was applied, resulting in an optimized cube. Finally, two classifications were obtained – before and after the optimization.

3.1. Study area

Since 2007, the Ukrainian network of test sites has provided validation and calibration of various remote sensing techniques and satellite-based products, including land cover classification [32]. The proposed technique was tested at the site within the Shatsk National Natural Park (SNNP). It is situated in the northwest of Ukraine, within Volyn' oblast, between 51° 28'25"N and 23° 49'29"E. The SNNP encompasses highly heterogeneous natural landscapes, like forests, peat bogs, transitional mires, meadows, and lakes.

The site comprises more than 100 georeferenced sample plots and gives comprehensive ground truth information about the representative landscapes of the West Polissia region (Figure 3).

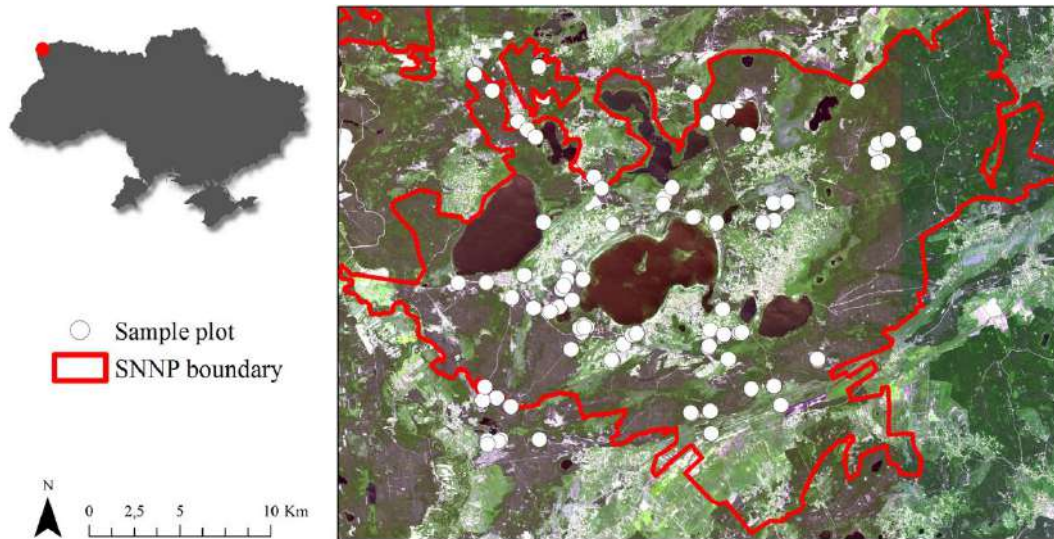


Figure 3: Location of the study area and sample plots within the Shatsk National Natural Park. The background is the true-colored composite of the Sentinel-2 Multispectral Instrument (MSI) image acquired on 1 June 2018

3.2. Training sample

Six broad land cover classes that characterize the study area were defined: artificial surfaces, tree-covered areas, grassland, agricultural areas, water bodies, and wetlands.

The given classes varied considerably both in spatial extent and heterogeneity. The smallest class of artificial surfaces included diverse features of built-up areas and transport units. While the biggest ones, like tree-covered areas and wetlands, included various sub-types that could still

be quite homogenous due to the big extent. Water bodies represent the most homogeneous class. Therefore, the number of training pixels of each class also varied disproportionally. The overall number of all training pixels accounted for 6474. Table 1 shows labels, descriptions, and training pixel amounts for the land cover classes assigned for the experiment.

Table 1
The classification scheme used in the experiment

#	Land Cover Class	Description	Training pixels
1	Artificial surfaces	Urban public and industrial built-up areas, transport units, and construction sites	319
2	Tree-covered areas	Broadleaved, coniferous, mixed and swamped forests, orchards, roadside tree lines	2313
3	Grasslands	Natural herbaceous vegetation, permanent grasslands of natural origin, pastures	634
4	Agricultural areas	Arable land, permanent crops, fallow lands, heterogeneous agricultural areas, open soils	887
5	Water bodies	Lakes, rivers and streams of natural origin, including man-made reservoirs and canals	634
6	Wetlands	Non-forested areas of peat bogs, transitional mires, eutrophic marshes, and reed beds	1687

3.3. Initial geospatial data cube

The experimental classification focuses on wetlands, the prevailing landscape of the test site and one of the most important for conservation within the SNNP. Evident differences in the seasonal development of wetlands and other vegetative land cover classes help distinguish them and require the application of multitemporal data [12].

The primary data source for forming the geospatial data cube was Sentinel-2 satellite imagery [33]. The images were selected for 4 dates (04.07.2018, 05.12.2018, 06.01.2018 and 10.14.2018) with minimal or no cloudiness.

Each Sentinel-2 image contains 13 spectral bands. At the preprocessing stage, atmospheric correction was performed for each image to eliminate the influence of the atmosphere and calculate the pixel values corresponding to the surface reflectance (bottom of atmosphere). During this procedure, 3 bands (B1, B9 and B10) that consider the effects of aerosols and water vapour on reflectance were removed. The spectral bands of the Sentinel-2 image have different spatial resolutions, namely 10, 20 and 60 meters.

Next, a complete set of normalized difference indices was calculated for each image. The following combinatorial formula is used to calculate such an index:

$$Index = \frac{b_i - b_j}{b_i + b_j}, i \neq j,$$

where b_i is the spectral band i .

This set could be presented in the following form:

$$NDI = \left\{ \frac{b_i - b_j}{b_i + b_j} \mid i, j \in \{1, 2, \dots, 10\}, i \neq j \right\}.$$

The cardinality of this set (i.e. number of normalized difference indices of one image) is calculated by the formula below:

$$|NDI| = C_n^m = \frac{n!}{m!(n-m)!} = \frac{10!}{2!(10-2)!} = \frac{8! * 9 * 10}{1 * 2 * 8!} = \frac{90}{2} = 45,$$

where variable n corresponds to the number of image bands, and m is the number of arguments in the index calculation formula.

As a result, 45 spectral indices were obtained for each image.

Another component of the input cube was the geomorphological data obtained from the ALOS PALSAR DEM [34]. In particular, this data contains the height above sea level and the slope. Their spatial resolution is 12.5 m.

All the above-described data must be spatially regularized to form the input geospatial data cube. It involves bringing all layers to the same spatial resolution, map projection, and size. With this in mind, all layers were scaled to a spatial resolution of 10 m, transformed to Universal Transverse Mercator Projection, Zone 34N (EPSG:32634) and resized so that all layers lie within the study area.

So, the input geospatial data cube contained 222 raster layers, namely 40 spectral bands of 4 different time Sentinel-2 satellite images, 180 corresponding spectral indices, and 2 raster layers of geomorphological parameters. The layers of the input cube are described in Table 2.

Table 2
Layers of the initial geospatial data cube

Date (DD/MM/YYYY)	Spectral bands	Spectral indices	Geomorphological data
07.04.2018	10	45	2
12.05.2018	10	45	
01.06.2018	10	45	
14.10.2018	10	45	

Hence, signatures of the initial training sample were assigned in each layer of the initial cube. To assess the separability of the training sample, the SITS was calculated using Formula 1:

$$SITS_{Initial} = \frac{\sum_{i=1}^K x_{ii}}{N} = \frac{6469}{6474} \approx 0.9992.$$

As seen above, this training sample had 5 misclassified signatures.

3.4. Optimized geospatial data cube

After applying the developed optimization technique, the geospatial data cube size was reduced from 222 to 40 layers. The selected layers are listed below in Table 3.

Table 3
Layer of the optimized geospatial data cube

Date (DD/MM/YYYY)	Spectral bands	Spectral indices	Geomorphological data
07.04.2018	0/10	4/45	1/2
12.05.2018	4/10	0/45	
01.06.2018	2/10	17/45	
14.10.2018	1/10	11/45	

Along with the cube optimization, the training sample signatures were reassigned according to the selected cube layers. The separability of that training sample was assessed by the SITS calculated below:

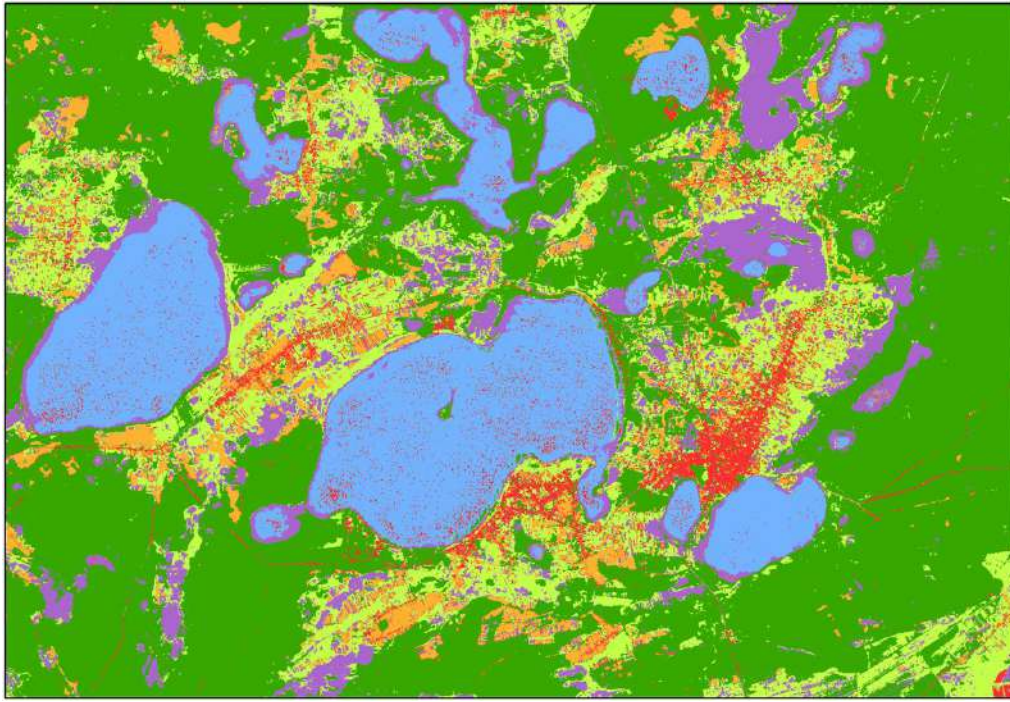
$$SITS_{Optimal} = \frac{\sum_{i=1}^K x_{ii}}{N} = \frac{6474}{6474} = 1.$$

This training sample had no misclassified signatures, so the optimized training sample was entirely separated.

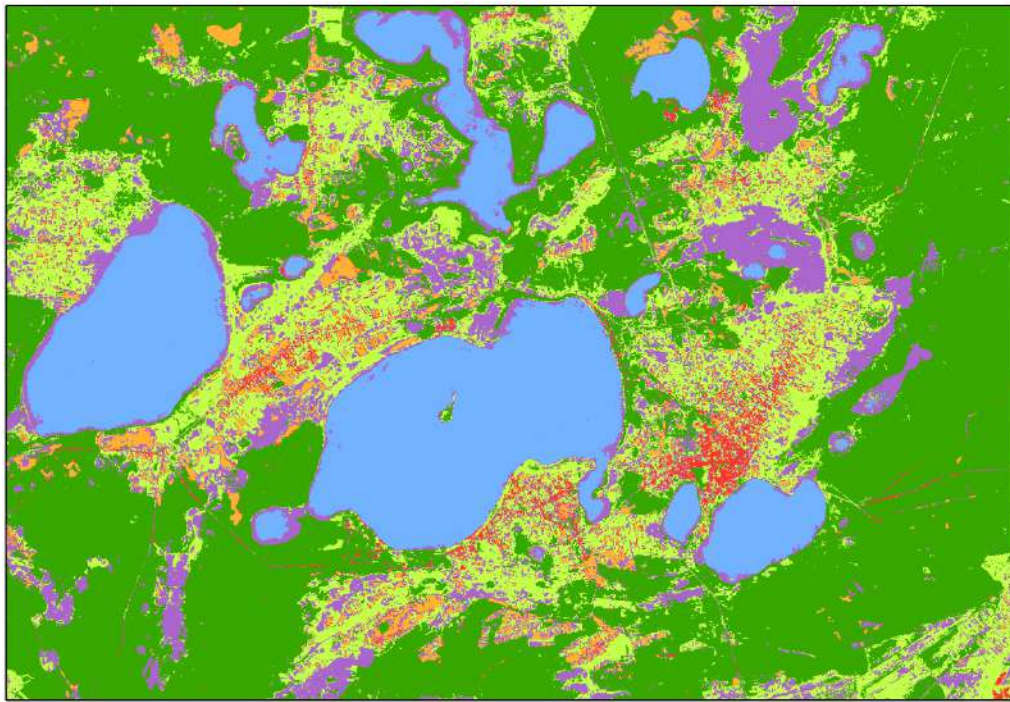
3.5. Land cover classifications

In order to test the effectiveness of the developed technique, the classifications using the initial cube and the optimized one were compared.

Firstly, the classification was obtained using the initial cube and the appropriate training sample. Then, the classification was obtained using the optimized cube and the appropriate training sample. These classifications are depicted in Figure 4.



a)



b)



Figure 4: Land cover maps of the study area were obtained using a) the initial cube and b) the optimized cube

The classifications above were obtained using Mahalanobis distance [35] as a supervised classification method. Exactly for this method, the separability assessment of the training sample was carried out for both initial and final geospatial data cubes.

3.6. Accuracy assessment

Classification accuracy assessment involved independent verification of initial and final land cover maps using proportionate stratified random samplings. This sampling technique produces sample set sizes directly related to the size of the classes and is widely used in assessing the classification accuracy of classes disproportionate in their extent. As the required sample size for a class, 0.01% of the total classified pixels of this class were analyzed. Thus, test sample sets were equal to 300 pixels for each land cover map. Satellite images (QuickBird) of high spatial resolution, available for 2018 in the Google Earth Pro app, were used for verification as reference data.

Table 4 shows the confusion matrix of the initial land cover map. In addition to the two dimensions ("Reference" and "Prediction"), this matrix shows metrics such as producer accuracy (PA) and user accuracy (UA) for each class [31].

Table 4
Confusion matrix of the initial land cover map

	Reference						Total (pixels)	UA
	1	2	3	4	5	6		
Prediction 1.Artificial surfaces	14	0	1	2	7	0	24	0,58
2.Tree- covered areas	0	134	0	0	1	0	135	0,99
3.Grasslands	0	3	29	5	0	1	38	0,76
4.Other lands	0	0	0	15	0	0	15	1
5.Water bodies	0	0	0	0	53	0	53	1
6.Wetlands	1	0	6	2	1	25	35	0,71
Total (pixels)	15	137	36	24	62	26	300	
PA	0,93	0,98	0,81	0,63	0,85	0,96		

The accuracy of the obtained land cover classification was assessed by indicators of overall accuracy and the kappa coefficient [31]. The overall accuracy value was calculated by the following formula:

$$OA = \frac{\sum_{i=1}^K x_{ii}}{N} = \frac{14 + 134 + 29 + 15 + 53 + 25}{300} = \frac{270}{300} = 0,9,$$

where K is the number of classes, N is the total number of test samples, x_{ii} is diagonal element i of the confusion matrix (i.e. number of correctly classified samples of class i). The value of the kappa coefficient was obtained following the calculations below:

$$Kappa = \frac{N * \sum_{i=1}^K x_{ii} - \sum_{i=1}^K (\sum_{j=1}^K x_{ij} * \sum_{j=1}^K x_{ji})}{N^2 - \sum_{i=1}^K (\sum_{j=1}^K x_{ij} * \sum_{j=1}^K x_{ji})} = \frac{300 * 270 - 24779}{300^2 - 24779} = \frac{56221}{65221} \approx 0.86.$$

Table 5 shows the confusion matrix of the final land cover map.

Table 5
Confusion matrix of the final land cover map

		Reference						Total (pixels)	UA
		1	2	3	4	5	6		
Prediction	1.Artificial surfaces	8	0	0	1	1	0	10	0,8
	2.Tree-covered areas	0	128	0	0	0	0	128	1
	3.Grasslands	0	6	40	1	0	0	47	0,85
	4.Other lands	1	0	0	14	0	0	15	0,93
	5.Water bodies	0	0	0	0	56	0	56	1
	6.Wetlands	1	1	1	5	0	36	44	0,82
	Total (pixels)	10	135	41	21	57	36	300	
	PA	0,8	0,95	0,98	0,67	0,98	1		

The same indicators were selected for the final land cover classification as for the initial one. Thus, the value of overall accuracy value is stated below:

$$OA = \frac{\sum_{i=1}^K x_{ii}}{N} = \frac{8+128+40+14+56+36}{300} = \frac{282}{300} = 0.94.$$

Then, the following calculations are for the kappa coefficient value:

$$Kappa = \frac{N * \sum_{i=1}^K x_{ii} - \sum_{i=1}^K (\sum_{j=1}^K x_{ij} * \sum_{j=1}^K x_{ji})}{N^2 - \sum_{i=1}^K (\sum_{j=1}^K x_{ij} * \sum_{j=1}^K x_{ji})} = \frac{300 * 282 - 24779}{300^2 - 24779} = \frac{59821}{65221} \approx 0.92.$$

4. Discussion

The aims of the developed optimization technique are stated as layers reduction of the geospatial data cube and enhancement of the classification. Thus, the experiment result should be considered in terms of these two aspects.

Firstly, the size of the optimized cube was 40 layers, whereas the initial one contained 222 layers. Therefore, the number of layers was reduced by 5.55 times.

Secondly, classification enhancement was evidenced by increasing indicators such as overall accuracy and the kappa coefficient. Namely, the overall accuracy increased by 0.04 from 0.9 to 0.94, and the kappa coefficient increased by 0.06 from 0.86 to 0.92. Since the classification focused on the wetlands, the accuracy of this class should be considered individually. Thereby, both user and producer accuracy of wetlands class were significantly increased, viz. by 0.11 from 0.71 to 0.82 and by 0.4 from 0.96 to 1, respectively.

5. Conclusion

This article presents an optimization technique to reduce geospatial data cube size and enhance land cover classification. The technique is based on the separability of the training sample, which is defined in each layer of the geospatial data cube. To assess the separability, the appropriate index (i.e. SITS) was developed and used as an object function in the technique frame. The

algorithm of the optimization technique implies stepwise band discarding to define the optimal set of the geospatial data cube layers. Such a set has the highest value of SITS among other options.

The conducted experiment implied techniques application to the land cover classification of the highly heterogeneous natural landscapes in the case of the Shatsky National Natural Park. This classification covered six land cover classes where wetlands are prevailing. The technique's effectiveness was approved by geospatial data cube reduction and classification accuracy enhancement, evidenced by the increase in such indicators as overall accuracy and kappa coefficient.

Further research should be aimed at technique application in other study areas and thematic tasks. Also, the separability assessment of the training sample could be extended by additional criteria. For example, the kappa coefficient could substitute overall accuracy as the basis of the developed separability index.

References

- [1] A.M. Melesse, Q. Weng, P.S. Thenkabail, G.B. Senay, Remote sensing sensors and applications in environmental resources mapping and modelling, *Sensors*, vol. 7, no. 12, pp. 3209–3241 (2007). doi: 10.3390/s7123209
- [2] A.H. Chughtai, H.U. Abbasi, İ. R. Karaş, A review on change detection method and accuracy assessment for land use land cover, *Remote Sensing Applications: Society and Environment*, vol. 22, p. 100482 (2021). doi: 10.1016/j.rsase.2021.100482
- [3] E. Zaitseva, S. Stankevich, A. Kozlova, I. Piestova, V. Levashenko, P. Rusnak, Assessment of the risk of disturbance impact on primeval and managed forests based on earth observation data using the example of Slovak Eastern Carpathians, *IEEE Access*, vol. 9, pp. 162847–162856 (2021). doi: 10.1109/access.2021.3134375
- [4] A. Kozlova, S. Stankevich, M. Svideniuk, A. Andreiev, Quantitative Assessment of Forest Disturbance with C-Band SAR Data for Decision Making Support in Forest Management, *Lecture notes on data engineering and communications technologies*, pp. 548–562 (2021). doi: 10.1007/978-3-030-82014-5_37
- [5] M.A. Popov, Topolnytskyi M.V., O.V. Titarenko, S.A. Stankevich, A.A. Andreiev, Forecasting gas and oil potential of subsoil plots via co-analysis of satellite, geological, geophysical and geochemical information by means of subjective logic, *WSEAS Transactions on Computer Research*, vol. 8, pp. 90–101 (2020). doi: 10.37394/232018.2020.8.11
- [6] M.A. Popov, S.A. Stankevich, S.P. Mosov, O.V. Titarenko, S.S. Dugin, S.I. Golubov, A.A. Andreiev, Method for Minefields Mapping by Imagery from Unmanned Aerial Vehicle, *Advances in Military Technology*, vol. 17, no. 2, pp. 211–229 (2022). doi: 10.3849/aimt.01722
- [7] A. Andries, S. Morse, R.J. Murphy, J.M. Lynch, E. Woolliams, J. Fonweban, Translation of Earth observation data into sustainable development indicators: An analytical framework, *Sustainable Development*, vol. 27, no. 3, pp. 366–376 (2018). doi: 10.1002/sd.1908
- [8] G. Scott, A. Rajabifard, Sustainable development and geospatial information: a strategic framework for integrating a global policy agenda into national geospatial capabilities, *Geospatial Information Science*, vol. 20, no. 2, pp. 59–76 (2017). doi: 10.1080/10095020.2017.1325594
- [9] M. Popov, S. Stankevich, Y. Kostyuchenko, A. Kozlova, Analysis of Local Climate Variations Using Correlation between Satellite Measurements of Methane Emission and Temperature Trends within Physiographic Regions of Ukraine, *International Journal of Mathematical, Engineering and Management Sciences*, vol. 4, no. 2, pp. 276–288 (2019). doi: 10.33889/ijmems.2019.4.2-023
- [10] O. Dubovyk, The role of Remote Sensing in land degradation assessments: opportunities and challenges, *European Journal of Remote Sensing*, vol. 50, no. 1, pp. 601–613 (2017). doi: 10.1080/22797254.2017.1378926
- [11] E. Agrillo, F. Filipponi, A. Pezzarossa, L. Casella, D. Smiraglia, A. Orasi, F. Attorre, A. Taramelli, Earth Observation and Biodiversity Big Data for forest habitat types classification and mapping, *Remote Sensing*, vol. 13, no. 7, p. 1231 (2021). doi: 10.3390/rs13071231
- [12] I. Dronova, P. Gong, L. Wang, L. Zhong, Mapping dynamic cover types in a large seasonally flooded wetland using extended principal component analysis and object-based

- classification, *Remote Sensing of Environment*, vol. 158, pp. 193–206 (2015). doi: 10.1016/j.rse.2014.10.027
- [13] P. Defourny, S. Bontemps, N. Bellemans, C. Cara, G. Dedieu, E. Guzzonato, O. Hagolle, J. Inglada, L. Nicola, T. Rabaute, M. Savinaud, C. Udrou, S. Valero, A. Bégué, J. Dejoux, A. Harti, J. Ezzahar, N. Kussul, K. Labbassi, V. Lebourgeois, M. Zhang, T. Newby, A. Nyamugama, N. Salh, A. Shelestov, V. Simonneaux, P. Traoré, S. Traoré, B. Koetz, Near real-time agriculture monitoring at national scale at parcel resolution: Performance assessment of the Sen2-Agri automated system in various cropping systems around the world, *Remote Sensing of Environment*, vol. 221, pp. 551–568 (2019). doi: 10.1016/j.rse.2018.11.007
- [14] S. Fritz, I. McCallum, L. You, A. Bun, E. Moltchanova, M. Duerauer, F. Albrecht, C. Schill, C. Perger, P. Havlík, A. Mosnier, P. Thornton, U. Wood-Sichra, M. Herrero, I. Becker-Reshef, C. Justice, M. Hansen, P. Gong, S. Aziz, M. Obersteiner, Mapping global cropland and field size, *Global Change Biology*, vol. 21, no. 5, pp. 1980–1992 (2015). doi: 10.1111/gcb.12838
- [15] R. Sathya, A. Abraham, Comparison of supervised and unsupervised learning algorithms for pattern classification, *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2 (2013). doi: 10.14569/ijarai.2013.020206
- [16] D. Montero, G. Kraemer, A. Anghel, C.A. Camacho, G. Brandt, G. Camps-Valls, F. Cremer, I. Flik, F. Gans, S. Habershon, C. Ji, T. Kattenborn, L. Martínez-Ferrer, F. Martinuzzi, M. Reinhardt, M. Söchting, K. Teber, M. Mahecha, Data Cubes for Earth System research: Challenges ahead, *EarthArXiv (California Digital Library)* (2023). doi: 10.31223/x58m2v
- [17] M. Schramm, E. Pebesma, M. Milenković, L. Foresta, J. Dries, A. Jacob, W. Wagner, M. Mohr, M. Neteler, M. Kadunc, T. Miksa, P. Kempeneers, J. Verbesselt, B. Gößwein, C. Navacchi, S. Lippens, J. Reiche, The OpeNEO API—Harmonizing the use of earth observation cloud services using virtual Data Cube functionalities, *Remote Sensing*, vol. 13, no. 6, p. 1125 (2021). doi: 10.3390/rs13061125
- [18] L.M. Estupiñán-Suárez, F. Gans, A. Brenning, V.H. Gutierrez-Velez, M.C. Londoño, D.E. Pabon-Moreno, G. Poveda, M. Reichstein, B. Reu, C. Sierra, U. Weber, M.D. Mahecha, A Regional Earth System Data Lab for Understanding Ecosystem Dynamics: An Example from Tropical South America, *Frontiers in Earth Science*, vol. 9 (2021). doi: 10.3389/feart.2021.613395
- [19] T. Hermosilla, M.A. Wulder, J.C. White, N.C. Coops, Land cover classification in an era of big and open data: Optimizing localized implementation and training data selection to improve mapping outcomes, *Remote Sensing of Environment*, vol. 268, p. 112780 (2022). doi: 10.1016/j.rse.2021.112780
- [20] J.R.M. Flórez, I. Lizarazo, Land cover classification at three different levels of detail from optical and radar Sentinel SAR data: a case study in Cundinamarca (Colombia), *Dynacolombia*, vol. 87, no. 215, pp. 136–145 (2020). doi: 10.15446/dyna.v87n215.84915
- [21] H. Li, J. Cui, X. Zhang, Y. Han, L. Cao, Dimensionality reduction and classification of hyperspectral remote sensing image feature extraction, *Remote Sensing*, vol. 14, no. 18, p. 4579 (2022). doi: 10.3390/rs14184579
- [22] I. Piestova, A. Kozlova, A. Andreiev, J. Rabcan Local Quality Improvement of Multispectral Imagery Classification with Radiometric-spatial Feedback. *Computer Modeling and Intelligent Systems* 2864:158–168 (2021). <https://doi.org/10.32782/cmris/2864-14>
- [23] F. Luo, L. Zhang, B. Du, L. Zhang, Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5336–5353 (2020).
- [24] H. Huang, G. Shi, H. He, Y. Duan, F. Luo, Dimensionality reduction of hyperspectral imagery based on spatial-spectral manifold learning, *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2604–2616 (2019).
- [25] N. Salem, S. Hussein, Data dimensional reduction and principal components analysis. *Procedia Computer Science*, 163, 292–299 (2019). <https://doi.org/10.1016/j.procs.2019.12.111>
- [26] A. Green, M. Berman, P. Switzer, M. Craig, A transformation for ordering multispectral data in terms of image quality with implications for noise removal. *IEEE Transactions on Geoscience and Remote Sensing*, 26(1), 65–74 (1988). <https://doi.org/10.1109/36.3001>

- [27] C. Chang, Q. Du, Interference and noise-adjusted principal components analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 37(5), 2387–2396 (1999). <https://doi.org/10.1109/36.789637>
- [28] A. Hyvärinen, J. Karhunen, E. Oja, *Independent Component Analysis*. John Wiley & Sons. (2001).
- [29] D. D. Lee, H. S. Seung, Algorithms for Non-negative Matrix Factorization. In *Advances in Neural Information Processing Systems 13 NIPS 2000*, 556-562 (2001).
- [30] X. Kong, Y. Zhao, J. Chan, J. Xue, Hyperspectral image restoration via Spatial-Spectral residual total variation regularized Low-Rank tensor decomposition. *Remote Sensing*, 14(3), 511 (2022). <https://doi.org/10.3390/rs14030511>
- [31] M.O. Popov, Methodology of accuracy assessment of classification of objects on space images, *Journal of Automation and Information Sciences*, vol. 39, pp. 1–10 (2007). doi: 10.1615/JAutomat Inf Scien.v39.i1.50
- [32] V.I. Lyalko, M.A. Popov, S.A. Stankevich, J.I. Zelyk, S.V. Cherny, Calibration/Validation Test Sites in Ukraine: current state and directions of further research and development, *Ukrainian Metrological Journal*, vol. 2, pp. 15-26 (2014).
- [33] I. Shurmer, F. Marchese, J.-M. Morales-Santiago, P. P. Emanuelli, Sentinels Optical Communications Payload (OCP) Operations: From Test to In-Flight Experience, Paper Presentation, 2018 SpaceOps Conference, p. 24. doi: 10.2514/6.2018-2654
- [34] J. S. Stewart, Combining satellite data with ancillary data to produce a refined land-use/land-cover map, pp. 1-10 (1998). doi: 10.3133/wri974203
- [35] P. C. Mahalanobis, On the Generalized Distance in Statistics, *Proceedings of the National Institute of Sciences of India*, vol. 2, no. 1, pp. 49–55 (1936).

Assessing Generative Pre-trained Transformer 4 in Clinical Trial Inclusion Criteria Matching

Ivan Dyyak¹, Vitaliy Horlatch¹, Tymofii Pasichnyk¹, and Vasyl Pasichnyk¹

¹ Ivan Franko National University of Lviv, 1 Universytetska St., Lviv, 79000, Ukraine

Abstract

Clinical trials play a very important role in improving the quality and expectancy of life. At the same time, this is a complicated process which requires a lot of time, effort, and costs. Effective interpretation and understanding of inclusion and exclusion criteria are key elements for streamlining trial documentation, enhancing patient engagement, and facilitating data analysis. The nuances in clinical criteria can significantly affect trial outcomes and participant selection across different linguistic and cultural backgrounds. The aim of this study is to assess the capability of the Generative Pre-trained Transformer 4 (GPT-4) model in comprehending and matching clinical trial inclusion criteria between English and Ukrainian, focusing on the model's potential to automate document processing and improve patient engagement in a multilingual context. The model was tested on 315 inclusion criteria from 15 clinical trials from ClinicalTrials.gov and their corresponding Ukrainian versions from ClinicalTrials.dec.gov.ua. The GPT-4 model accurately identified and matched 91% of the criteria pairs fully and demonstrated a detailed understanding of clinical language and nuances. An additional 4% of pairs were identified partially. All together it resulted in 95% accuracy for the tasks which accept partial matches. These findings indicate the model's substantial capacity for understanding complex medical and clinical terminologies and for facilitating the automation of clinical documents. Moreover, the experiment has already helped to identify several cases where the English and the Ukrainian versions had significant meaningful differences, most likely caused by human mistakes. The results of the study open a wide range of opportunities for automating clinical documentation processes and enhancing patient engagement through modern technologies. The scientific novelty is in applying Natural Language Processing (NLP) to understand and process clinical trial criteria in different languages, in this case Ukrainian, leading to more automated and patient-centered approaches in clinical research.

Keywords

Natural Language Processing, Generative Pre-trained Transformer, GPT-4, Clinical Trial Matching, Inclusion Criteria

1. Introduction

The increasing globalization of clinical trials presents significant challenges and opportunities in clinical research. One of the cases is related to the accessibility of trial participation in different regions and landscapes. Typically, clinical trials are conducted in many different countries. According to the regulations, all the related documentation should be translated into the corresponding languages. Also, in most cases, local Electronic Health Records (EHR) keep data in local languages too.

The use of automated tools and natural language processing (NLP) technologies is becoming increasingly common in addressing the complexities of multilingual documentation in clinical trials. Despite advancements, there remains a significant gap in the understanding and application of automated tools related to clinical trial criteria matching, especially across different languages. This gap not only prevents the efficiency of global trial operations but also impacts patient engagement and onboarding, which are crucial for the ethical and effective conduct of clinical research.

The object of this study is the process of interpreting and understanding clinical trial inclusion criteria in a multilingual context using computer sciences. The subject covers the methods and

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ ivan.dyyak@lnu.edu.ua (I. I. Dyyak); vitaliy.horlatch@lnu.edu.ua (V. M. Horlatch); tymofii.pasichnyk@lnu.edu.ua (T. V. Pasichnyk); vasyi.pasichnyk.apmi@lnu.edu.ua (V. T. Pasichnyk)

🆔 0000-0001-5841-2604 (I. I. Dyyak); 0000-0001-5401-1731 (V. M. Horlatch); 0000-0002-1495-7423 (T. V. Pasichnyk); 0009-0004-3738-6629 (V. T. Pasichnyk)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

technologies which are used to understand and match these criteria across languages, with a particular focus on the English and Ukrainian language pair, assuming that it can be generalized and extended to other languages too.

The goal of this work is to analyze and validate the use of artificial intelligence and the Generative Pre-trained Transformer 4 (GPT-4) in particular to enhance the understanding and application of clinical trial inclusion criteria in different linguistic contexts. This involves identifying, comparing, and assessing the equivalence of clinical trial criteria between English and Ukrainian, aiming to improve document automation processes, facilitate more effective patient engagement, and streamline data analysis procedures.

The importance of this research originated from the ongoing need to improve inclusivity and accessibility in global clinical trials, ensuring that non-English speaking populations are adequately represented and can participate fully and safely in clinical research. By leveraging advanced NLP tools like GPT-4, this study addresses a critical gap in current research methodologies, offering a potential for significant advancements in clinical trial design, execution, and analysis. The appropriateness of this article within the relevant field is underscored by its contribution to improving clinical trial efficiency and participant comprehension, ultimately promoting more fair and effective global health research.

2. Problem Statement

The core problem addressed in this study involves the accurate understanding and matching of clinical trial inclusion criteria across different languages, specifically English and Ukrainian. This problem can be formalized mathematically as follows.

Input Variables:

- A set of inclusion criteria in English extracted from ClinicalTrials.gov, which can be represented by the formula

$$I_E = \{e_1, e_2, \dots, e_n\} \quad (1)$$

where each e_i represents a specific criterion.

- A corresponding set of inclusion criteria in Ukrainian obtained from ClinicalTrials.gov.ua, which can be represented by the formula

$$I_U = \{u_1, u_2, \dots, u_n\} \quad (2)$$

where each u_i represents a specific criterion in Ukrainian.

Output Variables:

- A set of matched pairs between English and Ukrainian inclusion criteria:

$$M = \{(e_i, u_j) | e_i \in I_E, u_j \in I_U\} \quad (3)$$

where each pair consists of one English criterion and one Ukrainian criterion.

- A set of scores indicating the degree of match between pairs in M:

$$S = \{s_1, s_2, \dots, s_n\} \quad (4)$$

where each s_i corresponds to the semantic similarity between e_i and u_i .

Results Quality Evaluation Criteria:

- Accuracy (Acc) – The proportion of correctly matched criteria pairs in M to the total number of criteria pairs. It can be calculated by the formula:

$$Acc = \frac{|\{(e_i, u_j) \in M \mid s_i \text{ indicates a correct match}\}|}{n} \quad (5)$$

- Precision (P) – The proportion of true positive matches to the total number of positive matches identified by the model.
- Recall (R) – The proportion of true positive matches to the total number of actual positive matches across the datasets.

Necessary Definitions:

- Positive match identified by the model – a pair of criteria that the model identifies as a match, regardless of whether this match is correct.
- True positive match – a pair of criteria that are correctly identified as equivalent by the model.
- Actual positive match – a pair of criteria that an expert identifies as a match.
- False positive match – a pair of criteria that are identified as equivalent by the model but in fact is not a match.
- False negative match – a pair of criteria that are identified by the model as not a match but in fact is a match.

3. Review of the literature

The application of Artificial Intelligence (AI) and Natural Language Processing (NLP) in the field of clinical trials is a rapidly emerging area [1]. It is often being discussed in modern scientific and business conferences and a considerable number of scientific articles have been published. If properly used, AI can significantly enhance operational efficiencies, patient engagement, and the accuracy of data interpretation. Furthermore, the capacity of AI to innovate and refine the design of clinical trials is recognized as a potential catalyst for increasing the efficacy and efficiency of trial methodologies [2]. The advancements in these technologies underscore their potential to transform the landscape of clinical research by optimizing trial procedures and ensuring more targeted and effective patient engagement strategies.

A comprehensive overview of current capabilities and limitations of AI applications in clinical trials was provided, underscoring the transition from artificial to applied intelligence and highlighting the importance of these technologies in enhancing trial outcomes [3]. The transformative role of machine learning in clinical research focuses on the potential to automate and optimize various aspects of trial execution and evidence generation, aligning with findings that NLP can enhance the standardization and utilization of clinical data [4–10].

One of the key elements of each clinical trial is eligibility criteria. An accurate and comprehensive understanding of inclusion and exclusion criteria is fundamental to ensuring that the proper patients are enrolled. These criteria not only define the scope and structure of a trial but also ensure the safety and appropriateness of participants. Misinterpretations or miscommunications in these criteria can lead to incorrect selection of participants, regulatory issues, and even trial failure or health consequences for the participants [11, 12].

The need for precise matching of patients to trials based on inclusion and exclusion criteria is a complicated process. It requires a lot of manual effort and collaboration between different organizations and electronic systems. At the same time, considering that nowadays most of the information is tracked in electronic records, this process has many opportunities to be automated. The growing potential of NLP systems plays a critical role in streamlining the patient recruitment process [13].

However, a significant gap exists in the application of NLP technologies for understanding and matching clinical trial inclusion criteria across different languages. Research into NLP tasks in clinical settings has not specifically addressed the challenges of multilingual criteria interpretation, indicating a need in clinical research where language barriers can prevent participant enrollment and engagement [14].

Recent studies have explored the capabilities of GPT-4, a large language model, in clinical settings, demonstrating its potential in outperforming traditional methods and medical

professionals in certain tasks. These findings provide insights into its language understanding capabilities and suggest promising opportunities for utilizing GPT-4 in the interpretation and application of clinical trial criteria, potentially overcoming language and comprehension barriers [15–20].

In summary, while significant advancements have been made in AI and NLP applications within clinical research, a critical gap remains in the context of multilingual inclusion criteria interpretation. This study aims to bridge this gap by leveraging the capabilities of GPT-4, thus contributing to the existing body of knowledge and addressing the identified limitations. The research underscores the ongoing evolution of AI in medicine, aiming to enhance the accuracy, efficiency, and inclusivity of clinical trial processes [21].

4. Materials and Methods

The theoretical foundation of this research is grounded in artificial intelligence and machine learning, particularly in the context of natural language processing (NLP) and machine translation (MT). Even though, there are special models tuned for medicine (e.g. PubMedBERT, BioGPT, Med-PaLM, etc.), recent studies have shown that GPT-4 can outperform them [22]. Therefore, we chose the GPT-4 model to analyze and compare inclusion criteria across different languages in our study. The hypothesis posits that the GPT-4 model, trained on diverse linguistic datasets, can accurately match and assess the equivalence of clinical trial inclusion criteria in English and Ukrainian.

The research employs semantic analysis techniques inherent to GPT-4 to interpret and compare the meaning of text segments from the inclusion criteria. These techniques involve the decomposition of text into smaller, manageable units (bullets), enabling a detailed, point-by-point comparison. The theoretical significance lies in assessing the model's ability to understand and translate medical and clinical terminology accurately, a critical factor given the nuanced nature of clinical trial criteria.

The methodological approach encompasses several key components:

- **Data Preparation:** In order to get a proper data set which will be close to the real-world scenarios, the following subsets were prepared and mixed:
 1. Inclusion criteria of the same clinical trial in English and Ukrainian were taken from the corresponding sites without any changes. Even though they had a similar meaning, the text could be several pages long and semantically structured in a completely different way.
 2. The inclusion criteria were extracted and segregated into individual bullets. The number of bullets in the English and Ukrainian versions could differ. As well as their content.
 3. Similar to the above however English and Ukrainian criteria were intentionally taken from different clinical trials

This step ensured that the model was evaluated on different conditions, enriching the usability and applicability of the study.

- **Model Selection:** We utilized the GPT-4 model for identifying and comparing criteria pairs. This involved configuring the model to process pairs of text bullets and assess their semantic equivalence across languages. The model's performance metrics were predetermined to evaluate its accuracy in identifying exact matches, partially correct matches, and discrepancies.
- **Analytical Techniques:** The study applied quantitative methods to analyze the model's output, focusing on the proportion of fully correct, partially correct, and incorrect matches. Statistical analysis was used to assess the model's accuracy and reliability in translating and matching criteria across languages.
- **Validation Process:** To ensure the validity of the research results, a subset of the GPT-4-identified pairs was manually reviewed by an expert. This step served to cross-verify the accuracy of the automated assessments and to identify potential areas for model improvement.

The research was conducted in a sequential manner, beginning with the collection and preparation of data, followed by the deployment of the GPT-4 model, preparation of

corresponding prompts, data processing, analysis, and validation of the results. The important step was related to the manual assessment and interpretation of the results. This sequence was designed to ensure a logical progression of tasks and to build a coherent dataset for analysis.

The selection of the GPT-4 model as the primary research tool was justified by its advanced language processing capabilities and its proven effectiveness in similar linguistic tasks. Nowadays, it is easily accessible by different companies and organizations. That provides a significant boost for further research and development in this area. The use of statistical analysis for evaluating the model's performance provided a quantitative basis for assessing accuracy, reliability, and validity.

5. Experiments

The experiment was started from the data collection. We created the dataset which contained 315 inclusion criteria extracted from 15 distinct clinical trials. Each criterion had up to 400 words. English criteria were collected from publicly available records on ClinicalTrials.gov [23], a database of privately and publicly funded clinical studies conducted around the world. The corresponding Ukrainian translations were obtained from ClinicalTrials.dec.gov.ua, the official Ukrainian counterpart hosting clinical trial information.

As described in the section above, several types of data collection had a place. It is worth mentioning that even though typically inclusion criteria in the English and Ukrainian versions should be the same, in reality, they might have many additional details related to local regulations, measurement details, etc. For example, the English version of a criterion might look like “Has adequate organ function within 7 days prior to the start of study treatment” while the Ukrainian version will contain a detailed list of laboratory values to be met in order to be evaluated as “adequate organ function”.

For each clinical trial, was created a separate but exactly the same (except the input parameters) prompt to establish proper isolation. The prompt contained the instructions, a numerated set of inclusion criteria in English, and a numerated set of inclusion criteria in Ukrainian. The preprocessed inclusion criteria were inputted into the OpenAI GPT-4 model hosted on an isolated Microsoft Azure instance. The model's temperature was set to 0 to avoid randomness, and enhance the model's focus and predictability.

The model's task was to identify corresponding pairs of criteria between the English and Ukrainian datasets and then to assess whether the pairs matched completely or exhibited any meaningful differences. The outputs were systematically recorded, categorizing each pair into 'Full Match', 'Partial Match', 'No Ukrainian equivalent', 'No English equivalent'.

All outputs were collected together in a single table. Each result was manually reviewed by experts to validate the accuracy of the model's assessments. Then the data was summarized in different views and the corresponding parameters were calculated by the formulas defined in the “Problem Statement” section.

This setup ensures that the experiments can be replicated by other researchers or automated tools, assuming they use the same datasets and GPT-4 model configuration. It provides a clear and systematic approach to evaluating NLP models' accuracy in cross-linguistic analysis of clinical trial inclusion criteria.

6. Results

The summary of the results of the experiments is presented in the Table 1.

Table 1
Summary of the results

	GPT-4	Manual assessment
Full match	84	69
Partial match	30	38
Significant difference	4	12
No Ukrainian equivalent	16	15
No English equivalent	63	63
Total	197	197

Graphically they can be illustrated in the Fig. 1.

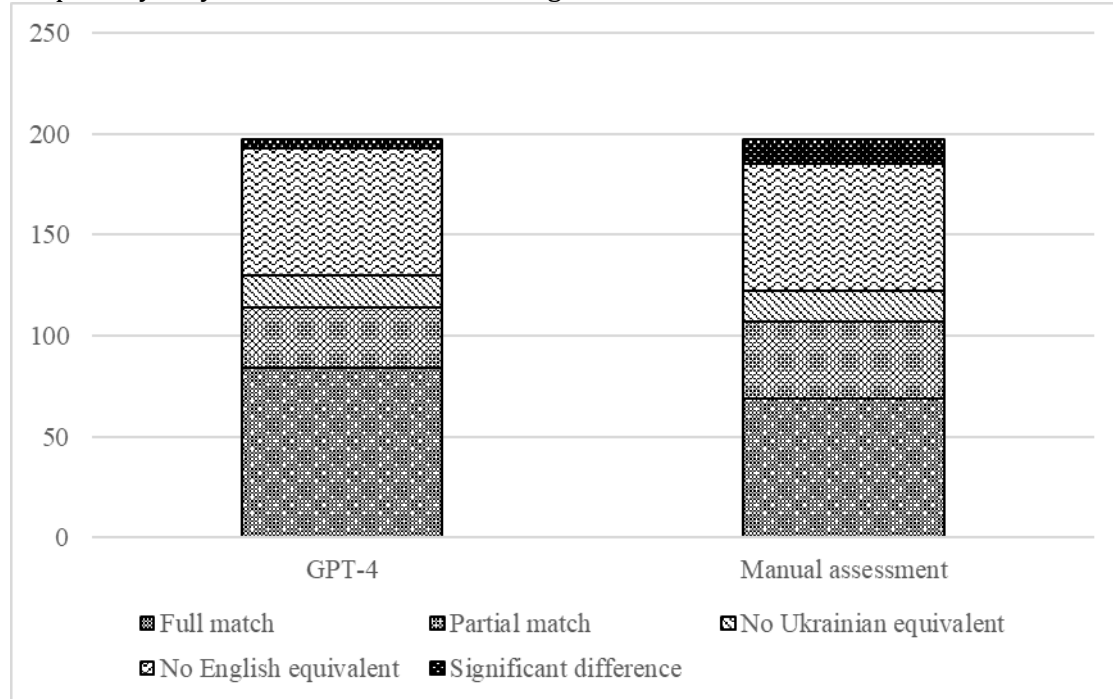


Figure 1: Graphical representation of the results.

Then we structured the data like in the Table 2 for proper accuracy, precision, and recall calculations. These metrics were selected for their relevance in evaluating the performance of NLP models, particularly in tasks involving text matching and semantic analysis.

Table 2
Summary of the results validation

Parameter	Value
True positive matches:	97
Total number of positive matches identified by the model:	114
Total number of actual positive matches across the datasets:	107

Based on that, we got the following results:

- **Accuracy (Acc).** The model achieved an overall accuracy rate of 91% in matching the correct pairs of inclusion criteria between the two languages. This measure indicates the proportion of all criteria pairs for which the model's predictions were correct to the total number of pairs.
- **Precision (P).** The model demonstrated a precision rate of 85%, reflecting the proportion of true positive matches to all matches identified by the model as correct.
- **Recall (R).** The recall rate obtained in the study was 91%. This metric measures the proportion of true positive matches to actual positive matches that the model successfully identified.

Since it is difficult to define the exact difference between “Full match” and “Partial match”, and since for some classes of tasks it is important just to identify whether there is a match or not, it is also worth looking at the results combining “Full match” and “Partial match” together and considering them as a “Match”. In this case, the data will look like the following.

Table 3
Summary of the results without the breakdown into “Full match” and “Partial match”

	GPT-4	Manual assessment
Match	114	107
Significant difference	4	12
No Ukrainian equivalent	16	15
No English equivalent	63	63
Total	197	197

Graphically this can be illustrated in the Fig. 2.

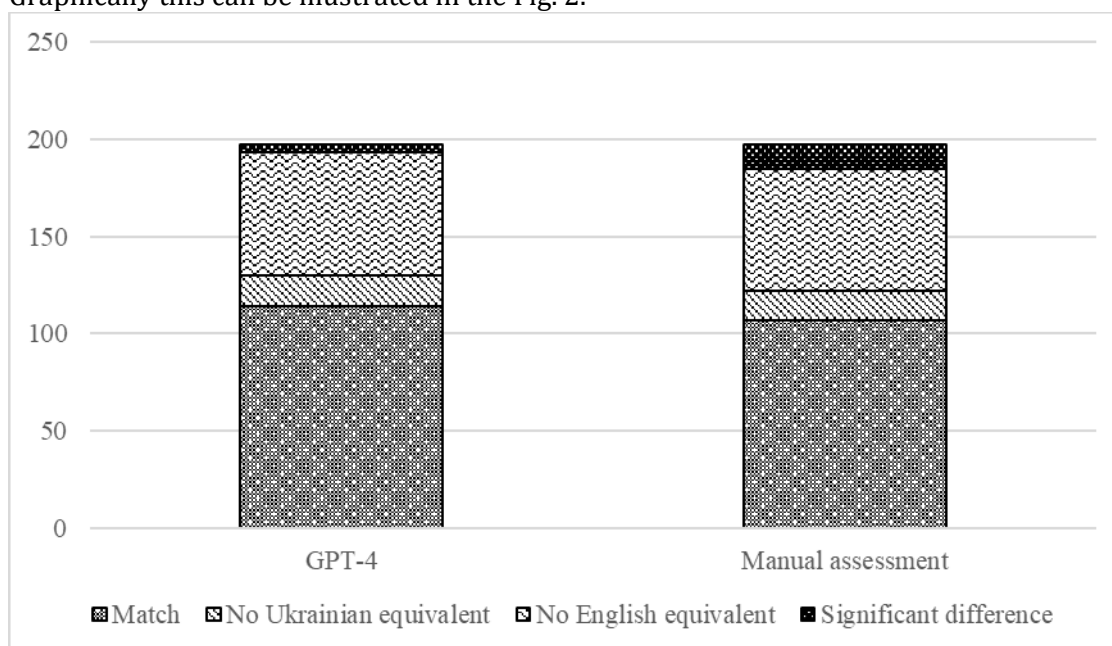


Figure 2: Graphical representation of the results without the breakdown into “Full match” and “Partial match”.

The summary of the resulting calculation is represented in the Table 4.

Table 4
Summary of the results validation without the breakdown into “Full match” and “Partial match”

Parameter	Value
True positive matches:	106
Total number of positive matches identified by the model:	114
Total number of actual positive matches across the datasets:	107

Accuracy (Acc), precision (P), and recall (R) in this case will be the following:

- Acc = 95%
- P = 93%
- R = 99%

Even though the analysis of the incorrect results is not in the scope of this research, we decided to attach the correlation between the length of a criterion in the English and the Ukrainian versions and the results. It is assumed that longer texts may contain more nuances, which could either increase or decrease the model's ability to find a correct match based on linguistic and semantic complexity. We calculated the length of each inclusion criterion by counting the words

in both the English and Ukrainian texts. This correlation is represented in Figure 3. Even though no correlation is visually noticed, this data can be used for further discussions and research.

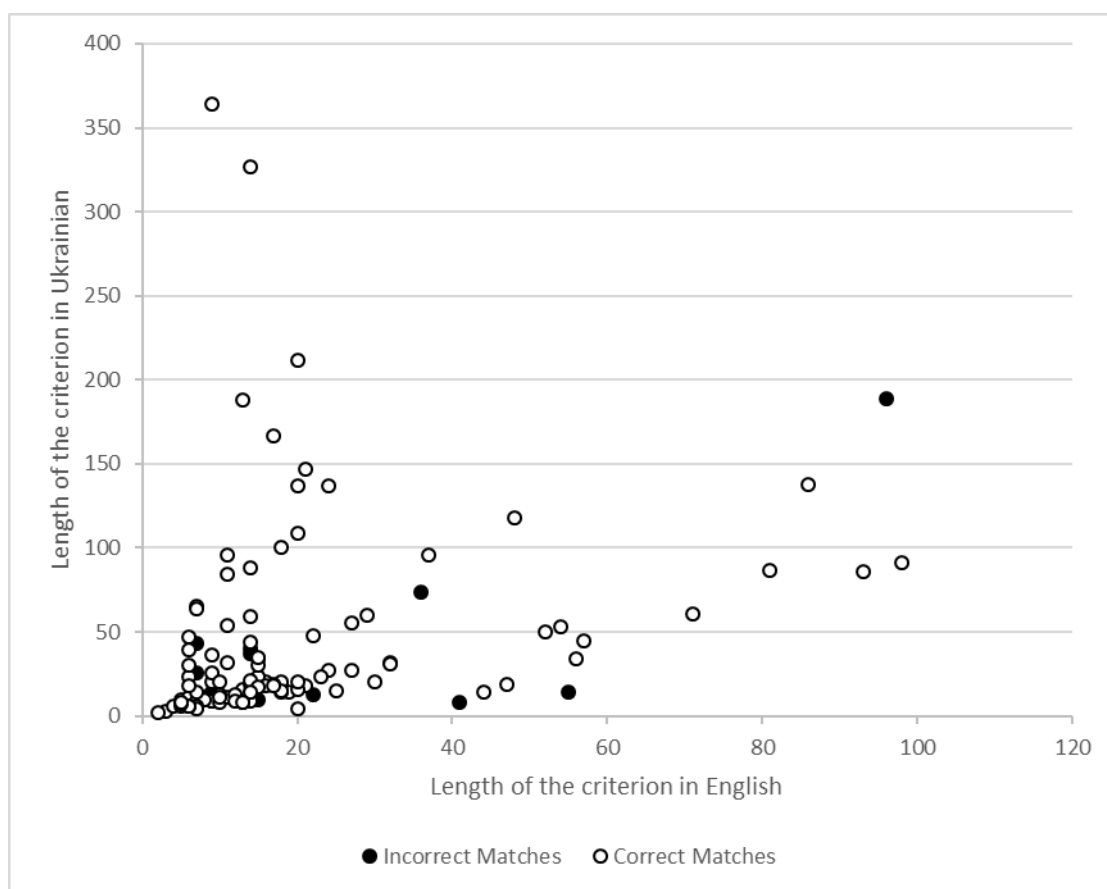


Figure 3: The correlation between the length of a criterion and the results.

In addition to the scientific observations, as a result of the experiments, the model identified four cases where unexpected significant meaningful differences had a place. In the first case, the English version of the inclusion criteria had the following text: “Full recovery from cystectomy and enrollment within 24 weeks following cystectomy.”. While in the Ukrainian version, 14 weeks were mentioned. In the second example, patients with tuberculosis were specifically mentioned in the English version of the exclusion criteria but not in the corresponding Ukrainian version. In the third and the fourth, the Ukrainian version included additional screening requirements for hepatitis B and C, which were not mentioned in the English version provided.

7. Discussion

The findings from this study illustrate the advanced capabilities of the GPT-4 model in understanding and accurately matching clinical trial inclusion criteria across languages, specifically between English and Ukrainian. The high performance in terms of accuracy, precision, and recall indicates a notable advancement in the application of natural language processing technologies in clinical research. This opens a variety of opportunities for further research and development.

GPT-4 model's language understanding capabilities demonstrate not only the validity but also highlight its potential to mitigate linguistic barriers in global clinical trials. For the cases where a deep understanding of the eligibility criteria is required, the success rate of 91% in accurately matching criteria pairs between English and Ukrainian shows great potential in the model to be used. The precision rate of 85% indicates a high level of specificity in the model's output, reducing the likelihood of false positives and thereby enhancing the reliability of trial participant matching. Furthermore, the recall rate of 91% suggests that the model effectively captures almost all of the

relevant matches, i.e. minimizing the risk of overlooking suitable trial participants due to language variations.

For the cases which do not require a very deep understanding of the eligibility criteria, the results are even better. The accuracy of 95%, precision of 93%, and recall of 99% indicate the readiness of the model to be used in real-life cases. For instance, in finding discrepancies in clinical trials' documentation in different languages. Even within the scope of this research, several such cases have been identified. Another example is matching which will be followed by a human validation.

Additional practical implications of this research are diverse. It can be used for automation and validation of clinical documentation, clinical trial recruitment, patient comprehension and engagement, adherence to regulatory compliance, and others. Also, it can be used for integrations between digital health systems. By leveraging the capabilities of advanced NLP technologies, this research contributes to making clinical research more efficient, inclusive, and accessible.

The promising results of this study justify further investigation into the application of artificial intelligence in clinical research. Future studies could explore the model's applicability across a broader range of languages and medical specializations to understand its full potential and limitations. This can lead to more inclusive, diverse, and representative clinical research, contributing to the advancement of global health outcomes.

Additional analysis of the incorrect matches, explorations of correlation between different parameters and matching results, open new opportunities for further research. The complexity of trial inclusion and exclusion criteria could impact the effectiveness of AI tools in clinical settings. This observation could lead to future AI model training processes, emphasizing the need for diverse and comprehensive datasets that combine various clinical documents in different languages.

Even though the study has demonstrated great results even using a general GPT-4 model and it was confirmed by other studies, it is worth keeping in mind special models which are tuned for medicine (e.g. PubMedBERT, BioGPT, Med-PaLM 2, BioBERT, and BioMegatron, etc.). Obviously, for some of the cases, their usage will be more feasible and relevant. The upcoming release of GPT-5 is another intriguing and anticipated event which will be very interesting for further research.

In conclusion, it is important to keep in mind ethical considerations. Patient data, especially in clinical trials, is highly sensitive. Also, AI systems can make mistakes which potentially might affect humans and have a severe sequence. These ethical challenges are not unique to our study but reflect broader concerns within the AI research community, especially as AI tools become more integrated into healthcare and clinical research. Efforts must be made to ensure that the corresponding systems are trained on diverse datasets, regularly evaluated, and compliant. Qualified medical personnel should always verify the results.

8. Conclusions

This research addresses the challenges of understanding and matching clinical trial inclusion criteria across languages, specifically English and Ukrainian, utilizing the GPT-4 model. The main scientific task was to assess whether advanced natural language processing (NLP) technologies could enhance the accuracy and efficiency of cross-lingual analysis in clinical research settings. The GPT-4 model has demonstrated a robust capability to understand and match clinical trial inclusion criteria between English and Ukrainian versions, showcasing high accuracy, precision, and recall.

The application of GPT-4 represents a significant advancement in bridging language barriers in clinical research, contributing to more efficient, inclusive, and equitable clinical trial processes. The study's quantitative and qualitative results provide a strong basis for the practical application of artificial intelligence in enhancing global health research methodologies in multilingual environments.

Scientific Novelty. The use of GPT-4 for matching English and Ukrainian clinical trial inclusion criteria matching demonstrated a high degree of scientific novelty, marking the first instance of such an application in the field. By breaking down and analyzing inclusion criteria, the study demonstrates how deep learning models can analyze and understand complex medical

terminology and nuances across languages. Compared to previously known methods, the GPT-4 model has shown superior performance, which allows for a more accurate and inclusive participant selection process, decreasing linguistic biases and barriers in global clinical research, and provides other advantages.

Practical Significance. The findings offer a practical framework for applying AI and NLP technologies in improving global clinical trial processes, particularly in patients' recruitment and documentation. The research outcomes can be used by Clinical Research Organizations (CRO), researchers, trial coordinators, and regulatory bodies to enhance the inclusivity, efficiency, and accuracy of clinical trials across different linguistic regions. It can also be used by regular people and patients who are struggling with severe disease and desperately looking for the most recent medicine which can potentially help with their treatment and give hope for a better and healthier life.

Recommendations for Practical Use include integrating NLP tools like GPT-4 into clinical trial management systems to automate and optimize the screening and matching of participants. In particular, considering the positive results of Ukrainian clinical text interpretation, Clinical Research Organizations (CRO) should seriously consider integrations with E-health (a medical information system for the Ministry of Healthcare of Ukraine) or local Electronic Health Records (EHR) systems like HELSI, DrEleks, and others.

Another area of use can be related to centralized data validation and minimization of human errors. Even during the experiments of this study, we have identified four significant differences in comparison of English and Ukrainian inclusion criteria, most likely caused by a human factor. Such mistakes can potentially cause incorrect patients' selection and affect their safety as well as the clinical study's results.

Further research should explore the application of GPT-4 and similar models across a broader range of languages and clinical contexts. Investigating the integration of NLP tools with other digital health systems like Clinical Trial Management Systems (CTMS) and Electronic Health Records (EHR) could yield additional improvements in clinical trial design and execution. Longitudinal studies to track the impact of NLP integration on trial outcomes and patient diversity could provide deeper insights into the long-term benefits and challenges of AI in clinical research.

9. Acknowledgements

The authors would like to thank the anonymous reviewers for their detailed and constructive feedback.

References

- [1] F. Cascini, F. Beccia, F.A. Causio, A. Melnyk, A. Zaino and W. Ricciardi. Scoping review of the current landscape of AI-based applications in clinical trials. *Frontiers Public Health*. 10:949377. (2022). doi:10.3389/fpubh.2022.949377
- [2] Bin Zhang, Lu Zhang, Q. Chen, Zhe Jin, S. Liu, S. Zhang. Harnessing artificial intelligence to improve clinical trial design. *Communication Medicine* 3, 191 (2023). doi:10.1038/s43856-023-00425-3
- [3] A.F. Hernandez, C.J. Lindsell. The Future of Clinical Trials: Artificial to Augmented to Applied Intelligence. *JAMA*. November 11. 330(21): (2023) 2061–2063. doi:10.1001/jama.2023.23822
- [4] E. H. Weissler, T. Naumann, T. Andersson et al. The role of machine learning in clinical research: transforming the future of evidence generation. *Trials* 22, 537 (2021). doi:10.1186/s13063-021-05489-x
- [5] K. Kreimeyer, M. Foster, A. Pandey, N. Arya, G. Halford, S. F. Jones, R. Forshee, M. Walderhaug, T. Botsis. Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review. *J. Biomedical Informatics*. V.73, (2017) 14-29. doi:10.1016/j.jbi.2017.07.012
- [6] Y. Gao, D. Dligach, L. Christensen, S. Tesch, R. Laffin, D. Xu, T. Miller, O. Uzuner, M.M Churpek, M. Afshar. A scoping review of publicly available language tasks in clinical natural language

- processing. J. of the American Medical Informatics Association. V.29. Issue 10. (2022) 1797-1806. doi:10.1093/jamia/ocac127
- [7] Ewoud Pons, Loes M. M. Braun, M. G. Myriam Hunink, Jan A. Kors. Natural Language Processing in Radiology: A Systematic Review. *Radiology*. May;279(2). (2016) 329-43. doi:10.1148/radiol.16142770
 - [8] T. Cai, A. A. Giannopoulos, S. Yu, T. Kelil, B. Ripley, K. K. Kumamaru, F. J. Rybicki, D. Mitsoura. Natural Language Processing Technologies in Radiology Research and Clinical Applications. *Radiographics*. Jan-Feb;36(1). (2016). doi:10.1148/rg.2016150080
 - [9] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan et al. Large language models in medicine. *Nature Medicine*. 29, (2023) 1930–1940. doi:10.1038/s41591-023-02448-8
 - [10] I. J. B. Young, S. Luz, N. Lone. A systematic review of natural language processing for classification tasks in the field of incident reporting and adverse event analysis. *International Journal of Medical Informatics*. V.132. (2019). doi:10.1016/j.ijmedinf.2019.103971
 - [11] A. Wong, J. M. Plasek, S. P. Montecalvo, L. Zhou. Natural Language Processing and Its Implications for the Future of Medication Safety: A Narrative Review of Recent Advances and Challenges. *Pharmacotherapy*. Aug;38(8). (2018) 822-841. doi:10.1002/phar.2151
 - [12] L. P. Nijhawan, M. D. Janodia, B. S. Muddukrishna, K. M. Bhat, K. L. Bairy, N. Udupa, P. B. Musmade. Informed consent: Issues and challenges. *J. of Advanced Pharmaceutical Technology & Research*. Jul;4(3). (2013). doi:10.4103/2231-4040.116779
 - [13] J. Kim, Y. Quintana. Review of the Performance Metrics for Natural Language Systems for Clinical Trials Matching. *Studies in Health Technology and Informatics*. Jun 6;290. (2022) 641-644. doi:10.3233/SHTI220156
 - [14] C. Fang, Y. Wu, W. Fu, J. Ling, Y. Wang, X. Liu, Y. Jiang, Y. Wu, Y. Chen, J. Zhou, Z. Zhu, Z. Yan, P. Yu, X. Liu. How does ChatGPT-4 preform on non-English national medical licensing examination? An evaluation in Chinese language. *PLOS Digit Health*. Dec 1;2(12). (2023). doi:10.1371/journal.pdig.0000397
 - [15] S. Voloshyn, V. Vysotska, O. Markiv, I. Dyyak, I. Budz and V. Schuchmann, "Sentiment Analysis Technology of English Newspapers Quotes Based on Neural Network as Public Opinion Influences Identification Tool," 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 2022, pp. 83-88, doi: 10.1109/CSIT56902.2022.10000627
 - [16] D. Ueda, S. L. Walston, T. Matsumoto et al. Evaluating GPT-4-based ChatGPT's clinical potential on the NEJM quiz. *BMC Digit Health* 2, 4 (2024). doi:10.1186/s44247-023-00058-5
 - [17] N. Oh, G.-S. Choi, W. Y. Lee. ChatGPT goes to the operating room: evaluating GPT-4 performance and its potential in surgical education and training in the era of large language models. *Annals of Surgical Treatment and Research*. May;104(5). (2023) 269-273. doi:10.4174/astr.2023.104.5.269
 - [18] E. Waisberg, J. Ong, M. Masalkhi, S. A. Kamran, N. Zaman, P. Sarker, A. G. Lee, A. Tavakkoli. GPT-4: a new era of artificial intelligence in medicine. *Irish Journal of Medical Science*. V.192(6). (2023) 3197-3200. doi:10.1007/s11845-023-03377-8
 - [19] G. A. Guerra, H. Hofmann, S. Sobhani, G. Hofmann, D. Gomez, D. Soroudi, B.S. Hopkins, J. Dallas, D. J. Pangal, S. Cheok, V. N. Nguyen, W. J. Mack, G. Zada. GPT-4 Artificial Intelligence Model Outperforms ChatGPT, Medical Students, and Neurosurgery Residents on Neurosurgery Written Board-Like Questions. *World Neurosurgery*. V.179. (2023) e160-e165. doi:10.1016/j.wneu.2023.08.042
 - [20] Sang-Wook Lee, Woo-Jong Choi. Utilizing ChatGPT in clinical research related to anesthesiology: a comprehensive review of opportunities and limitations. *Anesthesia and Pain Medicine*. 18(3). (2023) 244-251. doi:10.17085/apm.23056
 - [21] M. Rosoł, J. S. Gąsior, J. Łaba, K. Korzeniewski & M. Mlyńczak. Evaluation of the performance of GPT-3.5 and GPT-4 on the Polish Medical Final Examination. *Scientific Reports*. 13, 20512 (2023). doi:10.1038/s41598-023-46995-z
 - [22] H. Nori, Y. T. Lee, S. Zhang, D. Carignan, R. Edgar, N. Fusi, N. King, J. Larson, Y. Li, W. Liu, R. Luo, S. M. McKinney, R. O. Ness, H. Poon, T. Qin, N. Usuyama, C. White & E. Horvitz. Can

Generalist Foundation Models Outcompete Special-Purpose Tuning? Case Study in Medicine. ArXiv. Computer Science. Computation and Language. (2023)

- [23] A. Wang, X. Xiu, S. Liu, Q. Qian, S. Wu. Characteristics of Artificial Intelligence Clinical Trials in the Field of Healthcare: A Cross-Sectional Study on ClinicalTrials.gov. International Journal of Environmental Research and Public Health. 19(20). (2022). doi:10.3390/ijerph192013691

Development of an Intelligent Chatbot for a Hospital Website

Tetiana A. Vakaliuk^{1,2,3}, Dariia G. Skripchenko¹, Mariia O. Medvedieva⁴ and Mykhailo G. Medvediev⁵

¹ Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine

² Institute for Digitalisation of Education of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine

³ Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine

⁴ Pavlo Tychyna Uman State Pedagogical University, Sadova str., 2, Uman, Cherkasy region, 20300, Ukraine

⁵ ADA University, School of Information Technologies and Engineering, 61 Ahmadbey Aghaoghlu Str., Baku, AZ1008, Azerbaijan

Abstract

The study examined the ideas, procedures, and algorithms for creating chatbots and clinic websites. The study also looked at the prospects of chatbots and whether or not they are necessary for digital resources in general. A chatbot was created without using the ChatterBot framework, and its structure and data source were fully explained. The methodologies, algorithms, and libraries used to design and train the chatbot were also described. In addition, we carried out a visual analysis, discussed the findings, and made judgments regarding the chatbot's functionality.

Keywords

chatbot, AI, DNN, deep learning, neural networks

1. Introduction

Although medicine has long been viewed as a profession based solely on face-to-face communication, it has become increasingly clear that the use of IT, in general, and electronic communications, in particular, provides not only cost-effective healthcare but also a way to ensure remote communication between medical staff and patients or their carers.

The use of the Internet in medical activities has become a reality in recent years, and telemedicine, or virtual hospitals, has moved from the stage of science fiction to reality [1]. The development of the information society, the increase in the number of people with access to computers, even the elderly (including the development of introductory computer courses for them), the diversification of information, and the need for rapid interaction have led to the adaptation of medical units to the means of information transmission to make them more attractive and more accessible to operate and use. Developing systems with guaranteed quality information should be a constant concern of public health professionals, who should work with IT specialists to create their web pages.

Many Internet sites are trying to establish themselves as sources of medical information and as a means of connecting patients with doctors. The number of visitors to these sites and the considerable number of sites indicate public interest in these services. Therefore, they need to use the available technologies to communicate effectively with patients, continue to improve them and create new ones.

The development and improvement of clinic and hospital websites should be the focus of attention for healthcare and informatics professionals. Clinic and hospital websites should ensure compliance with ethical rules, be functional and professional from an informatics point of view, and provide adequate information to potential clients of medical units. Websites are a means of communication and a convenient IT tool for better management of these structures.

✉ tetianavakaliuk@gmail.com (T. A. Vakaliuk); dariaskripchenko@gmail.com (D. G. Skripchenko); medvedeva-masha25@ukr.net (M. O. Medvedieva); miserablewisdom@ukr.net (M. G. Medvediev)

🆔 0000-0001-6825-4697 (T. A. Vakaliuk); 0009-0000-5540-8677 (D. G. Skripchenko); 0000-0001-9330-5185 (M. O. Medvedieva); 0000-0002-3884-1118 (M. G. Medvediev)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The study aims to selection the methods and tools for developing a chatbot for a clinic/hospital information resource.

2. Theoretical background

2.1. Analysing the need to use chatbots

Researchers have looked into how people engage with chatbots in both lab and real-world environments in order to analyze user behavior. In contrast to user interactions during chats with other users, [1] observed that user interactions with chatbots involved longer conversations with more and shorter messages, as well as lower saturation than user interactions during chats with other users. In [4] studied the real-world use of a voice conversational agent and found that the real-world context posed challenges to the conversation and led to frequent failures in the chatbot interaction. Such studies provide insights into how users behave when interacting with or responding to chatbots. As a result, they offer helpful advice for designing chatbot interactions, for instance, to reduce usability problems and enhance conversational recovery.

In [5] investigated users' motivations for using chatbots. As a result, they offer helpful advice for designing chatbot interactions, for instance, to reduce usability problems and enhance conversational recovery. Researchers [6] also looked into user preferences for various chatbot personas and discovered that people like personas that show engagement and productivity. [7] studied users' experiences with voice chatbots and found that users struggle to understand chatbot features and interact effectively with them; they suggested that a game could be a helpful entry point for learning chatbot features and interacting effectively.

Relevant insights into chatbot use and user experience can also be gleaned from the literature on dialogue system evaluation, for example, regarding frameworks and metrics for dialogue evaluation [8], which target constructs such as efficiency, effectiveness, and user satisfaction with tasks: task-oriented systems, relevance, and human-likeness for systems focused on small talk [9]. Furthermore, competitions for dialogue systems, like the Alexa Prize [10], in which groups compete to create voice chatbots capable of social interaction, greatly advance our knowledge of user preferences and views of chatbots. But further research is needed to determine how pragmatic and hedonic characteristics, taken separately or together, affect the chatbot user experience as a whole.

Therefore, lessons have been learned that apply, in particular, to task-oriented chatbots [11].

1. In task-oriented chatbots, usefulness is paramount. For task-oriented chatbot applications, solving user problems and helping users achieve their goals efficiently and effectively is critical to ensuring a quality chatbot experience. Providing valuable assistance and support is essential for sustained engagement, and for almost all chatbot applications, it is crucial to interpret user intent and provide appropriate responses correctly. Although chatbots are still a new form of interactive system, it is essential for service providers to carefully ensure that their chatbots serve their intended purposes and that these purposes are perceived as valuable by their users.
2. Hedonic attributes can improve the user experience in task-oriented chatbots. For many task-oriented chatbot applications, the user experience can be enhanced by combining pragmatic and hedonic chatbot attributes. While a very useful chatbot can provide a good user experience, this experience can be further enhanced by carefully incorporating content and chatbot characteristics perceived as pleasant, engaging, or playful.
3. User reports are valuable. Understanding how users work with chatbots is difficult. Nevertheless, understanding such experiences is crucial for chatbot service providers to increase the acceptance of chatbots among the public. With the research approach presented here, we have demonstrated the feasibility and benefits of collecting user reports using a questionnaire based on critical incident techniques. We hope this will serve as an example of how service providers can approach the collection of high-quality user reports that provide much-needed rich information about the chatbot experience.
4. Different users have different needs. The natural language interaction in chatbots makes them very suitable for personalisation. An example is our finding that pragmatic and hedonic chatbot attributes have different meanings for chatbot users of varying age

groups. However, the ability to personalise chatbots has not been sufficiently implemented. On the contrary, current chatbots typically display the same personality and provide the exact content regardless of user characteristics. Chatbot service providers can benefit from investing in understanding the needs of their different user groups and configuring chatbots that can adapt accordingly.

2.2. Concepts, methods and algorithms of chatbot development

A chatbot is a computer program developed based on neural networks and machine learning technologies that allow communication in audio or text format. A chatbot performs specific tasks (e.g., obtaining reference information, performing calculations) or for entertainment.

Typically, a chatbot prompts the user to choose an option from a list of options. That is, obtaining information or performing other tasks is done by selecting the suggested answers or categories, and the user does not enter text into the chat window. More sophisticated bots can enter text and receive information per the user's request.

Virtual chatbots allow you to place orders or communicate online.

Chatbots can be divided into the following areas of application:

- p2p - personal communications (for personal communication);
- b2c - consumer (support for company customers on the corporate website and in mobile applications).

Chatbots mainly use artificial intelligence to communicate with users, providing relevant content and up-to-date offers. They operate based on a set of instructions or use machine learning.

Some of the main concepts associated with chatbot technology are as follows [12]:

- Pattern matching, which is based on typical stimulus-response blocks. A sentence (stimulus) is entered, and a response (reaction) is generated according to the data entered by the user.
- AIML (Artificial Intelligence Markup Language) is based on Pattern recognition or Pattern matching concepts.
- Latent semantic analysis can be used with AIML to develop chatbots. It is used to identify similarities between words as a vector representation.
- Chatscript, the successor to AIML, is an expert system consisting of an open-source scripting language and an engine that runs it. Chatscript includes rules associated with topics, finding the best element that matches the user's query string, and executing the rule in that topic.
- RiveScript is a plain-text string scripting language for developing chatbots and other conversational objects. RiveScript is open source with interfaces for Go, Java, JavaScript, Perl, and Python.
- Natural language processing (NLP) is a field of artificial intelligence that studies the computer-assisted manipulation of texts or natural language speech.
- Natural Language Understanding (NLU) is the foundation of any NLP task. It is a technique for implementing natural user interfaces such as chatbots. NLU seeks to extract context and meaning from user input, which may be unstructured and respond appropriately to the user's intentions [1].

A bot's intelligence level depends solely on how it is programmed. A chatbot based on machine learning performs better because it understands commands and language. Therefore, users do not need to enter exact words to get relevant answers. In addition, the bot learns from interactions with customers and can quickly solve similar situations when they arise. Each such dialogue improves the bot's intelligence [12].

Chatbots can use neural networks as a language model [13]. For example, generative pre-trained transformers (GPTs), which use a transformer architecture, have become standard for building sophisticated chatbots. The "pre-training" in its name refers to the process of initial training on a large text corpus that provides a solid foundation for the model to perform well on subsequent tasks with a limited amount of task-related data. An example of a GPT chatbot is ChatGPT. Despite criticism about its accuracy, ChatGPT has attracted attention for its detailed

answers and historical knowledge. Another example is BioGPT, developed by Microsoft, which focuses on answering biomedical questions.

Here are the most common limitations [14]:

- Since the I/O database is fixed and limited, chatbots can crash while processing an unsaved query.
- Chatbot performance heavily depends on speech processing and is limited by disturbances such as accents and spelling or grammatical errors.
- Chatbots cannot answer several questions simultaneously, limiting their communication capabilities.
- Chatbots require a large amount of conversational data to train them. Based on deep learning algorithms to create new answers word by word based on user input, generative models are usually trained on a large dataset of natural language phrases.
- It is difficult for chatbots to manage non-linear conversations that need to flow naturally and be relevant to the topic of the conversation with the user.
- As is usually the case with technological changes to existing services, some consumers, most often older ones, are uncomfortable with chatbots due to their limited understanding, which makes it evident that machines are processing their queries.

3. Proposed methodology.

Before you start working on a chatbot for a ready-made web resource, you should first understand the basic algorithms and concepts of chatbot technology development using Python.

It was decided that the chatbot would be based on a DNN (Deep Neural Network) to identify sentence patterns provided by the user as input and select a random answer related to this query. The chatbot will be implemented using Python libraries, such as the NLTK library in Python, which has functions that help identify the most relevant words in a sentence or paragraph, link them to their root meaning, and shorten them. This process is known as Stemming. The words are then converted to their corresponding numerical values, as neural networks only understand numbers. The process of converting text to numeric values is known as One-Hot Encoding. Once the data preprocessing is complete, we will create the neural networks using TFLearn and feed them with training data. After successful training, the model should be able to predict tags related to the user's query.

To train our chatbot, we need the "intents.json" file. Each intent contains a tag, templates, answers, and context. The templates are the data the user will likely enter, and the answers are the chatbot's results. This file contains general greeting phrases, questions about which doctors work in the clinic, and questions about the appointment, such as the appointment cost, which days you can make an appointment, and how you can make an appointment.

The challenge now is to get the model to learn the relationship between patterns and tags so that when a user enters a particular phrase, it can identify the corresponding tag and produce one of the answers resulting from the query.

The first step is to download the Punkt model provided by the NLTK library. In NLTK, PUNKT is an unsupervised learning model, meaning it can be trained on unlabeled data (data that has not been annotated with information about characteristics, properties, or categories is called unlabeled data).

Using unsupervised techniques, it generates a list of sentences from the text by developing a model for sentence starters, prepositional phrases, and contractions. It must be trained on a significant amount of open text in the target language without prior use.

Next, look at the libraries and modules that will be used to implement a chatbot. The NLTK supports research and teaching in NLP and related fields, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. NLTK is successfully used as a training tool and a platform for prototyping and building research systems. When working on any project in natural language processing, NLTK is the most critical module used.

TensorFlow is an open-source machine learning software library for a range of tasks developed by Google to meet its need for systems that can build and train neural networks to

detect and decipher patterns and correlations, similar to the learning and understanding used by humans. It is currently used for both research and product development at Google.

TensorFlow provides a library of ready-made algorithms for numerical computing implemented through data flow graphs. The nodes in such graphs execute mathematical operations or input/output points, while the graph's edges represent multidimensional data arrays (tensors) that flow between the nodes. Nodes can be assigned to computing devices and run asynchronously, processing all the parallel tensors that fit them together. This allows for the simultaneous operation of nodes in a neural network by analogy with the simultaneous activation of neurons in the brain.

However, TensorFlow is a low-level library. Using TensorFlow directly is challenging, as the API is highly verbose and prone to subtle actions. It is difficult to detect errors in TensorFlow. To overcome these difficulties, a shell called TFLearn was developed.

TFLearn is a modular and transparent deep-learning library built on top of TensorFlow. It was developed to provide a higher-level API for the TensorFlow library to make experiments easier and faster while remaining fully transparent and compatible.

After pre-loading the "intents.json" file and receiving the data, it is necessary to start pre-processing it, as shown in Fig. 1.

```
words = [stemmer.stem(w.lower()) for w in words if w not in "?"]
words = sorted(list(set(words)))
labels = sorted(labels)
```

Figure 1: Preliminary data processing

This recognises words and removes duplicate words from the word list. For this task use the Lancaster Stemmer algorithm to reduce words to their base (Figure 2). First, take a quick look at what stemming is. Stemming is the process of shortening a word to its base. A word stem is a word's base or root form and does not have to be an existing word.

Lancaster is one of the most aggressive stemmers because it tends to overstep many words. A Lancaster stemmer consists of a set of rules, where each rule specifies either the deletion or replacement of an ending. In addition, some rules are restricted to intact words, and some rules are applied iteratively as the word passes through them.

Because of the strict rules, there are two additional conditions to prevent different short-root words from occurring:

- If the word starts with a vowel, at least two letters must remain after the root (owing -> ow, but not ear -> e).
- If the word begins with a consonant, at least three letters must remain after the root letter, and at least one of them must be a vowel or "y" (speaking -> say, but not string -> str).

The Lancaster header stemmer has over 100 rules.

```
training = []
output = []
out_empty = [0 for _ in range(len(labels))]

for x, doc in enumerate(x_docs):
    bag = []
    wrds = [stemmer.stem(w) for w in doc]
    for w in words:
        if w in wrds:
            bag.append(1)
        else:
            bag.append(0)

    output_row = out_empty[:]
    output_row[labels.index(y_docs[x])] = 1

    training.append(bag)
    output.append(output_row)

training = np.array(training)
output = np.array(output)
```

Figure 2: One-Hot coding and training data preparation.

The resulting training and output data are encoded in One-Hot mode. The words are converted into ones and zeros, which are then added to the training list and the output list and then converted into NumPy arrays. We need to create our model using neural networks (Figure 3).

```
net = tflearn.input_data(shape=[None, len(training[0])])
net = tflearn.fully_connected(net, 10)
net = tflearn.fully_connected(net, 10)
net = tflearn.fully_connected(net, 10)
net = tflearn.fully_connected(net, len(output[0]), activation='softmax')
net = tflearn.regression(net)

model = tflearn.DNN(net)
model.fit(training, output, n_epoch=500, batch_size=8, show_metric=True)
```

Figure 3: Creating a model using neural networks.

The first layer is the input layer, with a parameter of equally sized input data. The following three middle layers are the hidden layers responsible for all input processing. The output layer provides the probabilities of different words in the training data.

The training data is fitted into the model, and epochs are set to 500, during which the training will continue until it reaches 500 iterations. An epoch in a neural network trains the neural network with all the training data in one cycle. Using more than one epoch is necessary to get good performance from the non-training data (in practice, this can be roughly estimated using a set of delays); usually (but not always), this takes more than one pass of training data.

We have trained the model with a list of words, or a so-called bag of words, so we also need to do the same to make predictions. We can now create a function that will give us a bag of words to predict for our model (Figure 4).

```
def bag_of_words(s, words):
    bag = [0 for _ in range(len(words))]
    s_words = nltk.word_tokenize(s)
    s_words = [stemmer.stem(word.lower()) for word in s_words]

    for s_word in s_words:
        for i, w in enumerate(words):
            if w == s_word:
                bag[i] = 1
    return np.array(bag)
```

Figure 4: Function to provide a set of words for model prediction.

This function helps to create a word pack for our model.

Now we need to create a chat function that brings all of this together (Figure 5).

```
def chat():
    while True:
        inp = input("\n\nYou: ")
        if inp.lower() == 'quit':
            break
        results = model.predict([bag_of_words(inp, words)])

        results_index = np.argmax(results)
        tag = labels[results_index]

        for tg in data['intents']:
            if tg['tag'] == tag:
                responses = tg['responses']
                print("Bot:\t" + random.choice(responses))
```

Figure 5: Chat function.

First, the model predicts outcomes using the word packet and user data and then returns a list of probabilities. Among the probabilities, the highest number is likely to be the result the user expects. So, we choose the index of the highest probability and find the tag and answers of that particular index. Then, we can select some random answers from the list of answers. Fig. 6 shows how a chatbot can identify a user input pattern and respond accordingly.

```

Training Step: 2999 | total loss: 0.00517 | time: 0.040s
| Adam | epoch: 500 | loss: 0.00517 - acc: 1.0000 -- iter: 40/41
Training Step: 3000 | total loss: 0.00550 | time: 0.046s
| Adam | epoch: 500 | loss: 0.00550 - acc: 1.0000 -- iter: 41/41
--

You: Hi
Bot: Hi there, how can I help?

You: List of specialties
Bot: There are specialists in gastroenterology, cardiology, orthopedy, otolaryngology and more. Please refer to our specialists page to learn more

You: How much it costs?
Bot: 100 for a consultation, 120 for a check-up and 200 for an analysis

You: Thanks
Bot: My pleasure

You: quit

```

Figure 6: An example of how a chatbot works.

We can see that the chatbot provides correct answers to user queries after the training. Typing in common phrases results in the chatbot answering the question, and typing in quit results in the chatbot terminating the conversation.

3.1. Analysis of chatbot performance and its results in charts and graphs

TensorBoard was used to visually analyse the chatbot. TensorBoard is a tool for providing measurements and visualisations required during the machine learning workflow. It allows you to track experiment metrics such as loss and accuracy, visualise the model graph, plot embeddings in a lower dimensional space, and much more. First, you must update the code, as shown in Figure 7.

```

log_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")

model = tflearn.DNN(net, tensorboard_verbose=3, tensorboard_dir=log_dir)

```

Figure 7: Listing of changes in the code.

TFLearn supports a detailed level for automatic management of the results:

0: Losses and performance (best speed).

1: Loss, metrics and gradients.

2: Loss, metrics, gradients and weights.

3: Loss, metrics, gradients, weights, activations, and sparsity (best visualisation).

For the `tensorboard_verbose` property, the value three was chosen for the best visualisation.

Accordingly, the logs will be stored in subdirectories with a time stamp so that you can easily select different training runs. To do this, create a `log_dir` variable that stores the subdirectory's name.

You can then launch Tensorboard to visualise the performance.

The TensorBoard Graphs dashboard is a tool for exploring your TensorFlow model. You can quickly view a conceptual graph of the model's structure to ensure it matches your intended design. You can also view the operational level graph to understand how TensorFlow understands the application. Examining the operational level graph can give you insight into how to modify your model. This is the tool that will be used for visual analysis.

As a result, we get the following graphs of accuracy and training losses, shown in Figs. 8-9.

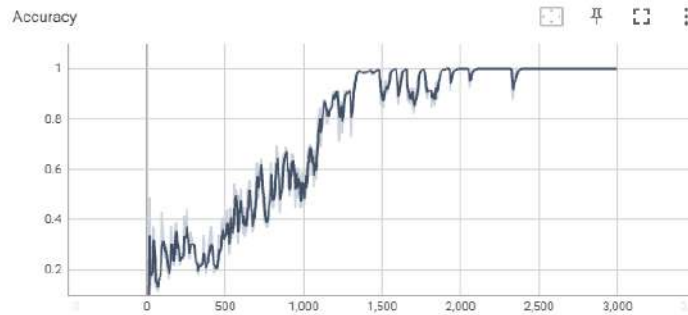


Figure 8: Accuracy chart.

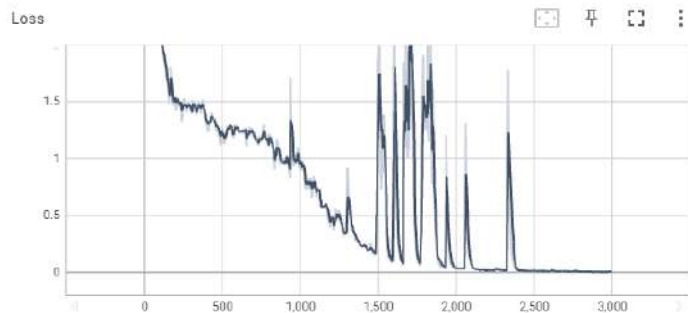


Figure 9: Schedule of losses during training.

In most cases, accuracy increases as loss decreases, but this is not always true. Accuracy and loss have different definitions and measures. They often appear to be inversely proportional, but the two have no mathematical relationship.

Low precision but high loss means the model makes significant errors in most data. However, if both loss and accuracy are lacking, the model makes minor errors in most data. However, if both are high, it results in significant errors in some data. Finally, if precision is high and loss is low, the model makes minor errors in only some data, which would be ideal.

Based on the data obtained, we can conclude that the chatbot learns quickly from the data provided and makes a small number of errors, indicating the low complexity of the data provided.

4. Results

An untrained ChatterBot instance starts without knowing how to communicate. Each time a user enters a statement, the library stores the text they entered and the text to which the statement was a response. As ChatterBot receives more data, the number of responses it can respond to and the accuracy of each response about the input statement increase.

The program selects the most appropriate answer by looking for the closest matching known statement that reaches the input and then choosing an answer from the list of available responses to that statement (Figure 10).

Taking into account the process of the frame, we first initialise the ChatBot instance as shown in Fig. 11.

ChatterBot preprocessors are simple functions that modify the input statement that the chatbot receives before processing the statement by the logic adapter.

Logic adapters define the logic of how ChatterBot chooses a response to a given input statement.

A typical Boolean adapter will use two main steps to return a response to an input statement. The first step involves searching the database for a known statement that matches or closely matches the input statement. Once a match is selected, the second step consists of establishing a general answer to the chosen match. Often, several existing statements will answer an available game, and the BestMatch logic adapter will determine the answer based on the best-known match with the given statement.

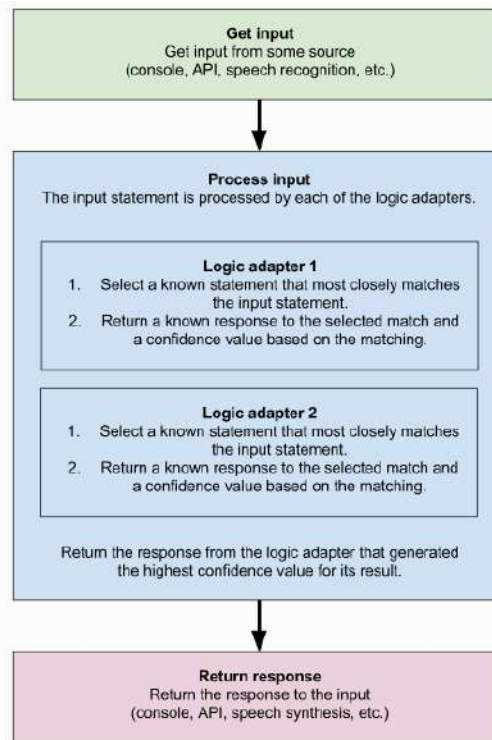


Figure 10: Diagram of the ChatterBot workflow.

```

bot = ChatBot('MedBot', read_only=True,
              preprocessors=[chatterbot.preprocessors.convert_to_ascii',
                             chatterbot.preprocessors.unescape_html',
                             chatterbot.preprocessors.clean_whitespace'],
              logic_adapters=[
                  {
                      'import_path': 'chatterbot.logic.BestMatch',
                      'default_response': 'Sorry, I am unable to process your request.',
                      'maximum_similarity_threshold': 0.90
                  }
              ],)
  
```

Figure 11: Initialising a ChatBot instance.

The best match adapter uses a function to compare the input statement with known statements. When it finds the closest match to the input operator, it uses another function to select one of the available answers to that operator.

We need to train the bot with training data to provide the bot with some knowledge. The training data is filled in a list that will represent the conversation. ChatterBot uses natural language processing to learn the data provided to the bot. Fig. 12 shows a listing of the training data file with site navigation, and Fig. 13 lists the training data file for delivering essential medical advice for minor medical problems.

```

1 Hi
2 Hello, how may I help?
3 Hello
4 Hello, how may I help?
5 Good morning
6 Good morning, how may I help?
7 Good evening
8 Good evening, how may I help?
9 I would like to book an appointment
10 Sure, please click on the Book Appointment button on the Home page.
11 I want to book an appointment
12 Sure, please click on the Book Appointment button on the Home page.
13 I want to see a doctor
14 Sure, please click on the Book Appointment button on the Home page.
15 How to book an appointment?
16 Please click on the Book Appointment button on the Home page.
17 Do I have to book an appointment online?
18 You can call the hospital. Our phone number is (093)-123-34-43.
  
```

Figure 12: Listing a file with training data with site navigation.

```

1 What to do if cuts?
2 Wash the cut properly to prevent infection and stop the bleeding by applying pressure for 1-2minutes until ble
3 How to cure cuts?
4 Wash the cut properly to prevent infection and stop the bleeding by applying pressure for 1-2minutes until ble
5 Which medicine to apply for cuts?
6 Wash the cut properly to prevent infection and stop the bleeding by applying pressure for 1-2minutes until ble
7 What to apply on cuts?
8 Wash the cut properly to prevent infection and stop the bleeding by applying pressure for 1-2minutes until ble
9 Cuts
10 Wash the cut properly to prevent infection and stop the bleeding by applying pressure for 1-2minutes until ble
11 How do you treat abrasions?
12 Begin with washed hands.Gently clean the area with cool to lukewarm water and mild soap. Remove dirt or other
13 Do abrasions cause scars?
14 Begin with washed hands.Gently clean the area with cool to lukewarm water and mild soap. Remove dirt or other
15 What to do if abrasions?
16 Begin with washed hands.Gently clean the area with cool to lukewarm water and mild soap. Remove dirt or other
17 Which medicine to apply for abrasions?
18 Begin with washed hands.Gently clean the area with cool to lukewarm water and mild soap. Remove dirt or other
19 How to cure abrasions?
20 Begin with washed hands.Gently clean the area with cool to lukewarm water and mild soap. Remove dirt or other
21 Abrasions
22 Begin with washed hands.Gently clean the area with cool to lukewarm water and mild soap. Remove dirt or other

```

Figure 13: Listing of a file with training data on giving basic health advice.

Further, these data should be processed as shown in Fig. 14.

```

data1 = open('training_data.txt').read()
conversations = data1.strip().split('\n')

data2 = open('training_help.txt').read()
helping = data2.strip().split('\n')

```

Figure 14: Data processing.

After that, we use the processed data to train the bot, as shown in Fig. 15.

```

trainer.train(conversations)
trainer.train(helping)

```

Figure 15: Training a chatbot with processed data.

Now the chatbot is ready to use. Fig. 16 shows an example of how a chatbot works.

```

(venv7) E:\vazhno\master\praktika23\client\api>py medchatbot.py
List Trainer: [#####] 100%
Welcome to the ChatBot Service for Med! How can I help you? (Enter 'Bye' to exit)
You:Hello
ChatBot: Hello, how may I help?
You:I want to book an appointment
ChatBot: Sure, please click on the Book Appointment button on the Home page.
You:Can i come without booking
ChatBot: You can visit the hospital without an appointment. If there will be an empty slot a doctor will see you.
You:What services do you offer
ChatBot: We offer check-ups, consultations as well as medical tests.
You:How much is a check-up
ChatBot: The check-up is 100.
You:Which doctors are there
ChatBot: We have specialists in the gastroenterology, cardiology, orthopedy, otolaryngology and endocrinology fields.
You:thanks
ChatBot: Of course!
You:bye
ChatBot: It was great talking to you!

```

Figure 16: An example of how a chatbot works.

The client side of the site was developed using React. The main page of the site looks like this, as shown in Fig. 17.



Figure 17: The main page of the site.

The main menu contains links to other sections of the website, which the user can access by clicking on. For example, clicking on the About link opens the section with information about the clinic.

When you click the Book Appointment button on the main page, a form for leaving a request for an appointment with a doctor opens.

The chatbot connects to a web resource using the Flask library on the backend, which was developed in Python, and the fetch method on the client side was developed with React.

The chatbot is available to users by clicking the Chatbot button in the main menu—an example of how the chatbot works is shown in Fig. 18.

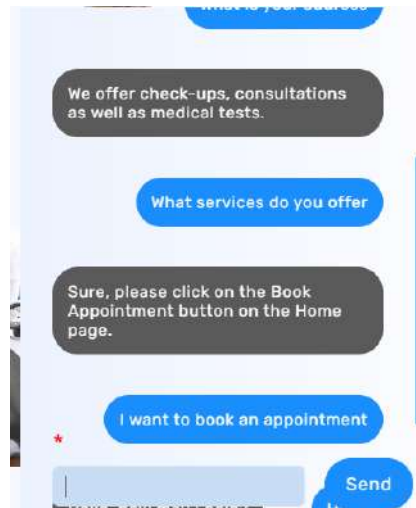


Figure 18: An example of how a chatbot works.

5. Discussion

The developed chatbot communicates exclusively in English. Therefore, the disadvantages include localisation in one language.

This is not critical, as English is an international language, firstly, and secondly, there is room for improvement and the addition of other functions. To further add other languages of communication, it will be necessary to train the proposed models in other languages.

The question of how much a chatbot can provide a correct and adequate answer to a user is also debatable. For this purpose, it is worth conducting an experiment on the quality of communication between a healthcare professional as a user and a chatbot. As a result, it would be possible to identify the existing shortcomings that need to be eliminated.

6. Conclusions

The study examined the concepts, methods, and algorithms for developing websites for clinics and chatbots. The study also examined the necessity of using chatbots for general web resources and their future prospects.

A chatbot was developed without using the ChatterBot framework. A full description of the data source used to train the chatbot was provided, and its structure and the implemented methods, algorithms, and libraries used to develop and train the chatbot were presented and described. We also conducted a visual analysis, reported its results, and drew conclusions about the chatbot's performance. After that, a chatbot was developed using the ChatterBot framework, which will be used for the finished web resource. The implemented methods and the process of developing and training a chatbot were described, and an example of its operation was presented.

In addition, the article demonstrates how the developed chatbot was integrated with the developed web resource and demonstrates the operation of the chatbot. As a result, the article studies the concepts and technologies of chatbot development, develops its chatbot connected to the developed web resource, and explores its necessity and relevance.

Since this chatbot was developed specifically for a hospital website, it is distinguished by training models for medical terminology. According to the analysis of existing chatbots, this is rare in this sector.

Further research includes experimental testing of the proposed chatbot in medical institutions and its expert evaluation. The possibility of adding several languages for communication is also among the prospects for further research.

References

- [1] Nikitchuk, T.M., Andreiev, O.V., Korenivska, O.L. and Medvediev, M.G., 2023. A model of an automated biotechnical system for analysing pulseograms as a kind of edge device. *Journal of Edge Computing* [Online], 2(1), pp. 64–83. Available from: <https://doi.org/10.55056/jec.627>
- [2] Korenivska, O.L., Benedytskyi, V.B., Andreiev, O.V. and Medvediev, M.G., 2023. A system for monitoring the microclimate parameters of premises based on the Internet of Things and edge devices. *Journal of Edge Computing* [Online], 2(2), pp. 125–147. Available from: <https://doi.org/10.55056/jec.614>
- [3] Escobar-Grisales, D., Vásquez-Correa, J.C. & Orozco-Arroyave, J.R. Evaluation of effectiveness in conversations between humans and chatbots using parallel convolutional neural networks with multiple temporal resolutions. *Multimed Tools Appl* 83 (2024) 5473–5492. doi: 10.1007/s11042-023-14896-y
- [4] Kendall, L., Chaudhuri, B. & Bhalla, A. Understanding Technology as Situated Practice: Everyday use of Voice User Interfaces Among Diverse Groups of Users in Urban India. *Inf Syst Front* 22 (2020) 585–605. doi: 10.1007/s10796-020-10015-6
- [5] Jindi Fu, Samar Mouakket, Yuan Sun. The role of chatbots' human-like characteristics in online shopping, *Electronic Commerce Research and Applications*, Volume 61 (2023) 101304. doi: 10.1016/j.elerap.2023.101304.
- [6] Brinda Mehra. Chatbot personality preferences in Global South urban English speakers, *Social Sciences & Humanities Open*, 3:1 (2021) 100131. doi: 10.1016/j.ssaho.2021.100131.
- [7] Rzepka, C., Berger, B. & Hess, T. Voice Assistant vs. Chatbot – Examining the Fit Between Conversational Agents' Interaction Modalities and Information Search Tasks. *Inf Syst Front* 24 (2022) 839–856. doi: 10.1007/s10796-021-10226-5
- [8] Haoming Jiang, Bo Dai, Mengjiao Yang, Tuo Zhao, and Wei Wei. Towards Automatic Evaluation of Dialog Systems: A Model-Free Off-Policy Evaluation Approach. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 7419–7451.
- [9] J. Deriu, A. Rodrigo, A. Otegi, et al., Survey on evaluation methods for dialogue systems, *Artificial Intelligence Review* 54 (2021) 755–810. doi: 10.1007/s10462-020-09866-x.
- [10] Sarah E. Finch, Jinho D. Choi. Towards Unified Dialogue System Evaluation: A Comprehensive Analysis of Current Evaluation Protocols. In: *Proceedings of the SIGdial 2020 Conference*. 2020, pp. 236–245. <https://www.sigdial.org/files/workshops/conference21/pdf/2020.sigdial-1.29.pdf>
- [11] A. Følstad, P. B. Brandtzaeg, Users' experiences with chatbots: findings from a questionnaire study, *Qual User Exp* 5, 3 (2020). Available online: <https://link.springer.com/article/10.1007/s41233-020-00033-2#Sec26>
- [12] Adamopoulou E, Moussiades L. An Overview of Chatbot Technology. *Artificial Intelligence Applications and Innovations*. 6;58 (2020) 373–383. doi: 10.1007/978-3-030-49186-4_31.
- [13] Kumar, Ramakrishna & Mahmoud, Maha. A Review on Chatbot Design and Implementation Techniques. *International Research Journal of Engineering Science Technology and Innovation* 07 (2020) 2791.
- [14] Sarker, Iqbal. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science* 2 (2021). doi: 10.1007/s42979-021-00815-1.

Use of Neural Network-based Models to Predict the Severity of Bronchial Asthma in Children

Oleh Pihnastyi¹, Olha Kozhyna² and Iuliia Karpushenko²

¹ National Technical University "KhPI", 2 Kyrpychova, Kharkiv, 61002, Ukraine

² Kharkiv National Medical University, 4 Nauky Avenue, Kharkiv, 61022, Ukraine

Abstract

Problem of early diagnosis of bronchial asthma in children is due to the heterogeneity of the disease. The time of onset of the first symptoms of the disease, their severity, and the possibility of controlling exacerbations are determined by the interrelation of a factors numerous. The aim of the study is to develop a method for predicting the severity of bronchial asthma based on the factors numerous for prediction. A clinical and paraclinical examination of 70 children with a diagnosis of bronchial asthma at the age from 6 to 18 years was carried out. 142 factors were analysed and the degree of interrelationship between them was determined. A model for predicting the degree of severity of bronchial asthma in children, the foundation of which is a neural network, is presented. A multilayer perceptron with an architecture containing one and two hidden layers was used to build the model. A comparative analysis of prediction results for models using neural network architecture with different number of nodes in hidden layers was performed. MSE (Mean Squared Error) values were calculated for training and test data set for architecture variants with different number of nodes in hidden layers. Comparative characteristics for models used for prediction of linear regression equation and models based on neural networks are presented. Quantitative results for predicting the severity class of the course of bronchial asthma are given. The results indicate the effectiveness of using neural network based models to predict outcomes that depend on a large number of factors.

Keywords

Correlation, perceptron, neural network, bronchial asthma, child

1. Introduction

Bronchial asthma is a noncommunicable disease that affects children. In 2019, the number of asthma patients was 262 million and 455,000 deaths due to this disease were reported [1]. The pathophysiological aspects of bronchial asthma are complex and diverse. Multifactoriality, the lack of reliable monopredictors of development and the peculiarities of the course of bronchial asthma cause the difficulty of diagnosing the disease in children [2]. The study of asthma pathogenesis mechanisms and the development of new effective medical drugs are aimed at determining their prognostic value in achieving control of disease symptoms. The problem of early diagnosis and timely prediction of the course of the disease remains acute.

To predict the bronchial asthma development, linear regression models are frequently utilized [3]. The work of Yan Zhao et al identified factors (age, parental asthma, early frequent wheezing, allergic rhinitis, eczema, allergic conjunctivitis, obesity and dust mite aeroallergen), and used a logistic regression model to predict asthma in school-age children [4]. Author Amani F. Hamad et al developed a model for predicting the risk of asthma in children using data on comorbid conditions among children and parents [5]. The use of a binary logistic regression model has improved the accuracy of diagnosing asthma in children and adults under 25 years of age in primary care by identifying the most valuable combination of predictors based on clinical assessment [6]. Studying the association between birth weight and asthma in children using univariate and multivariate logistic models identified factors (premature birth, age, sex, race, family poverty, health insurance, smoking, maternal age) that play an important role in the formation of diseases [7].

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ pihnastyi@gmail.com (O. Pihnastyi); olga.kozhyna.s@gmail.com (O. Kozhyna); yv.karpushenko@knmu.edu.ua (I. Karpushenko)

ORCID iD 0000-0002-5424-9843 (O. Pihnastyi); 0000-0002-4549-6105 (O. Kozhyna); 0000-0002-2196-8817 (I. Karpushenko)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Regression model relies on a non-invasive forecasting method and an expanded factors number. The identification of predictors for model development consists of analysing the observed value dependence on other factors. Because of the heterogeneity of childhood asthma, many predictors are considered: wheezing after exercise; wheezing causing dyspnoea; cough during exercise; atopic dermatitis and allergic sensitisation [8]. The use of CHART screening allows symptom-based identification of children at high risk of developing asthma as early as 3 years of age, developed using predictive models [9, 10]. The difference between models is the absence in the model of quantitative factors characterising the results of laboratory examinations. Linear regression four factors mode (hospitalisation, eczema, parental atopy, the positive and negative predictive value of specific Ig E to inhalant allergens) was studied in [11], which made it possible to increase the predictive value of the model. In [12], a simplified methodology allowing to construct the observed value dependence on quantitative factors was considered. When using a one-dimensional and three-dimensional model together instead of a four-dimensional one, the computational complexity of the process of constructing regression models is reduced. Another way to reduce the computational complexity is related to the use of approximate methods for building regression models [13]. The works used the technique of constructing approximate one- and two-parameter models (TSLP, severe, pillow feather, bronchial asthma in relatives of second generation, allergic rhinitis, atopic dermatitis, domestic dust) and using their combinations to analyse the degree of prognosis of the bronchial asthma course in children as a way of replacing multifactorial linear regression models was proposed.

To improve the accuracy of prognosis of the bronchial asthma course, models based on neural networks have been proposed.

To understand the heterogeneity of the disease and to identify asthma subtypes, cluster analysis was used based on the definition of four distinguishing features: age of onset, allergic sensitisation, severity and exacerbations in the previous year [14]. The authors M. Lovrić, I. Banić et al. applied neural network based models to predict treatment outcomes in children with mild to severe asthma by changes in asthma control, lung function (SPF1 and MEF50) and FENO values after 6 months of control medication [15]. A research paper [16] presents machine learning algorithms for asthma phenotype detection and prediction of clinical outcomes to find heterogeneous relationships between clinical features and outcomes [16]. In [17] a review of algorithms of machine learning methods for predicting asthma exacerbation was conducted to identify characteristic features of the disease course. Asthma prediction on conventional blood biomarkers using an improved classifier (AGEC) with affinity graph improves the accuracy of asthma prediction compared to five state-of-the-art prediction models [18]. An automatic machine learning model is used to develop a prediction model for severe asthma exacerbation [19]. The presented model demonstrates the ability to identify children who are not at risk of exacerbation. Data from retrospective electronic medical records (EMR) of patients were used to identify asthma signs in children in paediatric hospitals in China. To improve the quality of asthma diagnosis, models based on machine learning CatBoost, Naive Bayes, support vector machines (SVM) have been developed and are constantly being improved [20].

The aim of the study is to develop a method for predicting the severity of the bronchial asthma course, based on the factors numerous for prediction. The vast majority of studies devoted to prognosis of bronchial asthma in children are based on linear regression models that contain a small number of regressors (no more than ten) and one observed parameter. The feature of the study of the bronchial asthma severity is the dependence of the observed parameter on the factors numerous of regressors, which have a small correlation coefficient between the observed value and the regressor. The use of linear models based on a small number of regressors, as well as the lack of regression models for classifying the severity of bronchial asthma, does not allow obtaining the required accuracy of prediction results for clinical studies. In this regard, an urgent problem in conducting clinical research is the construction of nonlinear prediction models containing several dozen regressors in the presence of the required number of observable factors. The scientific novelty of this work is the construction of a classification nonlinear model based on a neural network to predict the severity of bronchial asthma, containing several dozen regressors.

2. Problem statement

90 children aged from 6 to 18 years were involved in the study of the course of bronchial asthma: the main group is 70 children diagnosed with bronchial asthma; the control group is 20 children. The

average age of children with bronchial asthma was 11 years. For each patient, information on 142 factors that could be the cause of bronchial asthma was collected, processed and analysed. The study is carried out in compliance with scientific ethics and biomedical standards [21]. Parents were interviewed about the patients' symptoms characteristic of bronchial asthma, as well as the patients' medical history. The survey data were added to the patient's materials. Clinical features of the disease and the results of laboratory methods of investigation were studied.

In this study, a prediction model based on a neural network was developed to predict the severity of the disease course. Selection criteria were introduced to select the m-factors of the model

$$r_{y_{\alpha} x_m} \rightarrow \max, \quad r_{x_m x_v} \rightarrow \min, \quad (1)$$

где $r_{y_{\alpha} x_m}$ is correlation coefficients between model factors model factor X_m and the observed parameter

Y_{α} ; $r_{x_m x_v}$ are correlation coefficients between factors X_m and X_v :

$$r_{x_k x_v} = \frac{\frac{1}{n} \sum_{i=1}^n (X_{ki} - m_{x_k})(X_{vi} - m_{x_v})}{\sigma_{x_k} \sigma_{x_v}}, \quad \sigma_{x_k} = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_{ki} - m_{x_k})^2}, \quad m_{x_k} = \frac{1}{n} \sum_{i=1}^n X_{ki}, \quad k = 1..K. \quad (2)$$

$$r_{y_{\alpha} x_k} = \frac{\frac{1}{n} \sum_{i=1}^n (X_{ki} - m_{x_k})(Y_{\alpha i} - m_{y_{\alpha}})}{\sigma_{x_k} \sigma_{y_{\alpha}}}, \quad \sigma_{y_{\alpha}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_{\alpha i} - m_{y_{\alpha}})^2}, \quad m_{y_{\alpha}} = \frac{1}{n} \sum_{i=1}^n Y_{\alpha i}, \quad \alpha = 1..Z. \quad (3)$$

Experimental data of laboratory clinical changes, which were used to calculate correlation coefficients $r_{y_{\alpha} x_m}$, $r_{x_m x_v}$ are presented in [22]. The numerical characteristics of the selected set of factors in accordance with the criteria (1) are given in Table 1.

Table 1
Numerical the factors characteristics

Code	Regressor name	m_x, m_y	σ_x, σ_y
X_1	Atopic dermatitis	0.0562	0.2303
X_2	Bronchial asthma in relatives of second generation	0.0658	0.2479
X_3	Allergic rhinitis	0.4494	0.4974
X_4	Sheep wool	0.5217	0.6507
X_5	Domestic dust	2.2319	1.1312
X_6	Pillow feather	0.7536	0.8059
X_7	Dog hair	0.5362	0,8090
X_8	Bronchial asthma in father	0.0864	0.2810
X_9	Age	11.0674	3,6220
X_{10}	CD25 10*3 cells	0.6937	0.3087
Y_1	SEVERE PERSISTENT	0.0444	0.2082
Y_2	MODERATE PERSISTENT	0.3111	0.4657
Y_3	MILD PERSISTENT	0.3111	0.4562
Y_4	INTERMITTENT	0.3333	0.4740

The calculated correlation coefficients values between the model factors $r_{x_m x_v}$, as well as between the model factors and the observed value are also $r_{y_{\alpha} x_m}$ presented in Table 2.

Table 2**Correlation coefficients values $r_{x_mx_v}$, $r_{y_\alpha x_m}$**

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_1	-	0.08	0.08	0.03	0.26	0.6	0.11	0.08	0.03	0.17
X_2	-0.08	-	0.10	0.21	0.25	0.29	0.00	-0.09	0.16	-0.22
X_3	0.08	0.10	-	-0.01	0.27	0.09	-0.19	0.24	0.16	-0.03
X_4	-0.03	0.21	-0.01	-	0.16	0.34	0.20	0.02	0.00	-0.17
X_5	0.26	0.25	0.27	0.16	-	0.19	0.12	-0.05	0.17	-0.07
X_6	0.06	0.29	0.09	0.34	0.19	-	0.11	-0.05	0.04	0.08
X_7	0.11	0.00	-0.19	0.20	0.12	0.11	-	-0.17	-0.02	-0.07
X_8	0.08	0.09	0.24	0.02	-0.05	-0.05	-0.17	-	0.12	-0.05
X_9	-0.03	0.16	0.16	0.00	0.17	0.04	-0.02	-0.12	-	-0.22
X_{10}	-0.17	-0.22	-0.03	-0.17	0.07	0.08	-0.07	-0.05	-0.22	-
Y_1	0.23	0.44	0.31	0.31	0.32	0.38	0.20	-0.10	0.18	-0.20
Y_2	-0.12	-0.22	0.23	-0.29	0.01	-0.28	-0.14	0.32	-0.17	-0.28
Y_3	0.02	-0.22	-0.08	-0.05	-0.14	-0.11	0.16	-0.05	-0.01	0.11
Y_4	-0.12	0.44	0.12	0.08	-0.11	-0.18	0.20	-0.05	-0.22	-0.18

To build a model for predicting the severity of bronchial asthma in children, a multilayer perceptron consisting of several layers of neurons, each of which is connected with the previous layer (from which it receives input data) and the subsequent layer (which it, in turn, affects), was used in this study. The task of classifying the severity of the course of bronchial asthma Y_α depending on the values of input factors X_m . Softmax activation function was used to form the output parameters. This approach guarantees that the output nodes take values from 0 to 1, and the sum of all values of the output nodes is equal to one, which can be associated with the probability of the degree of the bronchial asthma course Y_α .

3. Bronchial asthma severity prediction model

Linear models is studied in [23, 24] to predict of the disease course of bronchial asthma:

$$Y_{\alpha i} = w_{\alpha 0} + \sum_{m=1}^M w_{\alpha m} X_{mi}, \alpha = 1..Z. \quad (4)$$

Overcoming the limitation of a linear model of type (4) is achieved by including a nonlinear transformation mechanism in the presented model [25]:

$$Y_{\alpha i} = f\left(w_{\alpha 0} + \sum_{m=1}^M w_{\alpha m} X_{mi}\right), \quad (5)$$

where $f(x)$ - nonlinear transformation function. When using analytical methods to calculate the coefficients $w_{\alpha 0}$, $w_{\alpha m}$ in the linear regression model (4) and transformed linear regression model (5), as a rule, for each observed factor is performed independently [26]. On the other hand, the model coefficients are functions dependent on the values of the regressors X_m and the observed parameter Y_α :

$$w_{\alpha 0} = w_{\alpha 0}(Y_\alpha, X_1, \dots, X_M), \quad w_{\alpha m} = w_{\alpha m}(Y_\alpha, X_1, \dots, X_M). \quad (6)$$

Increasing the accuracy of predicting the values of the observed parameters is achieved by switching to numerical methods to calculate the coefficients $w_{\alpha 0}$, $w_{\alpha m}$ which depend on all the observed parameters Y_α :

$$w_{\alpha 0} = w_{\alpha 0}(Y_1, \dots, Y_Z, X_1, \dots, X_M), \quad w_{\alpha m} = w_{\alpha m}(Y_1, \dots, Y_Z, X_1, \dots, X_M). \quad (7)$$

Thus, the transition from the model consisting of -equations with independent coefficients (5) to the model represented by the system of -equations is made:

$$\begin{cases} Y_{1i} = f\left(w_{10} + \sum_{m=1}^M w_{1m} X_{mi}\right), \\ \dots\dots\dots \\ Y_{\alpha i} = f\left(w_{\alpha 0} + \sum_{m=1}^M w_{\alpha m} X_{mi}\right), \\ \dots\dots\dots \\ Y_{Zi} = f\left(w_{Z0} + \sum_{m=1}^M w_{Zm} X_{mi}\right). \end{cases} \quad (8)$$

In common, a unique transformation function $Y_{\alpha i} = f_{\alpha}(x_{\alpha})$ can be chosen for each observed quantity Y_{α} . Although the last system of equations contains nonlinear transformation functions for the output parameter, by introducing the inverse function of the $f^{-1}(y) = x$ for the reciprocal-ambiguous function $y = f(x)$ system of equations (8) is transformed into a linear system of equations for the transformed observables $Y_{T\alpha}$ relative to input factors X_m :

$$\begin{cases} Y_{T1i} = w_{10} + \sum_{m=1}^M w_{1m} X_{mi}, \\ \dots\dots\dots \\ Y_{T\alpha i} = w_{\alpha 0} + \sum_{m=1}^M w_{\alpha m} X_{mi}, \\ \dots\dots\dots \\ Y_{TZi} = w_{Z0} + \sum_{m=1}^M w_{Zm} X_{mi}, \end{cases} \quad Y_{T\alpha i} = f^{-1}(Y_{\alpha i}). \quad (9)$$

As the next step to build non-linear models for calculating the observed parameters Y_{α} uses a multi-layer architecture in which each k level (layer, $k = 1..L$) is represented by a separate system of equations:

$$\begin{cases} H_{11i} = f_1\left(w_{110} + \sum_{m=1}^M w_{11m} X_{mi}\right), \\ H_{12i} = f_1\left(w_{120} + \sum_{m=1}^M w_{12m} X_{mi}\right), \\ \dots\dots\dots \\ H_{1M_1i} = f_1\left(w_{1M_10} + \sum_{m=1}^M w_{1M_1m} X_{mi}\right), \end{cases} \begin{cases} H_{k1i} = f_k\left(w_{k10} + \sum_{m=1}^{M_{k-1}} w_{k1m} H_{(k-1)mi}\right), \\ H_{k2i} = f_k\left(w_{k20} + \sum_{m=1}^{M_{k-1}} w_{k2m} H_{(k-1)mi}\right), \\ \dots\dots\dots \\ H_{kM_ki} = f_k\left(w_{kM_k0} + \sum_{m=1}^{M_{k-1}} w_{kM_km} H_{(k-1)mi}\right), \end{cases} \begin{cases} Y_{1i} = f_Y\left(w_{Y10} + \sum_{m=1}^{M_L} w_{Y1m} H_{Lmi}\right), \\ \dots\dots\dots \\ Y_{\alpha i} = f_Y\left(w_{Y\alpha 0} + \sum_{m=1}^{M_L} w_{Y\alpha m} H_{Lmi}\right), \\ \dots\dots\dots \\ Y_{Zi} = f_Y\left(w_{YZ0} + \sum_{m=1}^{M_L} w_{YZm} H_{Lmi}\right). \end{cases} \quad (10)$$

Procedure for calculating the coefficients w_{kM_k0} , w_{kM_km} , w_{YZ0} , w_{YZm} requires a prepared dataset for training the neural network, in which each row of values of input factors X_m corresponds to the string of values of the observed parameters Y_{α} . One way to calculate the coefficients w_{kM_k0} , w_{kM_km} , w_{YZ0} , w_{YZm} is the method of error back propagation. The coefficients are calculated by the method of successive approximation. Stopping of calculations occurs when the accuracy of calculation of the values of the observed parameters Y_{α} reaches the specified accuracy or the number of approximations has exceeded the maximum permissible specified value. In direct propagation, intermediate values of variables are sequentially calculated and stored in the computational graph defined by the computational procedure. The calculation is performed sequentially from the input layer to the output layer. In back propagation, the gradients of intermediate variables and parameters are sequentially computed and stored. The computation is performed in reverse order sequentially from output layer to input layer.

A multilayer perceptron with an architecture containing a single M-M₁-Z (L=1) and two M-M₁-M₂-Z hidden layers (L=2), where M- number of input factors in the prediction model; M_k – the nodes

number in k hidden layer; Z - the nodes number in the output layer. The architecture of the neural network is shown in Fig. 1. The values of the neural network nodes are calculated in accordance with the system of equations (10).

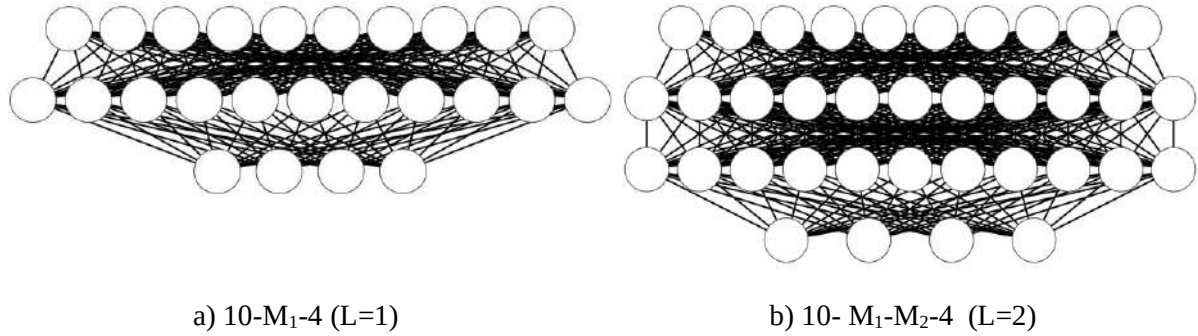


Figure 1: Neural network architecture in a model for predicting the severity of the course of bronchial asthma: (a) one M-M1-Z hidden layer (L=1); (b) two M-M1-M2-Z hidden layers (L=2).

For the hidden layer nodes, let us represent the activation function as a Sigmoid-function

$$f_1(x) = \frac{1}{1 + \exp(-x)}. \quad (11)$$

As the activation function for the last (output) layer, choose Softmax function:

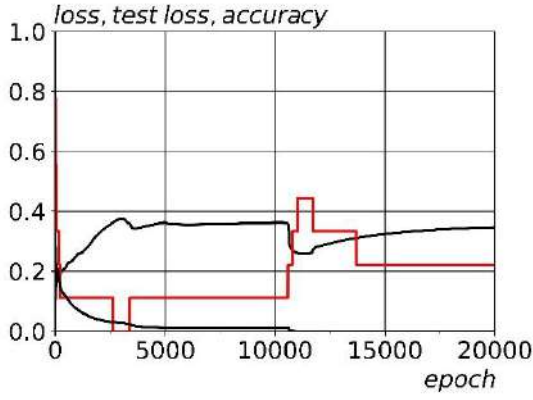
$$f_{Y\alpha}(x_\alpha) = \frac{\exp(-x_\alpha)}{\sum_{\alpha=1}^Z \exp(-x_\alpha)}. \quad (12)$$

This approach normalises the values of the output parameters of the bronchial asthma severity prediction model, optimises the discrete output spaces and treats these values as the probability of a patient having a disease corresponding to the bronchial asthma severity classification. The key point is the application of a probabilistic approach that allows discrete categories to be treated as examples of samples from a probability distribution. Thus, the use of softmax activation function for the output layer allowed us to transform the values of the output layer of the neural network into actual discrete probability distributions of bronchial asthma severity.

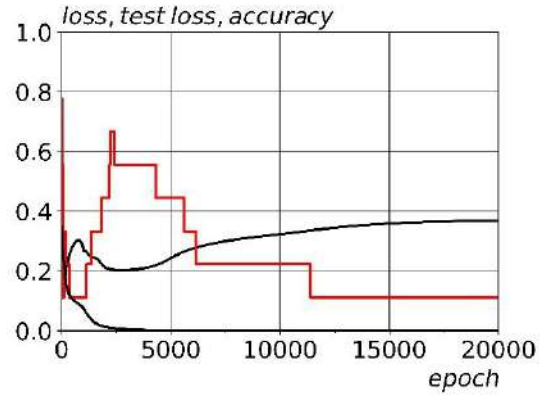
As the method of initialisation of model weights and bias, the normal law distribution of values with distribution parameters $N(0,1)$. As shown by numerical experiments, the method of initialisation of weights plays an important role in the construction of the prediction model, allows to accelerate the convergence of training and improve the quality of the model. When training the neural network in the prediction model, the following model hyperparameters are set: learning rate is $lr = 0.001$, maximum number of epochs is $epoch_{\max} = 20000$, the size of the packet used in training is $batch = 1$.

4. Analysing the results

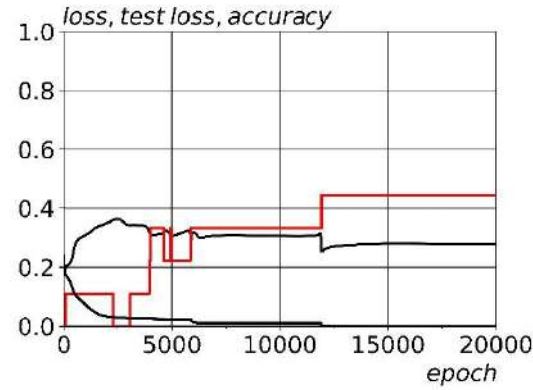
When building the prediction model, the sample for training the neural network is divided into two sets of data. The first dataset directly serves for training the neural network and makes up 80%, the second dataset (test dataset), which makes up 20%, is used to verify the prediction accuracy. The training process of the neural network with different architectures is shown in Fig. 2. Quantitative indicators characterising the quality of the neural network training process are presented in Table 3.



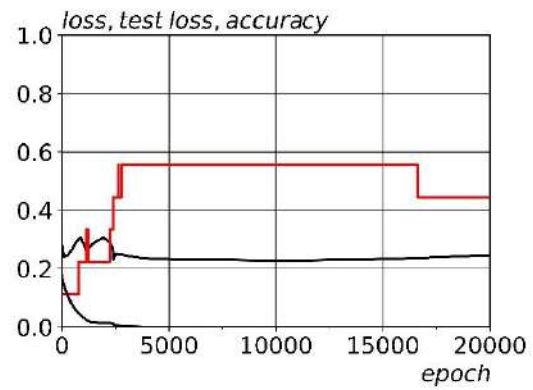
a) 10-15-4 (L=1)



b) 10-25-4 (L=1)



c) 10-10-10-4 (L=2)



d) 10-15-15-4 (L=2)

Figure 2: Neural network training results for different architectures with one hidden layer $M-M_1-Z$ (L=1) and two hidden layers $M-M_1-M_2-Z$ (L=2); the loss function for the learning dataset (loss) and the loss function for the test dataset (test loss) is black line; the prediction accuracy is red line.

Table 3
Learning rates of neural network for different architectures

	regression model	10-15-4 (L=1)	10-25-4 (L=1)	10-10-10-4 (L=2)	10-15-15-4 (L=2)
loss value, MSE	0.0713	5.77e-07	1.06e-07	3.94e-07	0.16e-07
best test loss value, MSE	0.5051	0.2607	0.2061	0.2674	0.2473
best test accuracy		0.4444	0.6677	0.4556	0.5556
best test epoch value		11027	2214	11905	2635

A test dataset was used to determine the prediction accuracy. The severity of the disease corresponded to the value of the predicted class having the maximum probability value. Table 3 shows the comparative analysis of the training results of the neural network with one and two hidden layers and different number of nodes in the hidden layer. When carrying out the comparative analysis, the random sequence of numbers used in the initialisation of the neural network weights was a fixed sequence of numbers. As would be expected, increasing the number of nodes for a neural network with both one and two hidden layers results in a significant decrease in MSE. Similar behaviour is observed for the MSE values corresponding to the test dataset. The decrease in MSE values leads to an increase in the prediction accuracy determined with the test dataset. As a comparison of the prediction quality, Table 3 presents the MSE value for the prediction model based on the linear regression model [27].

The coefficients of the linear regression equation for the above model are given in Table 4, for the calculation of which the same data set was used as for training the neural network. The MSE value for the test dataset in the neural network based prediction models is half that for the prediction model using the linear regression equation. Particular attention should be paid to the strong difference in the MSE

values obtained for the training dataset. The latter fact clearly demonstrates the property of the neural network to approximate the values for the test dataset. It is expected that as the sample size for training the neural network increases, a similar trend will be observed for the MSE values obtained for the test dataset of the dataset.

Linear regression mode coefficients for the value Severe Persisten [27].

No		$\sqrt{MSE}$	X_0	X_1	X_2	X_3	X_4	X_5	X_7	X_8	X_9	X_{10}	X_{12}
17	56	0.267	0.1617	0.1493	0.4965	-0.1139	0.3788	0.0314	0.0259	0.0775	0.0791	-0.0029	0.0004

To quantify the quality of prediction of the bronchial asthma course, Table 5 and Table 6 present the prediction values obtained for a model based on a neural network with a 10-15-4 architecture consisting of an input layer with ten nodes (model input factors X_m , Table 2), output layer with four nodes (observed model parameters Y_α , Table 2) and one hidden layer with fifteen nodes.

Table 5

Model prediction results for the training dataset

[illegible]

Table 6
Model prediction results for the test dataset

N#	initial				predict			
	SEVERE PERSISTENT	MODERATE PERSISTENT	MILD PERSISTENT	INTERMITTENT	SEVERE PERSISTENT	MODERATE PERSISTENT	MILD PERSISTENT	INTERMITTENT
1	0	0	0	1	0.0040	0.0012	0.8675	0.1273
2	0	0	0	1	0.0001	0.0030	0.9824	0.0145
3	0	1	0	0	0.0	0.7886	0.2009	0.0105
4	0	0	1	0	0.0002	0.9660	0.0131	0.0208
5	0	0	0	1	0.0200	0.3471	0.2856	0.3474
6	0	0	0	1	0.0473	0.1614	0.3864	0.4049
7	0	0	0	1	0.0089	0.3165	0.0480	0.6265
8	0	0	0	1	0.2180	0.0613	0.3427	0.3781
9	0	0	0	1	0.0016	0.0445	0.0721	0.8818

The accuracy of predicting the severity class of bronchial asthma using the developed model was determined as follows: for each class corresponding to the severity class, the prediction probability is calculated based on the Softmax function; the column with the maximum probability value for the training dataset is determined; the severity class for the resulting column is considered as the prediction result. The severity class of the disease course for each patient is determined according to the given value provided in the training dataset. This result is expected, which is explained by the low value of MSE corresponding to the training dataset. Table 6 gives the quantitative values of predicting of the bronchial asthma course for the test dataset. The bronchial asthma course severity was correctly predicted for 70% of the patients from the test dataset. This is an encouraging result compared to studies based on a linear regression model. To conduct the study, software libraries in Python were used for data processing and analysis: Pandas (version 2.0.0), Pytorch (version 2.0.0).

5. Conclusion

In this paper, a neural network based model for predicting the severity of the course of bronchial asthma in children has been reviewed. 142 factors that could be responsible for the occurrence of bronchial asthma disease were analysed. The training sample is divided into two parts, for training the neural network and testing the training results, in the proportion of 80/20. The model architecture is represented by a multilayer perceptron. The dependence of MSE value for training and test data set at different values of nodes in hidden layers has been analysed. A comparative analysis of MSE values for the model using linear regression equation for prediction and neural network based models is presented.

The developed model for predicting the severity of bronchial asthma, the foundation of which is a neural network, allowed to correctly determine the severity class of the disease for 70% of patients from the test dataset. For the remaining 30%, the model predicted the result as a neighbouring severity class. The result of further research is to investigate the dependence of the number of regressors of the model on the accuracy of the results of predicting the severity class of bronchial asthma. A critical issue for future research is to identify factors that can identify disease severity based on the results of clinical blood and urine tests. This will reduce the cost of conducting examinations, which require significant financial costs and conducting research in clinics with access to specialized and expensive equipment.

References

- [1] Global burden of 369 diseases and injuries in 204 countries and territories, 1990–2019: a systematic analysis for the Global Burden of Disease Study 2019, *Lancet*, 2020; 396(10258), pp. 1204-1222, [https://doi.org/10.1016/S0140-6736\(20\)30925-9](https://doi.org/10.1016/S0140-6736(20)30925-9).

- [2] WHO. Asthma/ WHO, 2023, <https://www.who.int/ru/news-room/fact-sheets/detail/asthma>
- [3] A. Fitzpatrick, L. Bacharier, D. Jackson et al. Heterogeneity of mild to moderate persistent asthma in children: confirmation by latent class analysis and association with 1-year outcomes, *J Allergy Clin Immunol Pract*, vol. 8(8), 2020, pp.2617-2627.e4, doi: 10.1016/j.jaip.2020.02.032.
- [4] Y. Zhao, J. Patel, X. Xu et al. Development and validation of a prediction model to predict school-age asthma in preschool children, *Pediatr Pulmonol*, vol. 58(5), 2023, pp. 1391-1400, doi: 10.1002/ppul.26331.
- [5] A. F. Hamad, L. Yan, M. J. Jozani et al. Developing a prediction model of children asthma risk using population-based family history health records, *Pediatr Allergy Immunol*, vol. 34(10), 2023, <https://doi.org/10.1111/pai.14032>.
- [6] L. Daines, L. J Bonnett, H. Tibble, A. Boyd, R. Thomas et al. Deriving and validating an asthma diagnosis prediction model for children and young people in primary care, *Wellcome Open Research*, vol. 8, 2023, <https://doi.org/10.12688/wellcomeopenres.19078.1>.
- [7] M. Ni, B. Li, Q. Zhang, J. Zhao, W. Li et al. Relationship between birth weight and asthma diagnosis: a cross-sectional survey study based on the National Survey of Children's Health in the U.S, *BMJ Open*, vol. 13, 2023, doi:10.1136/bmjopen-2023-076884.
- [8] R. Wang, A. Simpson, A. Custovic, P. Foden, D. Belgrave, C. Murray, Individual risk assessment tool for school-age asthma prediction in UK birth cohort, *Clin Exp Allergy*, , vol. 49(3), 2019, pp. 292-298, doi: 10.1111/cea.13319.
- [9] M. Reyna, R. Dai, M. Tran et al, Development of a symptom-based tool for screening of children at high risk of preschool asthma, *JAMA Network Open*, vol.5(10), 2022, doi:10.1001/jamanetworkopen.2022.34714.
- [10] D. Leisman, M. Harhay, D. Lederer, M. Abramson, A. A Adjei et al. Development and reporting of prediction models: guidance for authors from editors of respiratory, sleep, and critical care journals, *Crit CareMed*, vol. 48(5), 2020, pp. 623-633,doi:10.1097/CCM.0000000000004246.
- [11] N. A Boersma, R. WH Meijneke, J. C Kelder, C. K van der Ent, W. A F Balemans, Sensitization predicts asthma development among wheezing toddlers in secondary healthcare, *Pediatr Pulmonol*, vol. 52(6), 2017, pp. 729-736, doi: 10.1002/ppul.23668.
- [12] R. J Kurukulaarachy, S. Matthews, S. T Holgate, S. H Arshad, Predicting persistent disease among children who wheeze during early life, *Eur Respir J*, vol. 22(5), 2003, pp. 767-771, doi: 10.1183/09031936.03.00005903.
- [13] O. Pihnastyi, O. Kozhyna, Methods for constructing estimated two-factor linear regression models for diagnosing the severity of bronchial asthma in children, *Innovare Journal of Medical Sciences*, vol. 9(1), 2021; pp. 23-30, doi:10.22159/ijms.2021.v9i1.40408.
- [14] M. Deliu, TS. Yavuz, M. Sperrin M et al. Features of asthma which provide meaningful insights for understanding the disease heterogeneity, *Clin Exp Allergy*, vol. 48(1), 2018, pp. 39-47, doi: 10.1111/cea.13014.
- [15] M. Lovrić, I. Banić, E. Lacić, K. Pavlović, R. Kern et al, Predicting treatment outcomes using explainable machine learning in children with asthma, *Children (Basel)*, vol. 8(5), 2021, p. 376, doi: 10.3390/children8050376.
- [16] M. K Ross, J. Yoon, A. van der Schaar, M. van der Schaar, Discovering pediatric asthma phenotypes on the basis of response to controller medication using machine learning, *Ann Am Thorac Soc*, vol. 15(1), 2018, pp. 49-58, doi: 10.1513/AnnalsATS.201702-101OC.
- [17] N. A Molfino, G. Turcatel, D. Riskin, Machine learning approaches to predict asthma exacerbations: A narrative review, *Adv Ther*, vol. 41(2), 2024, pp. 534-552, doi: 10.1007/s12325-023-02743-3.
- [18] D. Li, S. Ebhohimhen Abhadiomhen, D. Zhou, Xiang-Jun Shen, L. Shi et al, Asthma prediction via affinity graph enhanced classifier: a machine learning approach based on routine blood biomarkers, *Journal of Translational Medicine*, vol. 22(1), 2024, doi:10.1186/s12967-024-04866-9.
- [19] G. Luo, S. He, B. L Stone, F. L Nkoy, M. D Johnson, Developing a model to predict hospital encounters for asthma in asthmatic patients: secondary analysis, *JMIR Med Inform* vol. 8(1), 2020, e16080, doi: 10.2196/16080.
- [20] G. Yu, Z. Li, S. Li, J. Liu, M. Sun et al, The role of artificial intelligence in identifying asthma in pediatric inpatient setting, *Ann Transl Med*, vol. 8(21), 2020, p.1367, doi: 10.21037/atm-20- 2.
- [21] UNESCO.ORG, Universal Declaration on Bioethics and Human Rights, 2005, <http://portal.unesco.org/en/ev.php>. 2005.

- [22] O. Kozhyna, O. Pihnastyi, Data structure of clinical research, *Human Health & Disease*, vol. 3(9), 2019, pp. 71-79.
- [23] A. J Farhan, D. M Kothalawala, R. J Kurukulaaratchy, R. Granell, A. Simpson et al, Prediction of adult asthma risk in early childhood using novel adult asthma predictive risk scores, *Allergy*, vol. 78(11), 2023, pp. 2969-2979. doi: 10.1111/all.15876.
- [24] A. F. Hamad, L. Yan, M. J Jozani, P. Hu, J. A. Delaney et al, Developing a prediction model of children asthma risk using population-based family history health records, *Pediatr Allergy Immunol*, vol. 34(10), 2023, doi: 10.1111/pai.14032.
- [25] C. Domínguez-Martína, A. Canoa, N. Díez-Mongea et al, Clinical performance of spirometry and respiratory oscillometry for prediction of severe exacerbations in schoolchildren with asthma, *Anales de Pediatría*, vol. 98(6), 2023, pp. 427-435, doi: 10.1016/j.anpede.2023.05.003
- [26] O. Pihnastyi, O. Kozhyna, T. Kulik, Linear Regression Approximate models for predicting severe course of bronchial asthma, *CEUR Workshop Proceedings*, vol. 3309, 2022, pp. 55-65.
- [27] O. Pihnastyi, O. Kozhyna, K. Voloshyn, Linear regression models for bronchial asthma severity prediction based on TSLP, *Proceedings of the 11-th International Conference "Information Control Systems & Technologies"*, vol. 3513, 2023, pp. 276-290.

Online Stacking Credibilistic Fuzzy Clustering for Data Stream Mining

Yevgeniy Bodyanskiy¹, Alina Shafronenko¹, Diana Rudenko¹ and Oleksii Tanianskyi¹

¹Kharkiv National University of Radio Electronics, Nauky ave 14, Kharkiv, 61166, Ukraine

Abstract

An important problem that arises when processing large amounts of observations is data compression to highlight the most essential information and identify certain latent factors that implicitly determine the nature of the phenomenon being studied. One of the most effective approaches to solving this problem is the apparatus of factor analysis, which has found wide application in problems of processing empirical data in various fields.

Fuzzy clustering is a popular approach for soft data partitioning, its use always encounters difficulties in solving the problems of processing high-dimensional real data with complex hidden distributions. This paper proposes a disclosure of a kind of stack fuzzy clustering method where the data is represented in a new feature space created by a stacking neural network. This approach aims to overcome the challenges associated with processing complex data and can bring significant improvements in the quality of clustering.

Keywords

Stack learning, data compression, fuzzy clustering, credibilistic fuzzy clustering, data stream mining

1. Introduction

Clustering is a technique in machine learning and data analysis that involves grouping a set of data points into subsets, or clusters, based on the similarity between them. Fuzzy clustering is a variation of traditional clustering methods that allows for more flexible and nuanced assignments of data points to clusters [1-5]. In contrast, fuzzy clustering allows data points to belong to multiple clusters simultaneously with varying degrees of membership. This reflects the inherent uncertainty or ambiguity present in real-world data.

The Fuzzy C-Means (FCM) algorithm, proposed by James Bezdek in 1973, is a prominent method in fuzzy clustering [6]. FCM assigns membership degrees to data points, indicating the likelihood of each point belonging to different clusters. This flexibility makes fuzzy clustering particularly useful in scenarios where data points may exhibit overlapping characteristics or uncertainty in their categorization.

Over the years, fuzzy clustering has found applications in diverse fields, including pattern recognition, image processing, and Data Mining. Researchers have developed various extensions and enhancements to the original FCM algorithm, addressing specific challenges and improving its adaptability to different data patterns.

The validity of fuzzy clustering solutions became a key focus, leading to the introduction of indices to assess the quality of clustering results. These indices help researchers and practitioners evaluate the effectiveness of fuzzy clustering algorithms in capturing meaningful patterns within datasets.

The evolution of fuzzy clustering has seen ongoing advancements, with researchers exploring sophisticated membership functions and integrating fuzzy clustering with other machine learning techniques. This integration has expanded the capabilities of fuzzy clustering, making it applicable to complex problems in large-scale data analysis.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ yevgeniy.bodyanskiy@nure.ua (Ye. Bodyanskiy); alina.shafronenko@nure.ua (A. Shafronenko); diana.rudenko@nure.ua (D. Rudenko), oleksii.tanianskyi@nure.ua (O. Tanianskyi)

ORCID 0000-0001-5418-2143 (Ye. Bodyanskiy); 0000-0002-8040-0279 (A. Shafronenko); 0000-0002-1792-5080 (D. Rudenko); 0009-0005-3491-4470 (O. Tanianskyi)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The era of big data has significantly influenced the field of clustering, including traditional clustering methods and the development of fuzzy clustering techniques. Deep learning in big data represents a powerful combination that has transformed various fields by enabling more sophisticated analysis, pattern recognition, and decision-making capabilities.

Recently, there has been significant research into leveraging deep learning to uncover meaningful data representations through neural networks. A notable area of exploration involves the integration of unsupervised clustering algorithms with stack neural networks. This synergy has become a vibrant field of research, aiming to jointly optimize the performance of deep learning models and clustering algorithms.

The goal of the work is propose the stack neuro-fuzzy system for Data Stream Mining using credibilistic approach and designed to work both in batch and its recurrent online version.

2. Neural Network Data Compression

An important problem that arises when processing large amounts of observations is data compression to highlight the most essential information and identify certain latent factors that implicitly determine the nature of the phenomenon being studied. One of the most effective approaches to solving this problem is the apparatus of factor analysis [7], which has found wide application in problems of processing empirical data in various fields: psychology, sociology, technology, economics, medicine, criminology, etc.

The basic idea of factor analysis, which allows for the presence of a priori unknown hidden factors, leads to the following informal task: by observing a large number of measured parameters (indicators), identify a small number of parameter-factors that mainly determine the behavior of the measured parameters, or in other words: knowing the values of a large number of measured functions parameters, set the appropriate values of the factor arguments common to all functions and restore the form of these functions.

The initial information for factor analysis is the $(N \times n)$ observation matrix:

$$X(N) = \begin{pmatrix} x_1(1) & x_2(1) & \dots & x_n(1) \\ x_1(2) & x_2(2) & \dots & x_n(2) \\ \dots & \dots & \dots & \dots \\ x_1(k) & x_2(k) & \dots & x_n(k) \\ \dots & \dots & \dots & \dots \\ x_1(N) & x_2(N) & \dots & x_n(N) \end{pmatrix} = \begin{pmatrix} x^T(1) \\ x^T(2) \\ \dots \\ x^T(k) \\ \dots \\ x^T(N) \end{pmatrix}, \quad (1)$$

that formed by an array of N n -th dimensional vectors $x(k) = (x_1, x_2, \dots, x_n)^T$ and autocorrelation matrix $(n \times n)$

$$R(N) = \frac{1}{N} \sum_{k=1}^N (x(k) - \bar{x}(N))(x(k) - \bar{x}(N))^T = \frac{1}{N} \sum_{k=1}^N \tilde{x}(k) \tilde{x}^T(k), \quad (2)$$

where

$$\bar{x}(k) = \frac{1}{N} \sum_{k=1}^N x(k), \quad \tilde{x}(k) = x(k) - \bar{x}(k) \quad (3)$$

vectors of measured indicators centered relative to the average of the data array.

One of the most common and effective methods for finding factors is the principal component method or component analysis, which is widely used in problems of data compression, pattern recognition, coding, image processing, spectral analysis, etc. and known in pattern recognition theory as the Karhunen-Loeve transform.

The task of component analysis is to project data vectors from the original n -dimensional space into a m -dimensional one ($m < n$) space of principal components and reduces to searching for a system $w_1, w_2, \dots, w_m$ orthonormal eigenvectors of the matrix $R(N)$ such that $w_1 = (w_{11}, w_{12}, \dots, w_{1n})^T$ corresponds to the largest eigenvalue λ_1 matrix $R(N)$, w_2 - second largest eigenvalue λ_2 , etc. In other words, the problem comes down to finding solutions to the matrix

equation:

$$(R(N) - \lambda_j I_n) w_j = 0$$

such, that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq \varepsilon \geq 0$ and $\|w_j\| = 1$.

The dimension of the space of principal components m is determined, as a rule, from empirical considerations and the required degree of compression of the data array.

Thus, in algebraic terms, solving a factor problem is closely related to the problem of eigenvalues and finding the rank of the correlation matrix; in a geometric sense, this is the problem of moving to a lower-dimensional space with minimal loss of information; in a statistical sense, this is the problem of finding a set of orthonormal vectors in the input space that "accept" the maximum possible variation of the data, and finally, in an algorithmic sense, this is the problem of sequentially determining a set of eigenvectors $w_1, w_2, \dots, w_m$ by optimizing a set of local criteria that form a global objective function

$$E^k = \frac{1}{k} \sum_{j=1}^m E_j^k = \frac{1}{k} \sum_{j=1}^m \sum_{p=1}^k (w_j^T \tilde{x}(p))^2$$

with constraints

$$w_j^T w_l = 0, j \neq l, w_j^T w_j = 1.$$

The first principal component w_1 can be found by maximizing the criterion:

$$E_1^k = \frac{1}{k} \sum_{p=1}^k (w_1^T \tilde{x}(p))^2 \quad (4)$$

by solving a nonlinear programming problem using uncertain Lagrange multipliers.

However, if data processing must be carried out in real time, neural network technologies come to the fore, among which the self-learning rule and E. Oja's neuron should be noted.

It is with the help of Oja's rule in the form:

$$\begin{cases} w_1(k+1) = w_1(k) + \eta(k) y_1(k) (\tilde{x}(k) - w_1(k) y_1(k)), \\ y_1(k) = w_1^T(k) \tilde{x}(k), w_1(0) \neq 0 \end{cases} \quad (5)$$

the first principal component can be isolated.

Next, following the procedure of standard principal component analysis, from each vector $\tilde{x}(k)$, $k = 1, 2, \dots, N$ its projection onto the first principal component is subtracted and the first principal component of the differences is calculated, which is the second principal component of the original data and the orthonormal first. The third principal component is calculated by projecting each original vector $\tilde{x}(k)$ into the first two components, subtracting this projection from $\tilde{x}(k)$ and finding the first principal component of the differences, which is the third principal component of the original data array. The remaining principal components are calculated recursively according to the described strategy.

It is this idea of recursive calculation of principal components that forms the basis of the algorithm proposed by T. Sanger [8] and in a modified form having the form [9]

$$\begin{cases}
w_j(k+1) = w_j(k) + \eta(k)e_j(k)y_j(k), \\
e_j(k) = e_{j-1}(k) - w_j(k)y_j(k), \\
y_j(k) = w_j^T(k)\tilde{x}(k), \quad w_j(0) \neq 0, \\
e_0(k) = \tilde{x}(k), \quad j = 0, 2, \dots, m, \\
\eta(k) = r^{-1}(k), \quad r(k) = \alpha r(k-1) + \|\tilde{x}(k)\|^2, \quad 0 \leq \alpha \leq 1.
\end{cases} \quad (6)$$

It is easy to see that the first principal component is calculated using the Oja algorithm, then the projection of the input vectors onto $w_1(k)$ are subtracted from the inputs and the differences are processed by the next neuron, etc.

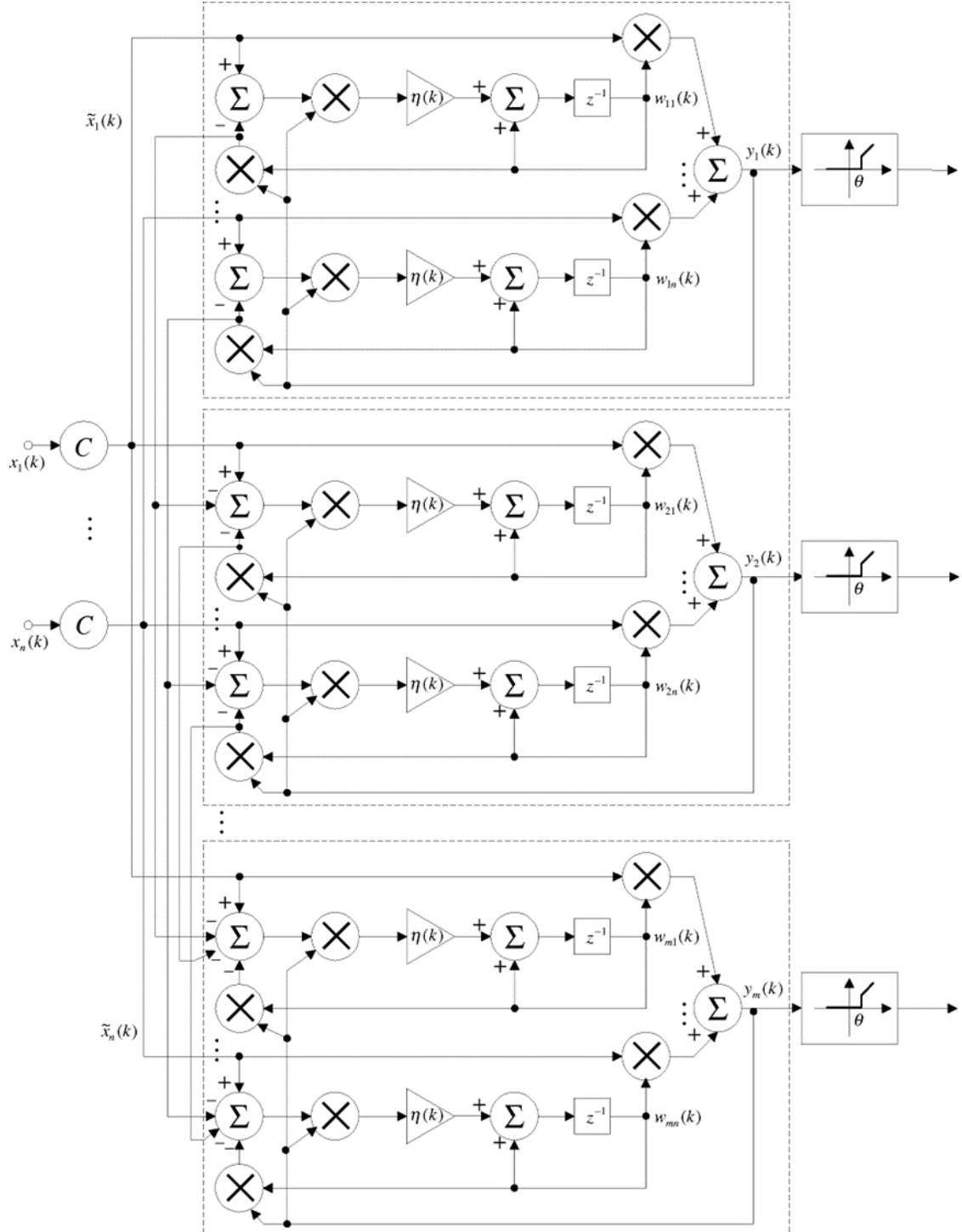


Figure 1: T. Sanger neural network

In Fig. 1 shows a diagram of a modified artificial T. Sanger's neural network, composed of E. Oya's neurons and implementing the algorithm (6).

The first layer of the network is formed by encoder neurons that pre-process signals by centering and normalizing them. Further signals $\tilde{x}_1(k), \tilde{x}_2(k), \dots, \tilde{x}_n(k)$ are processed in the second hidden layer formed by E. Oya's neurons, after which they are sent to the output layer formed by elements with activation rectifier functions with a dead zone

$$\psi(u) = \begin{cases} u, & \text{if } u \geq \theta, \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

which allows you to highlight informative signals $y_j(k)$ and filter out the noise.

The Sanger neural network is an effective means of compressing information with minimal loss of accuracy, but its capabilities are limited by the fact that, implementing essentially the standard technique of factor analysis, it solves a linear problem, while the main advantage of neural network technologies is the ability to work in purely nonlinear situations.

The problem of nonlinear factor analysis can be effectively solved using credibilistic theory and clustering analysis.

3. Fuzzy credibilistic clustering

Alternatively, to probabilistic and possibilistic procedures [10] it was introduced credibilistic fuzzy clustering approach using as its basis the credibility theory [11] and is largely devoid of the drawbacks of known methods.

The most common approach within the framework of probabilistic fuzzy clustering is associated with minimizing the goal function [12-14].

$$E(u_q(k), c_q) = \sum_{k=1}^N \sum_{q=1}^m u_q^\beta(k) d^2(x(k), c_q) \quad (8)$$

with constraints

$$\begin{cases} \sum_{q=1}^m u_q(k) = 1, \\ 0 < \sum_{k=1}^N u_q(k) < N. \end{cases} \quad (9)$$

Solution of nonlinear programming problem using the method of Lagrange indefinite multipliers leads to the well-known result [9, 11, 15]:

$$\begin{cases} u_q(k) = \frac{\left(d^2(x(k), c_q)\right)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m \left(d^2(x(k), c_l)\right)^{\frac{1}{1-\beta}}}, \\ c_q = \frac{\sum_{k=1}^N (u_q(k))^\beta x(k)}{\sum_{k=1}^N (u_q(k))^\beta} \end{cases} \quad (10)$$

coinciding with $\beta = 2$ a popular method of Fuzzy C-Means of J. Bezdek (FCM) [6].

If the data are fed to processing sequentially, the solution of the nonlinear programming problem (8), (9) using the Arrow-Hurwitz-Uzawa algorithm leads to an online procedure:

$$\begin{cases} u_q(k+1) = \frac{\left(d^2(x(k+1), c_q(k))\right)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m \left(d^2(x(k+1), c_l(k))\right)^{\frac{1}{1-\beta}}}, \\ c_q(k+1) = c(k) + \eta(k+1)u_q^\beta(k+1)(x(k+1) - c_q(k)). \end{cases} \quad (11)$$

The goal function of credibilistic fuzzy clustering has the form [6, 11] close to (8)

$$E(Cred_q(k), c_q) = \sum_{k=1}^N \sum_{q=1}^m Cred_q^\beta(k) d^2(x(k), c_q) \quad (12)$$

with "softer" than (9) constraints:

$$\begin{cases} 0 \leq Cred_q(k) \leq 1, \text{ for all } q \text{ and } k, \\ \sup Cred_q(k) \geq 0.5, \text{ for all } k, \\ Cred_q(k) + \sup Cred_l(k) = 1, \\ \text{for any } q \text{ and } k, \text{ for which } Cred_q(k) \geq 0.5. \end{cases} \quad (13)$$

It should be noted that the goal functions (8) and (12) are similar and that there are no rigid probabilistic constraints in (13) on the sum of the membership in (9).

In the procedures of credibilistic clustering, there is also the concept of fuzzy membership, which is calculated using the neighborhood function of the form:

$$u_q(k) = \varphi_q(d(x(k), c_q)) \quad (14)$$

monotonically decreasing on the interval $[0, \infty]$ so that $\varphi_q(0) = 1, \varphi_q(\infty) \rightarrow 0$.

Such a function is essentially an empirical similarity measure of [13, 15, 16] related to distance by the relation:

$$u_q(k) = \frac{1}{1 + d^2(x(k), c_q)}. \quad (15)$$

Note also that earlier it was shown in [14] that the first relation (10) for $\beta = 2$ can be rewritten as

$$u_q(k) = \left(1 + \frac{d^2(x(k), c_q)}{\sigma_q^2}\right)^{-1}, \quad (16)$$

where

$$\sigma_q^2 = \left(\sum_{\substack{l=1 \\ l \neq q}}^m d^2(x(k), c_l)\right)^{-1} \quad (17)$$

which is a generalization of the function (15) (for $\sigma_q^2 = 1$ (15) coincides with (17)) and satisfies all the conditions for (14).

In batch form the algorithm of credibilistic fuzzy clustering in the accepted notation can be written as

$$\begin{cases} u_q(k) = \left(1 + d^2(x(k), c_q)\right)^{-1}, \\ u_q^*(k) = u_q(k) \left(\sup_{l \neq q} u_l(k)\right)^{-1}, \\ Cred_q(k) = \frac{1}{2} \left(u_q^*(k) + 1 - \sup_{l \neq q} u_l^*(k)\right), \\ c_q = \frac{\sum_{k=1}^N (Cred_q(k))^\beta x(k)}{\sum_{k=1}^N (Cred_q(k))^\beta} \end{cases} \quad (18)$$

and in the online mode, considering (16), (17):

$$\begin{cases} \sigma_q^2(k+1) = \frac{1}{\sum_{\substack{l=1 \\ l \neq q}}^m d^2(x(k+1), c_l(k))}, \\ u_q(k+1) = \left(1 + \frac{d^2(x(k+1), c_q(k))}{\sigma_q^2(k+1)}\right)^{-1}, \\ u_q^*(k+1) = \frac{u_q(k+1)}{\sup_{l \neq q} u_l(k+1)}, \\ Cred_q(k+1) = \frac{1}{2} \left(u_q^*(k+1) + 1 - \sup_{l \neq q} u_l^*(k)\right), \\ c_q(k+1) = c_q(k) + \eta(k+1) Cred_q^\beta(k+1) (x(k+1) - c_q(k)). \end{cases} \quad (19)$$

From the point of view of computational implementation, algorithm (19) is not more complicated than procedure (11) and, in the general case, is its generalization to the case of credibilistic approach to fuzzy clustering.

4. Experimental research

Conducting experimental studies and comparative analysis of the quality of data clustering using various metrics allows you to objectively assess the effectiveness of the developed method in accordance with analogues. To estimate the quality of the method we used quality criteria partitioning into clusters such as [3, 6]:

- Partition Coefficient (PC);
- Classification Entropy (CE);
- Partition Index (SC);
- Separation Index (S);
- Xie and Beni's Index (XB);
- Dunn's Index (DI).

Partition Coefficient (PC): measures the amount of "overlapping" between clusters.

Classification Entropy (CE): it measures the fuzzyness of the cluster partition only, which is similar to the Partition Coefficient.

Partition Index (SC): is the ratio of the sum of compactness and separation of the clusters. It is a sum of individual cluster validity measures normalized through division by the fuzzy cardinality

of each cluster. SC is useful when comparing different partitions having equal number of clusters. A lower value of SC indicates a better partition.

Separation Index (S): on the contrary of partition index (SC), the separation index uses a minimum-distance separation for partition validity.

Xie and Beni's Index (XB): it aims to quantify the ratio of the total variation within clusters and the separation of clusters. The optimal number of clusters should minimize the value of the index.

Dunn's Index (DI): this index is originally proposed to use at the identification of "compact and well separated clusters".

So the result of the clustering has to be recalculated as it was a hard partition method. The specific information of the data sets is shown in Table 1.

Table 1

Information of the data dets

Data set	Observations	Clusters
Parkinson's telemonitoring	5875	21
Superconductor temperature prediction	10000	81

Table 2

Results of experiments for Parkinson's telemonitoring data set

Methods	PC	CE	SC	S	XB	DI	PC
Online Stack Fuzzy Credibilistic Clustering for Data Stream Mining	9.1249e-07	-2.3308e-04	0.3733	48.7067	24.4028	0.2005	0.582e-13
SOM based on possibilistic fuzzy clustering	0.3808	0.21415	0.00715	0.7473e-04	1.92845	0.01375	0.3954
SOM based on probabilistic fuzzy clustering	0.4731	0.05725	0.2395	0.0032	1.7309	0.1699	0.27535

Table 3

Results of experiments for Superconductor data set

Methods	CE	SC	S	XB	DI	PC	CE
Online Stack Fuzzy Credibilistic Clustering for Data Stream Mining	-0.00022	369360	271 150 000	135 900 000	0.0109	4.56245e-07	-0.000228
SOM based on possibilistic fuzzy clustering	0.1903	0.000366	0.0000034	2.8555	0.00585	0.38085	0.1903
SOM based on probabilistic fuzzy clustering	0.31965	4.29665	0.02415	0.5375	0.05075	0.4731	0.31965

Table 4
A comparison of accuracy

Data	Methods	Accuracy	
		Highest	Mean
Parkinson's telemonitoring	Online Stack Fuzzy Credibilistic Clustering for Data Stream Mining	61.68	60.98
	SOM based on possibilistic fuzzy clustering	61.75	61.30
	SOM based on probabilistic fuzzy clustering	62.68	62.98
Superconductor temperature prediction	Online Stack Fuzzy Credibilistic Clustering for Data Stream Mining	63.27	58.45
	SOM based on possibilistic fuzzy clustering	64.68	55.55
	SOM based on probabilistic fuzzy clustering	65.45	57.33

Table 5
Comparative characteristics of the average error with different number of observations (%)

Algorithm	50	t	100	t	150	t
FCM	1.62	1.19	1.35	2.55	0.98	3.03
SOM based on probabilistic fuzzy clustering	1.66	1.62	1.32	2.72	0.99	3.12
SOM based on possibilistic fuzzy clustering	1.22	1.15	1.02	2.02	0.75	2.10
Online Stack Fuzzy Credibilistic Clustering for Data Stream Mining	0.68	1.00	0.45	1.25	0.12	1.33

5. Discussions

Upon analyzing the results acquired, it can be inferred that irrespective of the volume of the initial data provided, the processing through the proposed method exhibits comparable speed and clustering quality when contrasted with established clustering algorithms and methodologies.

The obtained results confirm that the performance stack neuro-fuzzy system is better than other network structures, and it can be a viable structure for Data Stream Mining.

The results of accuracy that demonstrated in Table 4 confirm that proposed method online stack fuzzy credibilistic clustering for Data Stream Mining time more superiority regardless of the number observations that are fed on clustering process.

Based on the experimental findings, it is advisable to endorse the proposed system for practical application in addressing the challenges associated with automatic clustering of large datasets.

6. Conclusion

The problem of fuzzy clustering of data streams by stack neuro-fuzzy system is considered. In the paper was proposed the stack neuro-fuzzy system for Data Stream Mining using credibilistic approach and designed to work both in batch and its recurrent online version.

The network shows that stack structures based on fuzzy models can be applicable in data clustering. The proposed stack neuro-fuzzy system is quite simple in numerical implementation and can use the well-known online fuzzy clustering methods intended for solving Data Stream Mining problems.

Future research endeavors could explore the potential of employing stack neuro-fuzzy systems for fuzzy clustering of data streams, aiming to address the complexities inherent in automatic clustering of big data.

Acknowledgements

The work is supported by the state budget scientific research project of Kharkiv National University of Radio Electronics "Deep hybrid systems of computational intelligence for data stream mining and their fast learning" (state registration number 0119U001403).

References

- [1] GAs, Genetic Algorithms. "On Genetic-Fuzzy Data-Mining Techniques." (2023).
- [2] Liu, Hongfu, et al. "Infinite ensemble for image clustering." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.
- [3] Xu, Jindong, et al. "A fuzzy C-means clustering algorithm based on spatial context model for image segmentation." International Journal of Fuzzy Systems 23 (2021): 816-832.
- [4] Gong, Maoguo, et al. "Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images." IEEE Transactions on Fuzzy Systems 22.1 (2013): 98-109.
- [5] Rani, V., and S. Kumar. "Dissimilarity measure between intuitionistic Fuzzy sets and its applications in pattern recognition and clustering analysis." Journal of Applied Mathematics, Statistics and Informatics 19.1 (2023): 61-77.
- [6] Bezdek, James C., Robert Ehrlich, and William Full. "FCM: The fuzzy c-means clustering algorithm." Computers & geosciences 10.2-3 (1984): 191-203.
- [7] Cureton, Edward E., and Ralph B. D'Agostino. Factor analysis: An applied approach. Psychology press, 2013." (1967).
- [8] Sanger, Terence D. "Optimal unsupervised learning in a single-layer linear feedforward neural network." Neural networks 2.6 (1989): 459-473.
- [9] Specht, Donald F. "A general regression neural network." IEEE transactions on neural networks 2.6 (1991): 568-576.
- [10] Höppner, Frank, et al. Fuzzy cluster analysis: methods for classification, data analysis and image recognition. John Wiley & Sons, 1999.
- [11] Zhou, Jian, et al. "Credibilistic clustering: the model and algorithms." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 23.04 (2015): 545-564.
- [12] Zhang, Yong, et al. "Possibilistic c-means clustering based on the nearest-neighbour isolation similarity." Journal of Intelligent & Fuzzy Systems 44.2 (2023): 1781-1792.
- [13] Hu, Zhengbing, et al. "Fuzzy clustering of incomplete data by means of similarity measures." 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON). IEEE, 2019. doi: 10.1109/UKRCON.2019.8879844
- [14] Shafronenko, Alina, et al. "Online credibilistic fuzzy clustering of data using membership functions of special type." CMIS. 2020.
- [15] Bodyanskiy, Yevgeniy, Alina Shafronenko, and Sergii Mashtalir. "Online robust fuzzy clustering of data with omissions using similarity measure of special type." Lecture Notes in Computational Intelligence and Decision Making: Proceedings of the XV International Scientific Conference "Intellectual Systems of Decision Making and Problems of Computational Intelligence" (ISDMCI'2019), Ukraine, May 21–25, 2019 15. Springer International Publishing, 2020.
- [16] Bodyanskiy, Ye V., A. Yu Shafronenko, and I. N. Klymova. "Online fuzzy clustering of incomplete data using credibilistic approach and similarity measure of special type." Radio Electronics, Computer Science, Control 1 (2021): 97-104.

Recognition of images of wavelet spectra of Chirp signals using a neural network

Yuri Taranenko^{1,†}, Danilo Onufrienko², Olha Oliinyk^{3,*†} and Valerii Lopatin⁴

¹ Company «Likopak» st. Kachalova, 1, Dnipro, 49005, Ukraine

² National Technical University "Kharkiv Polytechnic Institute", st. Kirpycheva, 2, Kharkiv, 61000, Ukraine

³ College of Radio Electronics, st. Stepan Bandera, 18, Dnipro, 49000, Ukraine

⁴ Poljakov Institute of Geotechnical Mechanics of the National Academy of Sciences of Ukraine st. Simferopolska., 2a, Dnipro, 49005, Ukraine

Abstract

The article developed an algorithm for recognizing images of wavelet spectra of Chirp-type radar signals. Known augmentation methods cannot be used to expand the data set for training and improving the convolutional neural network. Generation of images of signal spectra is carried out with the addition of additive noise to the Chirp signal. The frequency modulation coefficient changes regardless of the presence of amplitude modulation by a half-sine wave. When preparing the data, the limitation of the power range of the Gaussian noise additively added to the signal was applied. Limitation is carried out in the frequency domain by comparing the wavelet coherence of the noise and the noisy signal using the example of the most common signal with linear frequency modulation of the Chirp type. It is shown that the wavelet autocohereance of Gaussian white noise has a constant value over the entire range of noise power variation. At the same time, the numerical value of autocohereance depends exclusively on the choice of wavelet. The wavelet autocohereance of the noisy signal when the noise power changes intersects with the noise autocohereance at the power value, which, in addition to the wavelet, depends on the frequency modulation coefficient. The described procedure for preparing spectrum images for processing in a neural network increases the probability of recognizing a given type of signal due to the exclusion of signals that cannot be recognized due to the lack of distinction from noise. For each continuous wavelet, such a level of non-stationarity is determined, at which a noisy signal can be recognized. This allows you to expand the database. Perform augmentation by changing the wavelet, as well as amplitude modulation of the signal. The effectiveness of the developed model was evaluated and the results were compared with known analogues. The trained neural network model for image recognition of continuous wavelet spectra using the example of the Chirp signal provides up to 100% accuracy of image detection and classification (the best result of the analogue is up to 95.7%).

Keywords

augmentation, wavelet-spectrum, convolutional neural networks, wavelet-autocohereance

1. Introduction

Since the appearance of radar with pulse compression, the Chirp (Compressed High-Intensity Radiated Pulse, LFM) signal has become one of the most common forms of radar signal [1].

The radar signal is modulated by phase or frequency, and this makes it possible to separate targets in space when receiving signals using special methods, the echoes of which intersect [2]. It should also be noted that pulse compression technology was developed for conditions where

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ tatanen@ukr.net. (Yu. Taranenko); onufrienkoresearcher@gmail.com (D. Onufrienko); oleinik_o@ukr.net (O.Oliinyk); vlop@ukr.net. (V. Lopatin)

🆔 0000-0003-2209-2244 (Yu. Taranenko); 0000-0002-53851365-798X (D. Onufrienko); 0000-0003-2666-3825 (O.Oliinyk); 0000-0003-2448-0857 (V. Lopatin)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

noise in the receiver always has a wide frequency band and random distribution [3]. It was believed that the spectral density of the noise is quite small compared to the echo signal. Since the noise intensity in the signal after the compression filter is significantly reduced, it makes it possible to identify the target in cases where the amplitude of the echo signal is less than the noise level [3, 4].

By changing the signal modulation to nonlinear frequency modulation (NLFM), better performance values in terms of peak voltage to side lobe ratio (PSLR) can be achieved. Changing the modulation of the signal provides a mitigating effect of masking nearby targets and can generally increase the useful dynamic range [5]. When adding appropriate amplitude modulation, PSLR can reach very low values (about -60 dB) [6].

Noise Radar Technology (NRT) is a viable alternative to deterministic waveforms. It uses pseudo-random waveforms to implement the noise process [6]. Common to NRT and deterministic radar technology is the use of a matched filter or an approximation thereof to maximize the signal-to-noise ratio (SNR). However, a noisy radar is capable of transmitting an almost unlimited set of realizations ("sampling functions") of a random process with an appropriate matched filter adjusted in real time to implement the well-known "correlation receiver" [7]. Thus, the main advantage of noise radar technology is the ability to work with highly noisy signals, in conditions of interference, since interference is perceived by the radar as a kind of noise [6].

The greatest contribution to the development of the theory of radar signal processing was made by K. Shanon, D. Flynn, S. Kay, A. Widoms, and R. L. Kriplin. Numerous publications in recent years by A. Gayvel, E. Liu, E. Monge, J. Delinget and many others confirm that the development and improvement of devices for the formation and processing of radar signals is an urgent task, especially for our country during a full-scale war. LFM signals can rightly be considered today as one of the elements of ensuring information security in telecommunication systems [8].

2. Analysis of literary data and statement of the problem

Recognition of images of noisy signals is an urgent task for many fields of technology. The rapid development of unmanned aerial vehicle technologies also requires solving the problems of image recognition in real time. The task is further complicated by the fact that in real conditions more complex forms of radar signals are used than those that have been well studied by researchers [9, 10]. Accordingly, there is a need to improve signal processing algorithms and image preparation for analysis and classification.

The development of machine learning made it possible to implement neural networks for processing signals of various types: data from sensors [11], audio signals [12] and images [13]. In recent years, various architectures of neural networks have been developed, which are designed to solve different types of problems. Neural network training methods are constantly self-improving and developing, retraining is reduced, and learning speed is increased. The latest models are characterized by increased resistance to noise and changes in input data, which increases their efficiency. The scaling of neural networks allows processing large data sets and solving complex tasks, including the processing of radar signals [14].

One of the most common tools for image classification and recognition today is a convolutional neural network (CNN) [15]. As a classification method, the convolutional neural network was developed and became the most widely used in computer vision technology [16].

The restraining factor in the development of CNN models is the presence of an output analog signal. A promising approach is the transition to time-frequency images using signal classification in convolutional neural networks. A difficult issue is the analysis of non-stationary signals. The use of wavelet transformation in time-frequency image research shows good results [17].

The accuracy of classification of wavelet spectrum images depends on the ability to learn the similarity of the neural network. The authors have already investigated the problem of not having enough data set for model training [18]. To increase the representativeness of the data set and improve the performance of neural networks, in particular, in the areas of computer vision and image processing, various augmentation methods are used [19].

Geometric transformations such as rotations, shifts, mirroring, scaling change the spatial position of the object and the model is able to learn at different positions of the objects. Changing the brightness, contrast and color allows you to increase the variety in the training set and improve the model's robustness to changes in lighting and color. Adding noise corresponding to different distributions helps train the model to effectively recognize objects in environments where different levels of noise are present. But in the given task, the known methods of augmentation do not allow to obtain the desired result, since the number of classification groups is unknown in advance during object recognition.

The second factor affecting classification accuracy is the influence of conversion parameters, frequency modulation coefficient, and noise added to the signal additively. The authors do not provide data on the effect of changes in noise power on Chirp-type signals. There is also no information on the effect of noise autocorrelation on the wavelet autocorrelation of a noisy signal, the use of additional signal modulation, and the visualization of these changes in the image of wavelet spectra.

The purpose of this article is to develop an algorithm for recognizing images of wavelet spectra of Chirp-type radar signals based on the numerical value of the frequency modulation coefficient while limiting the power of additively added noise by comparing the autocorrelation of the signal and noise in the frequency domain using a convolutional neural network model. As an algorithm for the image augmentation procedure in the neural network, the method of changing continuous wavelets is investigated, and the division of wavelet spectrum images into classes is performed by checking the homogeneity of the class according to the Shannon entropy value.

3. Algorithm for image recognition of continuous wavelet spectra of Chirp-type signals

3.1. Mathematical model of the Chirp signal in the time domain, taking into account the variable coefficient of linear frequency and amplitude modulation

We will use the ratio for the Chirp signal with linear frequency modulation:

$$x(t_i) = \text{anp} \cos(2\pi f_0 t_i + \pi \beta t_i^2) + \eta_i, \quad (1)$$

where anp – signal amplitude; f_0 – initial value of the frequency; β – coefficient of linear frequency modulation; η uncorrelated Gaussian noise, mathematical expectation is zero..

To take into account the amplitude modulation, we add a multiplier in the form of half-sine waves ($\text{anp} \sin(\pi t)$):

$$x(t_i) = \text{anp} \sin(\pi t) \cos(2\pi f_0 t_i + \pi \beta t_i^2) + \eta_i. \quad (2)$$

The modulation band to use the ratios must satisfy the inequality $\beta < 0.5N$, where N – length of the signal, for our case $N=2048$, that is, the condition for $\beta_{\max}=512 < 1024$ is fulfilled.

Let's examine the Fourier spectra of the Chirp signal by changing the coefficient of linear frequency modulation using linear frequency modulation and adding amplitude modulation by a half-sine wave (Fig. 1).

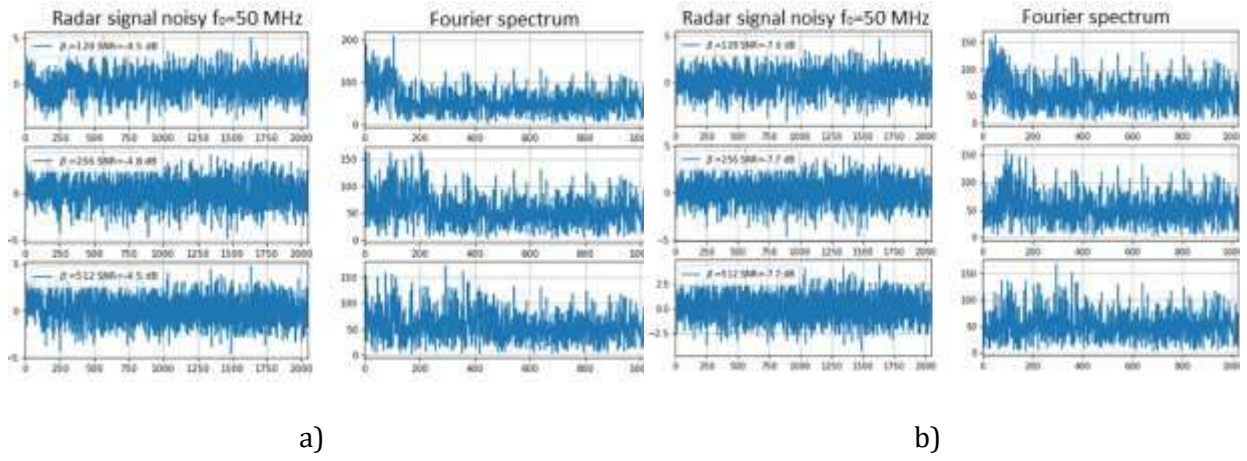


Figure 1: Fourier spectra of the Chirp signal with parameters $f_0=50$ MHz, $\beta=128; 256; 512$ for signals: a) according to ratio (1); b) by relation (2)

The results obtained in the graphs of Fig. 1 show that for the same root mean square deviation of the additive noise with additional amplitude modulation by a half-sine according to the ratio (2), the relative noise level in decibels is higher.

3.2. Mathematical model of image generation of continuous wavelet spectra for Chirp-type signals in the frequency domain

Wavelet coefficients will be determined by the formula for a continuous wavelet spectrum [9, 20]:

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt, \quad (3)$$

$f(t)$ – signal with a random component; $\psi\left(\frac{t-b}{a}\right)$ – the basic wavelet can be chosen from the list cgau1, cgau2, cgau3, cgau4, cgau5, cgau6, cgau7, cgau8, cmor, fbsp, gaus1, gaus2, gaus3, gaus4, gaus5, gaus6, gaus7, gaus8, mexh, morl, shan; $a \neq 0$ scale parameter; $b \geq 0$ – shift parameter [20]. The studied data are discrete, therefore formula (1) will be presented in the form, by selecting two arrays for the scales of *coeffs* for shifts *fred*:

$$coeffs, fred = \frac{\Delta t}{\sqrt{a}} \sum_{i=0}^{N-2} x(t_i) \psi\left(\frac{t_i - b}{a}\right). \quad (4)$$

To exclude the dependence on the displacement b , we obtain a representative amplitude of the inhomogeneity of the scale *coeffs* for the displacements *fred*. The formation of an experimental data set (Dataset) for training a convolutional neural network will be carried out by generating images of signal spectra with the addition of additive noise to the Chirp signal. We will change the frequency modulation coefficient regardless of the presence of amplitude modulation by a half-sine wave (Fig. 2).

The results obtained in the graphs of Fig. 2 for images of wavelet spectra, as well as for Fourier spectra, confirm the preliminary conclusion to Fig. 1 that for the same root mean square deviation of additive noise with additional amplitude modulation by a half-sine according to the ratio (2), the relative noise level in decibels is higher.

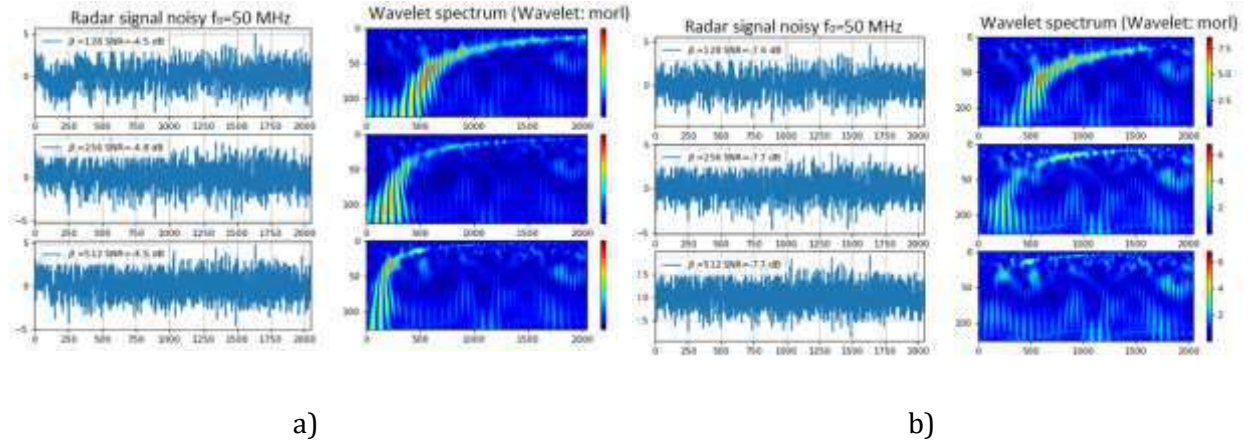


Figure 2: Wavelet-spectra of the Chirp signal for the wavelet morl, $f_0=50$ MHz, $\beta=128; 256; 512$ for signals a) according to ratio (1); b) by relation (2)

3.3. The method of detecting the noise threshold in the frequency domain

Consider the cross wavelet spectrum to check the level of influence of various interferences on the signal: white Gaussian noise and additively added noise. To assess the impact, we use autocoherece, which was already used in previous works to solve a similar problem [21]:

$$\left. \begin{aligned} S_x &= \frac{\Delta t}{\sqrt{a}} \sum_{i=0}^{N-2} x(t_i) \psi\left(\frac{t_i - b}{a}\right), \\ S_y &= \frac{\Delta t}{\sqrt{a}} \sum_{i=0}^{N-2} \frac{dx(t_i)}{dt_i} \psi\left(\frac{t_i - b}{a}\right), \\ \kappa^2 &= \frac{|S_{xy}|^2}{|S_x|^2 |S_y|^2}. \end{aligned} \right\} \quad (5)$$

where $\psi(\frac{t-b}{a})$ – the wavelet function; S_x – scale-time spectrum of the signal $x(t)$; S_y – scale-time spectrum of the derivative signal $\frac{dx(t_i)}{dt_i}$; a – scale factor; b – elimination of the signal along the time axis; κ^2 – the square of the coherence coefficient, which varies from zero to one, characterizes the level of stationarity of signal observation in the presence of noise.

It was established that for Gaussian noise, autocoherece according to relation (5) does not depend on the power of the noise. It is shown that the self-coherence depends only on the wavelet with the help of which a number of large-scale wavelet noise coefficients are formed (Fig. 3). The results of studies of the influence of a wider list of wavelet types will be given in future publications.

For the case under consideration, the mexh wavelet provides the minimum self-coherence and, accordingly, the maximum non-stationarity. Let's determine the noise threshold for the Chirp signal for this wavelet according to ratios (1), (2) with the noise measured in decibels. Autocoherece according to relation (5) is used twice: first for noise, and then for a signal with the addition of noise (Fig. 4).

Based on the obtained results (Fig. 4), when preparing the DataSet for the convolutional neural network model, we will have to limit the noise levels depending on the value of the modulation coefficients and the type of wavelet.

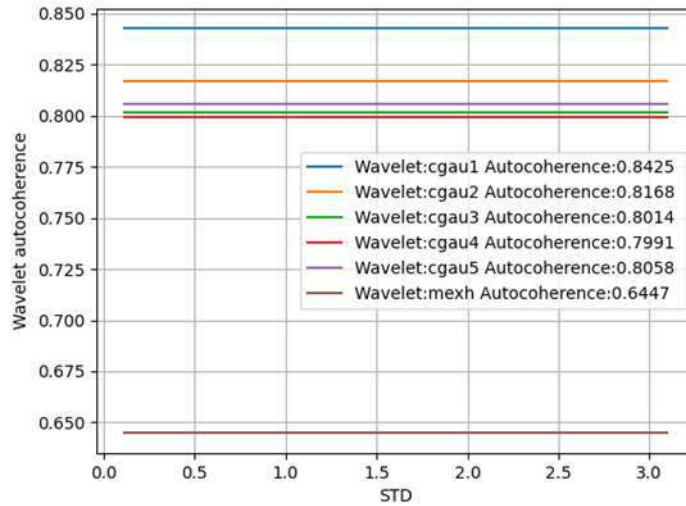


Figure 3: Dependence of the autocohereance of Gaussian noise on a continuous wavelet

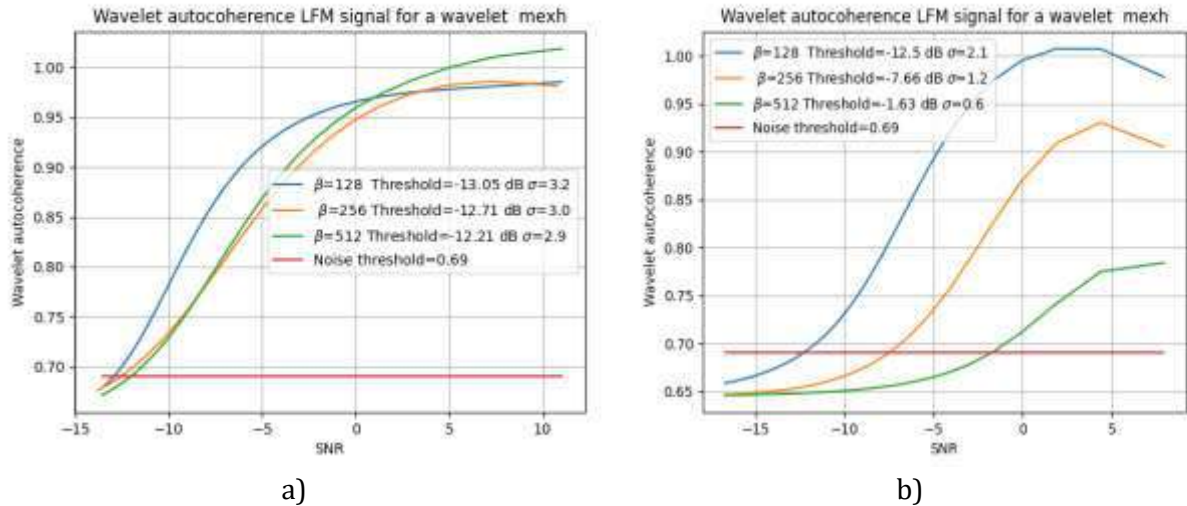


Figure 4: Graphs for determining the noise threshold based on the equality of non-stationarity of the signal and noise for the signal: a) for relation (1); b) for relation (2)

3.4. Peculiarities of the algorithm for preparation of the experimental data set CNN

Convolutional neural networks use images as input data. In this way, special procedures for recognizing specific elements or symbols can be introduced into the CNN architecture [22]. Changing the architecture of multilayer models such as ResNet50, VGG16, VGG19 by using an additional layer does not provide a solution to the problem, complicates fine-tuning of the model and increases the training period.

By analogy with previous works, the TensorFlow framework was used to build a neural network model - a computing library for building neural networks of various architectures with the Keras deep learning library, which provides full access to the scalability and cross-platform capabilities of TensorFlow [23]. Due to the lack of a dataset required for research in the Keras library, it is necessary to create your own experimental dataset. The block diagram of the developed algorithm for preparing wavelet spectrum images for the built high-precision neural network is shown in Fig. 5.

The accuracy of the deep learning model depends on the procedure for extracting data features, which is built on a large number of various experimental data [24]. In the absence of

such data, the augmentation procedure is used - a method of artificial generation of a "training" data set, which is used to train the model. Usually, augmentation is implemented by the Tensorflow image generator, with the help of which each "training" image is modified randomly: it is rotated to a certain angle, changes its size or contrast, is mirrored [25]. Thanks to this approach, an artificial increase in the representativeness of the data set for neural network training is achieved [26].

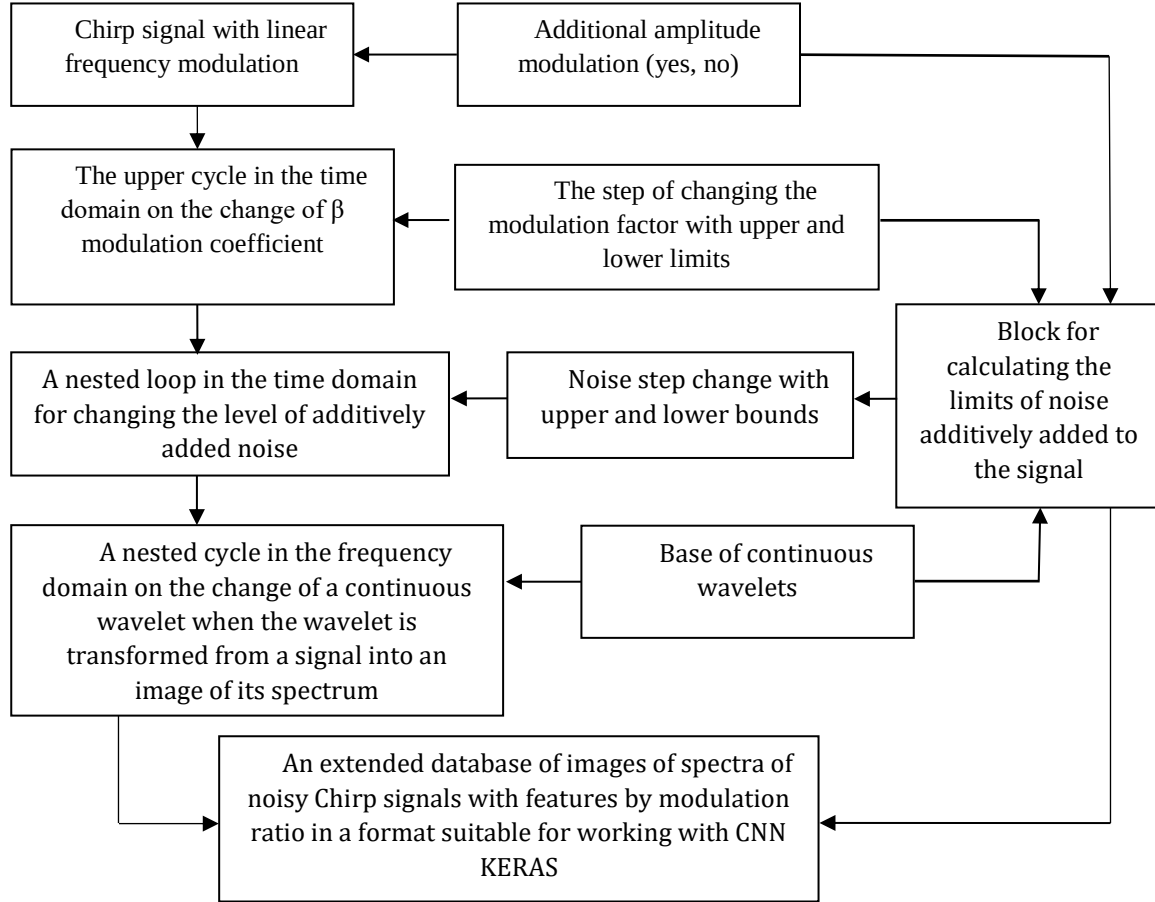


Figure 5: Block diagram of the DataSet preparation algorithm for image recognition of Chirp-type radar signal spectra taking into account the modulation coefficient

The obtained set of experimental images is formed into three subsets: training (7056 samples), validation (2520 samples) and test (504 samples) samples. The training sample is used to train the network; the validation sample in the learning process serves to select the hyper parameters of the network; a test sample is a set of images used to evaluate the performance of the network after training.

A convolutional neural network is a stack of two-dimensional convolutional layers with the activation function of the Rectified Linear Unit (ReLU), which alternate with MaxPooling 2D layers. In addition, the depth value of the hidden layers gradually increases from 32 to 64, while the size of the feature maps decreases from 248×248 to 29×29 .

The CNN architecture for the detection model is parameterized to train and predict images with a size of 250×250 pixels and three RGB channels, and the size of the output images is 640×480 pixels. The model has a binary output, so the detection task is focused on six classes of signals. Each class is configured to detect the signal spectrum belonging to the corresponding signal type as listed 128ChirpN, 256ChirpN, 512ChirpN, 128ChirpM, 256ChirpM, 512ChirpM.

The training schedule of the developed neural network model is shown in Fig. 6 - Fig. 7. The calculation was performed for 50 training epochs. The results show that the accuracy of the developed model is higher at the validation stage, this is explained by the fact that the

generalization of the model training is higher than the training data. Already at the tenth epoch, a plateau is noticeable. At the training stage, model losses decrease sharply, which is explained by a decrease in the complexity of recognizing signals when they are distorted by the addition of noise, changes in the wavelet function, and changes in frequency and amplitude modulation coefficients.

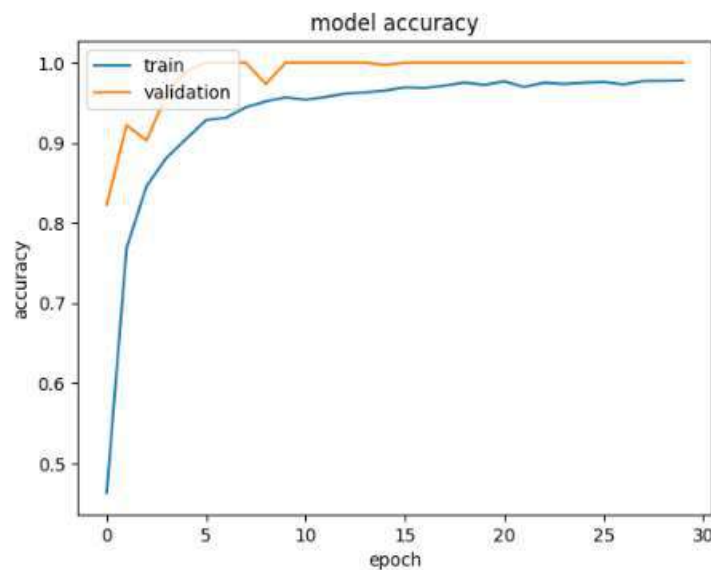


Figure 6: Accuracy of training and validation when using Wavelet Dataset

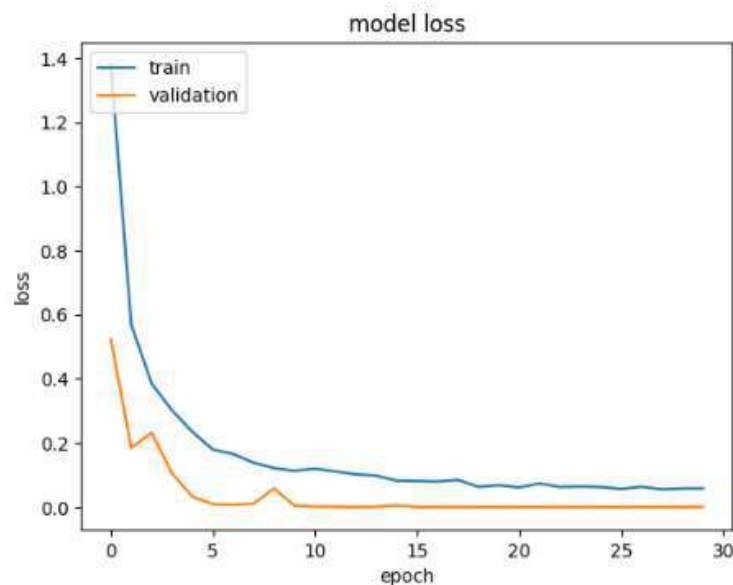


Figure 7: Training and validation losses when using Dataset wavelets

The loss reduction is associated with the limitation of the upper threshold of additively added noise, according to the proposed algorithm for its determination in the frequency domain. This excluded from the analysis the images of spectra that have already turned into solid noise and their further recognition is not possible at all.

It should be noted that it is the changes in the frequency and amplitude modulation coefficients that cause oscillation, which is particularly noticeable on the validation curve (Fig.6– Fig. 7). Accuracy increases rapidly thanks to the proposed algorithm (Fig. 5). According to the algorithm, the upper limit of additive noise, which is determined under the conditions of the ability to recognize the image of the spectrum, is created using the technology of continuous

wavelet transformation. At the same time, the limit corresponds to the moment when recognition is still possible.

For the convenience of managing the model, an interface was developed that simplifies the visualization of the results of recognizing the component spectra of a complex signal after digitization by generating an image from a digital matrix (Fig. 8).

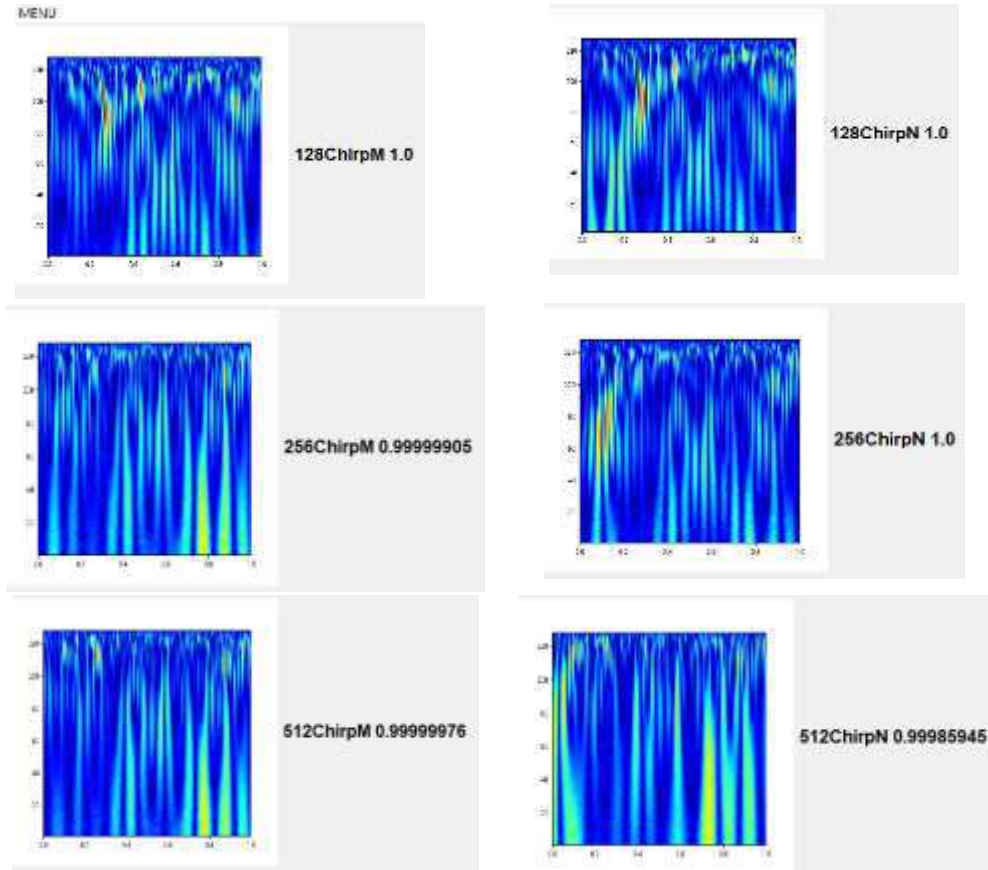


Figure 8: The operation of a simple interface for recognizing the components of the signal spectrum: 128ChirpN, 256ChirpN, 512ChirpN, 128ChirpM, 256ChirpM, 512ChirpM

Checking the accuracy on the test set shows the effectiveness of recognition:

```
Found 504 images belonging to 6 classes.  
4/4 [=====] - 4s 877ms/step - loss: 0.0023 - accuracy: 1.0000
```

To evaluate the effectiveness of the developed algorithm for preparing images of wavelet spectra for a high-precision neural network with augmentation using a continuous wavelet transformation and limiting the upper limit of noise in the frequency domain during wavelet image generation, we will perform a comparison with a known model [26].

The authors of the study [25] investigated the image recognition model of thermal imagers. The convolutional neural network model presented in [26] with an artificial increase in representativeness provides image classification with an accuracy of up to 99.37% on the test sample, but cannot be used for the classification of wavelet spectra, as it analyzes a monochrome image.

The trained neural network model using the Tensorflow library ensures the reliability of image detection and classification at the level of up to 95.7% [26]. A neural network with augmentation using primitive wavelet spectra and limiting the upper limit of noise in the frequency domain for wavelet image generation has an accuracy of up to 100% (Table 1).

The proposed spectrum recognition system takes into account signal digitization on an analog local oscillator. The presence of many signals distorts the shape of the spectrum, which is analogous to the use of different wavelets.

Table 1**Comparative analysis of model training efficiency using continuous wavelet-spectrale**

A set of experimental data (Dataset)	Classification accuracy on the training sample, %	Classification accuracy on the validation sample, %	Classification accuracy on the test sample, %
Without using augmentation and limiting the upper limit of the noise	84,3	88,1	93,1
Augmentation using Tensorflow tools, which consist of reducing, enlarging and rotating images, changing the palette	86,3	89,8	95,7
Augmentation using continuous wavelet spectra and limiting the upper limit of noise in the frequency domain for wavelet image generation	90,1	92,9	100

Discussion

In the work, when forming the data, the limitation of the range of the power change of the Gaussian noise additively added to the signal was applied. The limitation is carried out in the frequency domain by comparing the wavelet coherence of the noise and the noisy signal using the example of the most common signal with linear frequency modulation of the Chirp type.

It was established that the wavelet autocohereance of Gaussian white noise has a constant value in the entire range of noise power change. At the same time, the numerical value of autocohereance depends exclusively on the choice of wavelet. The wavelet autocohereance of the noisy signal when the noise power changes intersects with the noise autocohereance at the power value, which, in addition to the wavelet, depends on the frequency modulation coefficient. A variable Chirp waveform with additional amplitude modulation by a half-sine wave was also investigated.

The accuracy of recognition of a given type of signal in the developed model was increased due to the exclusion from the analysis of signals that cannot be recognized due to lack of distinction from noise. For each continuous wavelet, such a level of non-stationarity is determined, at which a noisy signal can be recognized. This allows you to expand the database, implement augmentation by changing the wavelet, as well as amplitude modulation of the signal.

Conclusions

An effective algorithm for image recognition of continuous wavelet spectra of noisy Chirp-type radar signals under conditions of change in the coefficient of linear frequency modulation has been developed. Formation of an experimental data set for training a convolutional neural network is carried out by generating images of signal spectra with the addition of additive noise to the Chirp signal. The frequency modulation coefficient changes regardless of the presence of amplitude modulation by a half-sine wave

The developed neural network model for image recognition of continuous wavelet spectra using the Chirp signal as an example has an accuracy of up to 100% (the best analog result is up to 95.7%).

The prospect of research is a detailed study of the potential of various wavelets or their influence on signal recognition.

References

- [1] M. V. Bugayov, S. P. Samoilyk. Extending the limits of unambiguous range and radial velocity measurement by using bundles of multicomponent signals, Problems of creation, testing, application and operation of complex information systems. 18, 91 (2020) doi:10.52013/2658-7556-64-2-7
- [2] T. T. Mar, S. S. Y. Mon, Pulse compression method for radar signal processing, International Journal of Science and Engineering Applications. 3 (2014) 31-35.
- [3] K. Savci, A. G. Stove, F. De Palo, A. Y. Erdogan, G. Galati, K. A. Lukin, C. Wasserzier, Noise Radar—Overview and Recent Developments, IEEE Aerospace and Electronic Systems Magazine. 35.9 (2020) 8-20, doi: 10.1109/MAES.2020.2990591.
- [4] K. Siddiq, M. K. Hobden, S. R. Pennock, R. J. Watson, Phase noise in FMCW radar systems, IEEE Transactions on Aerospace and Electronic Systems. 55(1) (2018) 70-81.
- [5] F. D. Palo, G. Galati, G. Pavan, C. Wasserzier, K. Savci, Introduction to noise radar and its waveforms, Sensors. 20(18) (2020) 5187.
- [6] G. Galati, G. Pavan, F. De Palo, Chirp Signals and Noisy Waveforms for Solid-State Surveillance Radars, Aerospace. 4(1) (2017) 15; doi: 10.3390/aerospace4010015.
- [7] F. D. Palo, G. Galati, G. Pavan, C. Wasserzier, Signal design and processing for noise radar, EURASIP J. Adv. Signal Process. 52 (2022). doi: 10.1186/s13634-022-00884-1
- [8] E. Antolinos, J. Grajal, Comprehensive Comparison of Continuous-Wave and Linear-Frequency-Modulated Continuous-Wave Radars for Short-Range Vital Sign Monitoring, IEEE transactions on biomedical circuits and systems. 17.2 (2023) 229-245.
- [9] M. Walencykowska, A. Kawalec, K. Krenc, An Application of Analytic Wavelet Transform and Convolutional Neural Network for Radar Intrapulse Modulation Recognition, Sensors. 23.4 (2023) 1986.
- [10] H. Li, J. Zhao, Analysis of a combined waveform of linear frequency modulation and phase coded modulation, In Proceedings of the 2016 11th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Guilin, China, 2016, 539-541.
- [11] A. A. Semenoglou, E. Spiliotis, V. Assimakopoulos, Image-based time series forecasting: A deep convolutional neural network approach, Neural Networks, 157, 2023, pp.39-53.
- [12] K. Bhangale, M. Kothandaraman, Speech emotion recognition based on multiple acoustic features and deep convolutional neural network, Electronics. 12(4) (2023) 839.
- [13] A. T. Mahmoud, W. A. Awad, G. Behery, M. Abouhawwash, M. Masud, H. Aljuaid, A. I. Ebada, An Automatic Deep Neural Network Model for Fingerprint Classification, Intelligent Automation & Soft Computing. 36(2) (2023) 2007-2023.
- [14] J. Vatter, R. Mayer, H. A. Jacobsen, The evolution of distributed systems for graph neural networks and their origin in graph processing and deep learning: A survey, ACM Computing Surveys. 56(1) (2023) 1-37.
- [15] Q. Zhang, J. Xiao, C. Tian, J. Chun-Wei Lin, S. Zhang, A robust deformed convolutional neural network (CNN) for image denoising, CAAI Transactions on Intelligence Technology. 8(2) (2023) 331-342.
- [16] S. A. Singh, K. A. Desai, Automated surface defect detection framework using machine vision and convolutional neural networks, Journal of Intelligent Manufacturing. 34(4) (2023) 1995-2011.
- [17] Yu K. Taranenko, O. Yu. Oliynyk, Optimization of the packet wavelet filtering algorithm of signals, Cybernetics and system analysis. 60 (2024) 163-174.
- [18] Yu K. Taranenko, V. V. Lopatin, O. Yu. Oliynyk, Wavelet filtering by using nonthreshold method and example of model Doppler function, Radioelectronics and Communications Systems. 64 (2021) 380-389.
- [19] D. Lewy, J. Mańdziuk, An overview of mixing augmentation methods and augmentation strategies, Artificial Intelligence Review. 56.3 (2023), 2111-2169.

- [20] O. Oliinyk, Yu. Taranenko, V. Lopatin, Analysis of Discrete Wavelet Spectra of Broadband Signals, CMIS 2023. pp. 188-198.
- [21] D. Onufrienko, Y. Taranenko, Filtering and Compression of Signals by the Method of Discrete Wavelet Decomposition into One-Dimensional, Cybernetics and Systems Analysis. (2023) 1-8.
- [22] C. Termritthikun, Y. Jamtsho, P. Muneesawang, J. Zhao, I. Lee, Evolutionary neural architecture search based on efficient CNN models population for image classification, Multimedia Tools and Applications, 82(16) (2023) 23917-23943.
- [23] F. J. Joseph, S. Nonsiri, A. Monsakul, Keras and TensorFlow: A hands-on experience. Advanced deep learning for engineers and scientists: A practical approach. 2021, 85-111.
- [24] Z. Yang, R. O. Sinnott, J. Bailey, Q. Ke, A survey of automated data augmentation algorithms for deep learning-based image classification tasks, Knowledge and Information Systems. 65(7) (2023) 2805-2861.
- [25] D. Haba, Data Augmentation with Python: Enhance deep learning accuracy with data augmentation methods for image, text, audio, and tabular data, Packt Publishing Ltd, 2023.
- [26] I. O. Skladchikov, Automated analysis of security thermal imager data based on deep learning. XIII All-Ukrainian scientific and practical conference of students, postgraduates and young scientists "Looking into the future of instrument building, May 13-14 2020, KPI named after Igor Sikorskyi, Kyiv, Ukraine, 2020, pp. 315-318.
<https://ela.kpi.ua/server/api/core/bitstreams/1b62d261-da48-4f95-a9a2-26b4c55e2121/content>

Application of a Ten-Variate Prediction Ellipsoid for Normalized Data and Machine Learning Algorithms for Face Recognition

Sergiy Prykhodko^{1,2} and Artem Trukhov¹

¹ Admiral Makarov National University of Shipbuilding, Heroes of Ukraine Ave., 9, Mykolaiv, 54007, Ukraine

² Odesa Polytechnic National University, Shevchenko Ave., 1, Odesa, 65044, Ukraine

Abstract

Face recognition plays a pivotal role in enhancing security, improving efficiency, and enabling innovative applications across diverse industries, making it an indispensable tool in today's world. This study presents a comparative investigation between prediction ellipsoid and machine learning algorithms, such as a one-class support vector machine, isolation forest, and an autoencoder based on a neural network, in the context of face recognition. The construction of a prediction ellipsoid is based on the assumption of a multivariate normal distribution of the data, however, in real-world scenarios, the data often deviates from this assumption and may exhibit a non-Gaussian distribution. To mitigate this, the dataset underwent normalization via the multivariate Box-Cox transformation, facilitating the development of a decision rule based on a ten-variate prediction ellipsoid for the normalized data. This approach yielded notable enhancements in accuracy and method robustness. All the developed models have shown good efficiency in their respective performances. However, the application of the multivariate Box-Cox normalizing transformation significantly enhanced the performance of the prediction ellipsoid. As a result, the ten-variable prediction ellipsoid for normalized data has demonstrated superior efficiency compared to the other models. The study also highlights the critical role of feature dimensionality in face recognition, emphasizing the necessity of expanding the number of features and utilizing more complex models to achieve optimal performance.

Keywords

Face recognition, multivariate normal distribution, Box-Cox transformation, machine learning

1. Introduction

Face recognition, a quintessential task in computer vision, holds significant importance in various fields, including security, surveillance, human-computer interaction, and biometrics. It entails identifying or verifying individuals by analyzing and comparing facial features captured through images or videos.

The face recognition task contains several crucial steps, each essential for ensuring the system's accuracy, robustness, and reliability [1]. Firstly, image preprocessing enhances the quality and consistency of facial images for subsequent analysis. Next, face detection locates and recognizes faces within an image, laying the foundation for recognition procedures. This step employs methods like the Viola-Jones, MTCNN, or computer vision libraries such as DLib. Following detection, feature extraction characterizes each face uniquely. This involves capturing geometric characteristics like distances between facial landmarks, e.g., eyes, nose, and mouth, generating a distinctive feature vector. The final step involves the application of decision rules to determine whether the feature vector belongs to the representative of the target class.

Traditionally, face recognition systems have been approached through classification methods, where the goal is to classify an input face image into one of several predefined classes. However, these methods often face challenges in scenarios where the available data for training does not adequately represent the entire population, leading to poor performance on novel faces. To address this limitation, the one-class classification approach has emerged. This approach, closely related to anomaly detection [2], focuses on learning representations of a single class, typically

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ sergiy.prykhodko@nuos.edu.ua (S. Prykhodko); artem.trukhov@gmail.com (A. Trukhov)

ORCID iD 0000-0002-2325-018X (S. Prykhodko); 0000-0002-7160-8609 (A. Trukhov)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

the target, without explicit knowledge of the negative classes, representatives of these classes are considered anomalies.

In the domain of face recognition, one-class classification stands as a pivotal methodology, enabling the identification of target classes amidst varying data distributions. Prediction ellipsoid and machine learning methods emerge as frontrunners among the popular approaches in this realm [3-5]. Given their significance in the field, this study aims to compare these diverse approaches in the context of face recognition tasks, examining their performance, robustness, and applicability.

2. Literature review

The field of face recognition has seen significant advancements in recent years, with various techniques developed such as linear discriminate analysis, support vector machine, and principal component analysis combined with face recognition by using K-Nearest-Neighbor and neural networks, including convolutional neural networks [6]. Traditional methods for face recognition typically rely on supervised learning approaches, where the system is trained on a dataset containing samples from each class or individual to be recognized. However, these methods require a predefined set of classes or identifiers, which makes them less suitable for scenarios where comprehensive training data is not available or when the study population is highly diverse. Another disadvantage is that such systems are poorly suited for the development of an identification model, where it is necessary to verify whether a specific individual belongs to a particular class. This makes one-class classification particularly well-suited for scenarios where only data from the target class is available or when the negative class is poorly defined. That is why for abnormal image detection, abnormal event detection, active authentication, and antispoofing one-class classification methods are extensively used [7].

Among the common approaches to one-class classification, prediction ellipsoid [3, 4] and machine learning methods [5] have emerged as popular choices. Notable examples include one-class support vector machine (OCSVM) [8, 9], isolation forest (IF) [10], and autoencoder (AE) [11, 12]. OCSVM is a support vector machine trained on a target class of data, learning a decision boundary that separates the data points from the origin in the feature space while maximizing the margin. IF is an ensemble learning method that isolates anomalies by randomly selecting features and partitioning data points until anomalies are isolated into small partitions, requiring fewer splits compared to target points. AE is a neural network designed to learn efficient representations of data by encoding the input into a lower-dimensional space—encoder, and then reconstructing the original input from this space—decoder, anomalies are discerned by evaluating the reconstruction error, with heightened errors indicating potential outliers.

The application of a prediction ellipsoid relies on the assumption of a multivariate normal distribution within the data, which may not always hold, this discrepancy necessitates the utilization of normalization transformations [13]. These transformations aim to align the data more closely with the assumptions of a multivariate normal distribution, thereby enhancing the accuracy and robustness of the analysis. Normalization techniques encompass various approaches, including univariate transformations such as logarithmic or Box-Cox transformations, which operate independently on each feature. Conversely, multivariate transformations like the multivariate Box-Cox transformation consider inter-feature relationships, resulting in more effective normalization of the data [14].

Our study primarily investigates the method based on prediction ellipsoid and machine learning algorithms, such as OCSVM, IF, and AE, which are commonly utilized and offer distinct approaches to one-class classification. Specifically, OCSVM delineates a discerning boundary around normal instances, IF isolates anomalies through iterative partitioning of the feature space, AE employs a neural network approach, and the prediction ellipsoid separates normal instances from potential outliers based on their position in the feature space.

In the realm of face recognition, where accuracy and efficiency are paramount, it's imperative to evaluate the efficacy of different approaches. While prediction ellipsoid offers interpretability and computational efficiency, they may encounter limitations with non-Gaussian data distributions. In contrast, machine learning methods such as OCSVM, IF, or AE present alternative methodologies, each with unique strengths and challenges.

3. Face detection and feature extraction

In the domain of computer vision, a variety of techniques are employed for detecting faces in images, each characterized by distinct strengths and functionalities. Notable among these methods are the Viola-Jones, Multi-Task Cascaded Convolutional Neural Network (MTCNN), and DLib toolkit [15].

The Viola-Jones represents a classic yet effective approach to face detection, renowned for its simplicity and efficiency. Operating by scanning images at multiple scales, it utilizes a predefined set of features to pinpoint regions of interest potentially containing faces. It is implemented in the Python OpenCV library.

MTCNN, on the other hand, stands out as a deep learning-based face detection tool prized for its remarkable accuracy and resilience. Comprising three stages—face detection, bounding box regression, and facial landmark localization—MTCNN leverages a cascade of convolutional neural networks to precisely identify faces in images while concurrently estimating facial landmarks. This method is particularly advantageous for applications necessitating precise localization of facial features.

DLib emerges as a comprehensive C++ toolkit with formidable capabilities in facial detection and recognition. Leveraging a Histogram of Oriented Gradients and Linear Support Vector Machine face detector, DLib excels in detecting faces and estimating 68 facial points. These points facilitate accurate detection and localization of facial features, enabling seamless implementation of face alignment, merging, and other transformations.

While all face detection methods exhibit commendable performance in high-quality images, they display variations in performance under different conditions. MTCNN and DLib demonstrate comparable accuracy, with MTCNN exhibiting superior performance in recognizing faces in small images and low-light conditions. In contrast, the Viola-Jones method may encounter challenges with distortions. Performance evaluations conducted using a standardized test dataset reveal that Viola-Jones operates at the highest speed, processing 11.25 frames per second (fps), followed closely by DLib at 9.41 fps, and MTCNN at 4.92 fps.

Given its good speed and proficiency in extracting a substantial number of key points, DLib emerges as the preferred choice for face detection and feature extraction tasks, particularly in scenarios where real-time processing or high throughput is essential.

A program was created using the DLib computer vision library in Python to generate feature vectors. Once a face is detected in the input image, the program begins several image-processing tasks. These tasks include cropping and aligning the face to ensure that the eyes are aligned at the same level. This preprocessing step helps reduce distortions caused by the face's orientation in the input image. In the last phase, the program extracts a series of attributes from the aligned image. Each attribute denotes the pixel distance between facial landmarks identified by the DLib library. It used 17 key landmarks across the face (fig. 1).

Utilizing the pixel distances between these landmarks, a feature vector composed of 10 distinct attributes was meticulously crafted. Symmetrical distances were thoughtfully averaged to derive the following features:

1. The eyes-to-nose midpoint distance indicates the facial symmetry and proportionality between the eyes and the nose.
2. The eyes-to-mouth center distance reflects the vertical alignment and proportionality between the eyes and the mouth.
3. The eyes-to-eyebrows center distance offers insights into the spatial relationship between the eyes and the eyebrows.
4. The eyebrows-to-nose bridge distance delineates the vertical dimension and proportionality between the eyebrows and the nose bridge.
5. The eye's corners-to-nose bridge distance provides insights into the lateral dimension and proportionality between the eye's corners and the nose bridge.
6. The eyebrows distance.
7. The nose-to-mouth midpoint distance offers insights into the vertical alignment between the nose and the mouth.
8. The corners of the mouth distance, reflect the width of the mouth.
9. The edges of the nose distance, delineate the width of the nose.

10. The mouth-to-chin distance provides insights into the vertical dimension between the mouth and the chin.

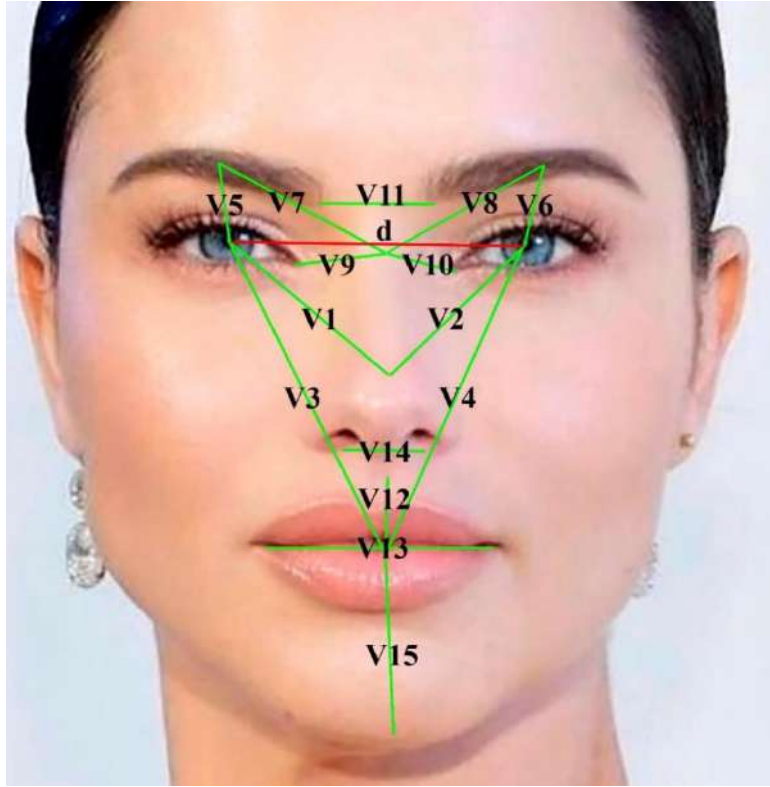


Figure 1: Distances between facial landmarks used to generate the feature vector

To adapt to variations in facial positioning within the image and varying distances from the camera, a normalization procedure is implemented. This involves dividing each feature by the distance between the eyes. By scaling each feature relative to the interocular distance, the normalization process ensures that facial characteristics remain consistent across different images and distances, enhancing the robustness and accuracy of subsequent analysis and recognition tasks.

For the research, the well-known dataset Pins Face Recognition was chosen, which is used in building decision rules for facial recognition. To ensure the reliability and accuracy of the study, the dataset underwent filtering, followed by additional downloading of photos from open sources and expansion of the dataset. In total, 400 high-quality photos were obtained, 200 for each of the two faces. Consequently, 400 feature vectors were acquired, each corresponding to a single photo, comprising 10 elements. Out of these, 100 vectors of a single person were used to construct a prediction ellipsoid and train the machine learning models for identifying the individual, while the remaining 300 photos were allocated as the test set.

4. Materials and methods

4.1. Prediction ellipsoid for normalized data

In the construction of the prediction ellipsoid, each observation's squared Mahalanobis distance is calculated and compared to critical values derived from the Chi-square distribution [16].

$$(X - \bar{X})^T S_X^{-1} (X - \bar{X}) \leq \chi_{k, \alpha}^2, \quad (1)$$

where α is the confidence level, k is the number of dimensions in the data,

$$S_X = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})(X_i - \bar{X})^T. \quad (2)$$

Figure 2 illustrates the example of a 3-variate prediction ellipsoid. In our study, we use a 10-variate prediction ellipsoid. Objects within its boundaries are classified as the target class, while those outside are identified as anomalies.

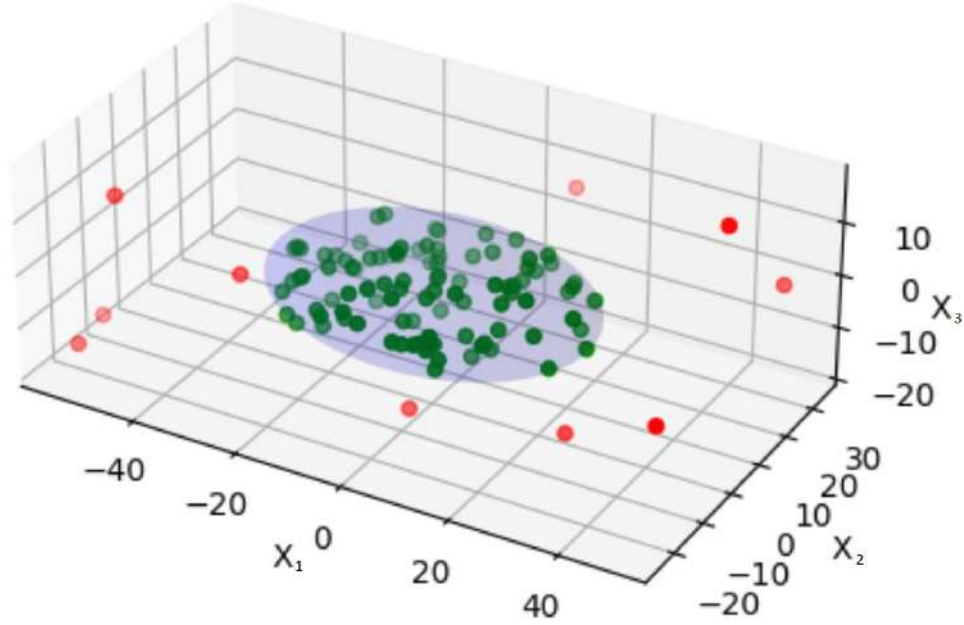


Figure 2: Example of a 3-variate prediction ellipsoid

Since the prediction ellipsoid relies on the assumption of data normality, the Mardia test was employed to evaluate the departure of the multivariate data distribution from normality.

$$\beta_1 = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \{(X_i - \bar{X})^T S_X^{-1} (X_j - \bar{X})\}^3; \quad (3)$$

$$\beta_2 = \frac{1}{N} \sum_{j=1}^N \{(X_j - \bar{X})^T S_X^{-1} (X_j - \bar{X})\}^2. \quad (4)$$

It is a statistical test used to assess whether a dataset follows a multivariate normal distribution. It measures the multivariate skewness β_1 (3) and multivariate kurtosis β_2 (4) of the data to determine how closely it resembles a normal distribution in multiple dimensions. This test is particularly useful in various fields such as statistics, econometrics, and multivariate analysis, where the assumption of multivariate normality is important for accurate interpretation and analysis of data.

Based on the results of the Mardia test, it is evident that the multivariate distribution of the training sample deviates from Gaussian. This is indicated by the test statistic for multivariate skewness $N\beta_1/6$, which measures 289.20, surpassing the quantile of the Chi-Square distribution set at 277.77 for 220 degrees of freedom and a significance level of 0.005. Conversely, the test statistic for multivariate kurtosis β_2 with a value of 122.35 does not exceed the Gaussian distribution quantile, which stands at 127.97 for a mean of 120, a variance of 9.6, and a significance level of 0.005. Consequently, it is imperative to apply a normalizing transformation like the Box-Cox transformation on a non-Gaussian random vector $X = \{X_1, X_2, \dots, X_{10}\}^T$, to convert it into a Gaussian random vector $Z = \{Z_1, Z_2, \dots, Z_{10}\}^T$.

The Box-Cox transformation is a statistical technique used to stabilize variance and make data conform more closely to the assumptions of normality. It is typically applied to univariate data, but there is also a multivariate extension known as the multivariate Box-Cox transformation.

$$Z_j = x(\lambda_j) = \begin{cases} (X_j^{\lambda_j} - 1)/\lambda_j, & \lambda_j \neq 0; \\ \ln(X_j), & \lambda_j = 0. \end{cases} \quad (5)$$

In the multivariate context, the BCT aims to normalize data across multiple variables simultaneously. It is designed to address situations where there are multiple correlated variables

and normalizing each variable individually may not be sufficient to achieve multivariate normality. The multivariate BCT involves estimating transformation parameters for each variable in the sample, similar to the univariate case. However, it also considers the interrelationships between variables to ensure that the transformation maintains the structure of the multivariate distribution. One of the methods to determine parameters for each variable in the dataset involves maximizing the log-likelihood of the transformed data:

$$l(X, \theta) = \sum_{j=1}^k (\lambda_j - 1) \sum_{i=1}^N \ln(x_{ji}) - \frac{N}{2} \ln[\det(S_Z)], \quad (6)$$

where S_Z is calculated as (2) using the normalized sample Z instead of X .

Following the completion of the task utilizing the maximum likelihood method of the logarithmic function (6), the ensuing parameter estimates were acquired: $\hat{\lambda}_1 = -0.5799$, $\hat{\lambda}_2 = -0.9732$, $\hat{\lambda}_3 = 0.4871$, $\hat{\lambda}_4 = 2.888$, $\hat{\lambda}_5 = 4.0714$, $\hat{\lambda}_6 = -0.231$, $\hat{\lambda}_7 = 0.087$, $\hat{\lambda}_8 = -1.2973$, $\hat{\lambda}_9 = -1.1866$, $\hat{\lambda}_{10} = 1.3321$.

As a result of the BCT application with components (5), a sample was obtained, featuring the resulting vector of means: $\bar{Z} = \{0.52627; 0.13911; -0.91083; 0.25654; -0.24379; 1.13224; -1.14383; -0.26329; 2.03304; -0.34494\}$.

The covariance matrix of the normalized sample S_Z is:

$$\begin{bmatrix} 0.0^244 & 0.0^213 & -0.0^22 & -0.0^46 & -0.0^56 & -0.0^38 & -0.0^23 & 0.0^470 & 0.0^262 & -0.0^22 \\ 0.0^213 & 0.0^211 & -0.0^42 & 0.0^477 & -0.0^53 & 0.0^317 & 0.0^211 & -0.0^44 & 0.0^228 & 0.0^313 \\ -0.0^22 & -0.0^42 & 0.0^237 & 0.0^336 & 0.0^651 & 0.0^212 & 0.0^238 & -0.0^34 & -0.0^22 & 0.0^219 \\ -0.0^46 & 0.0^477 & 0.0^336 & 0.0^466 & -0.0^63 & 0.0^323 & 0.0^338 & -0.0^32 & -0.0^33 & 0.0^312 \\ -0.0^56 & 0.0^528 & 0.0^651 & 0.0^627 & 0.0^748 & 0.0^516 & 0.0^521 & -0.0^64 & -0.0^42 & 0.0^517 \\ -0.0^38 & 0.0^317 & 0.0^212 & 0.0^323 & 0.0^517 & 0.0^285 & 0.0^227 & 0.0^312 & -0.0^21 & 0.0^489 \\ -0.0^23 & 0.0^211 & 0.0^238 & 0.0^338 & 0.0^521 & 0.0^227 & 0.0105 & -0.0^39 & -0.0^23 & 0.0^238 \\ 0.0^470 & -0.0^44 & -0.0^33 & -0.0^32 & -0.0^64 & 0.0^312 & -0.0^39 & 0.0^271 & 0.0139 & 0.0^217 \\ 0.0^262 & 0.0^228 & -0.0^22 & -0.0^33 & -0.0^42 & -0.0^21 & -0.0^23 & 0.0139 & 0.0857 & 0.0^212 \\ -0.0^22 & 0.0^313 & 0.0^219 & 0.0^312 & 0.0^517 & 0.0^489 & 0.0^238 & 0.0^217 & 0.0^212 & 0.0^231 \end{bmatrix}$$

The normalized sample shows no departure from the multivariate normal distribution. This is supported by the test statistics: the multivariate skewness test statistic $N\beta_1/6$, measuring at 265.59 does not exceed the critical value of 277.77, and the multivariate kurtosis test statistic β_2 , recorded at 121.52, is lower than the critical value of 127.97.

After applying the normalization transformations, a ten-variate prediction ellipsoid is constructed based on equation (1):

$$(Z - \bar{Z})^T S_Z^{-1} (Z - \bar{Z}) \leq \chi_{10, 0.005}^2. \quad (7)$$

The quantile value of the Chi-square distribution is 25.19 for 10 degrees of freedom and a significance level of 0.005.

4.2. Machine learning algorithms

Several popular techniques in machine learning serve as effective tools for outlier detection and can be used for face recognition, including a one-class support vector machine, isolation forest, and autoencoder. These methods are particularly notable for their application in unsupervised learning and they can be trained only on data from the target class, studying the features of the provided data.

4.2.1. One-class support vector machine

OCSVM constructs a decision function or boundary that effectively separates the target data from the rest of the feature space. This boundary is established by finding a hyperplane with a maximum margin and optimizing the distance between the hyperplane and the origin in the high-dimensional feature space.

In an OCSVM, an implicit transformation function $\varphi(\cdot)$ is employed, it is a non-linear projection that is evaluated through a kernel function, which serves as a mapping from the original feature space to a potentially higher-dimensional feature space: $k(x, y) = \varphi(x) \cdot \varphi(y)$ [17]. Commonly used kernel functions include: the linear kernel, which computes dot products in the original

feature space and is suitable for linearly separable data; the polynomial kernel captures non-linear relationships between data points by raising dot products to specified powers, allowing for the modeling of complex decision boundaries; the radial basis function kernel, employing a Gaussian function, is highly effective in capturing intricate relationships between data points, particularly in scenarios where data is not linearly separable; lastly, the sigmoid kernel, based on the hyperbolic tangent function, adeptly captures non-linear patterns in the data, making it useful for handling complex relationships between features and classes.

Subsequently, the algorithm learns a decision boundary aimed at segregating the majority of the data from the origin, it is defined as [18]:

$$g(x) = \omega^T \varphi(x) - \rho,$$

where ω the normal vector of the hyperplane, ρ is the bias.

Formulating OCSVM as a quadratic optimization problem, the main aim is to minimize the weight vector while simultaneously maximizing the margin, subject to specific constraints:

$$\min_{\omega, \xi, \rho} \frac{\|\omega\|^2}{2} - \rho + \frac{1}{vN} \sum_{i=1}^N \xi_i,$$

$$\text{subject to: } \omega^T \varphi(x_i) \geq \rho - \xi_i, \xi_i \geq 0,$$

where ξ_i is slack variables, it is utilized to model separation errors; $v \in (0,1]$ is the regularization parameter, it characterizes the solution by setting an upper bound on the fraction of outliers, such as the training examples regarded as out-of-class, and simultaneously establishes a lower bound on the number of training examples utilized as support vectors [19].

The optimization problem is typically solved in its dual form, resulting in a decision function that can classify new data points as either belonging to the target class or representing anomalies. This function yields a positive value for the target and a negative value otherwise:

$$f(x) = \text{sgn}(g(x)).$$

In Python, implementing OCSVM typically involves creating a `OneClassSVM` object from the `SVM` module in `sci-kit-learn`. This allows users to adjust essential parameters such as the choice of kernel function, regularization parameter v , and kernel-specific parameters. The model is configured with a v parameter of 0.15, which determines the fraction of training errors and the upper bound on the fraction of training set outliers. For the kernel function, the radial basis function is utilized. This kernel is popular for SVMs due to its flexibility in capturing non-linear relationships between data points. Additionally, the gamma parameter is set to `auto`, which means its value is automatically calculated based on the inverse of the number of features. Gamma defines the influence range of a single training example, with low values meaning far and high values meaning close.

4.2.2. Isolation forest

Departing from traditional approaches reliant on profiling normal data points, isolation forest takes a unique route by directly focusing on isolating anomalies. This method hinges on constructing isolation trees, binary trees where internal nodes represent features and split values, while leaf nodes represent individual data points [20].

The process of constructing isolation trees initiates with the random selection of a feature and split value within its range. This randomness recurs until each data point is secluded in its own leaf node, or until a predetermined maximum tree depth is attained. The method's ability to efficiently isolate anomalies sans prior knowledge of the dataset's distribution stems from this stochastic feature and split value selection. Anomalies, expected to be inherently easier to isolate than normal data points, typically reside in sparser regions of the feature space. Consequently, they necessitate fewer splits along paths from the root to leaf nodes for isolation. Hence, the average path length from the root to leaf nodes for each data point is computed across all trees within the forest [21].

Subsequently, an anomaly score is computed for each data point based on its average path length [22]:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}},$$

where $E(h(x)) = \frac{\sum_{i=1}^t h_i(x)}{t}$ is the average length of x over t isolation trees; $c(n)$ is the average length of unsuccessful search in a binary tree:

$$c(n) = 2H(n-1) - (2(n-1)/n),$$

where $H(i) = \ln(i) + \gamma$, γ is Euler's constant.

Data points with shorter path lengths, indicative of proximity to the root, are deemed more likely to be anomalies, while those with longer paths are considered more typical. A threshold is established to classify data points as anomalies or normal points based on their anomaly scores, with points above the threshold classified as anomalies and those below as normal.

Isolation forest harnesses an ensemble of these isolation trees, where each tree operates independently, contributing to the final anomaly score for a given data point. This ensemble approach enhances the algorithm's robustness and accuracy, rendering it more adept at handling noise and data variability.

The IsolationForest algorithm is implemented in the widely-used Python machine learning library sci-kit-learn and offers various adjustable parameters for optimizing its performance. Among these is the contamination parameter, essential for determining the threshold in the decision function, which distinguishes new data points as either target or anomalous [23]. After conducting experiments, a value of 0.08 was selected for the contamination parameter as it strikes the optimal balance between detecting true anomalies and minimizing false alarms.

Other significant parameters of the IsolationForest model include: `n_estimators`, specifying the number of decision trees within the forest, set to 100; `max_samples`, indicating the maximum number of samples used for training each tree, set to 256; and `max_features`, determining the maximum number of features used for splitting each node, set to 1.0, indicating that all features are used. These parameter settings were chosen based on their widespread usage and their ability to strike an optimal balance between model complexity and computational efficiency.

To classify a sample as either target or anomalous, the anomaly score is compared to a threshold value. These scores can range from positive to negative, with negative values indicating samples more likely to be the target and positive values indicating samples more likely to be anomalous. The selection of the threshold value depends on the specific application and the characteristics of the data. In our case, a threshold value of 0 yielded the best results.

4.2.3. Autoencoder

An autoencoder is a form of artificial neural network employed for learning efficient data representations, dimensionality reduction, and detecting anomalies. It is an unsupervised learning method composed of two primary components: an encoder and a decoder. The main objective of an autoencoder is to learn a condensed and meaningful representation of the input data, referred to as the latent space or bottleneck, by reconstructing the input data with minimal loss [24].

The encoder's role is to map the input data to the latent space, compressing the information into a lower-dimensional representation. It is generally a neural network with multiple layers, where each layer applies a non-linear transformation to the input data. The encoder's output, the latent space, captures the most significant features and patterns in the input data.

In contrast, the decoder is responsible for reconstructing the input data from the latent space representation. It is also a neural network with multiple layers, but its architecture is typically the inverse of the encoder's architecture. The decoder takes the latent space representation as input and tries to recreate the original input data by applying a series of non-linear transformations through its layers.

During training, the autoencoder learns to minimize the reconstruction error, which is the discrepancy between the original input data and the reconstructed data. This is usually accomplished by optimizing a loss function, such as mean squared error or binary cross-entropy, using gradient-based optimization techniques like backpropagation.

In the context of recognition, the autoencoder is trained using target class instances, enabling it to learn the patterns and structure of normal data. When presented with new data, the autoencoder will reconstruct the input with a low reconstruction error if the input is the target. However, if the input is an anomaly, the reconstruction error will be higher, as the autoencoder

has not been trained to reconstruct such instances effectively [25]. By setting an appropriate threshold on the reconstruction error, it is possible to detect and distinguish anomalies from target instances.

When considering the development of autoencoder models, various machine learning frameworks present distinct advantages. TensorFlow and PyTorch emerge as prominent contenders in this domain. TensorFlow, originating from Google, offers a rich ecosystem of tools and libraries, rendering it a favored option among both researchers and practitioners. Its utilization of a static computation graph facilitates efficient optimization and distributed computing, making it particularly suited for handling large-scale neural network models. Conversely, PyTorch, hailing from Facebook's AI Research lab, emphasizes flexibility and user-friendliness. With its dynamic computation graph and intuitive API, PyTorch becomes the preferred choice for swift prototyping and experimentation [26].

In the creation of autoencoder models, TensorFlow and Keras were selected due to their collective strengths in flexibility, scalability, and usability. Keras, functioning as a high-level neural networks API layered atop TensorFlow, streamlines the process of constructing and training autoencoder models. Meanwhile, TensorFlow furnishes the foundational infrastructure for proficient computation, ensuring optimal performance throughout the training and inference phases.

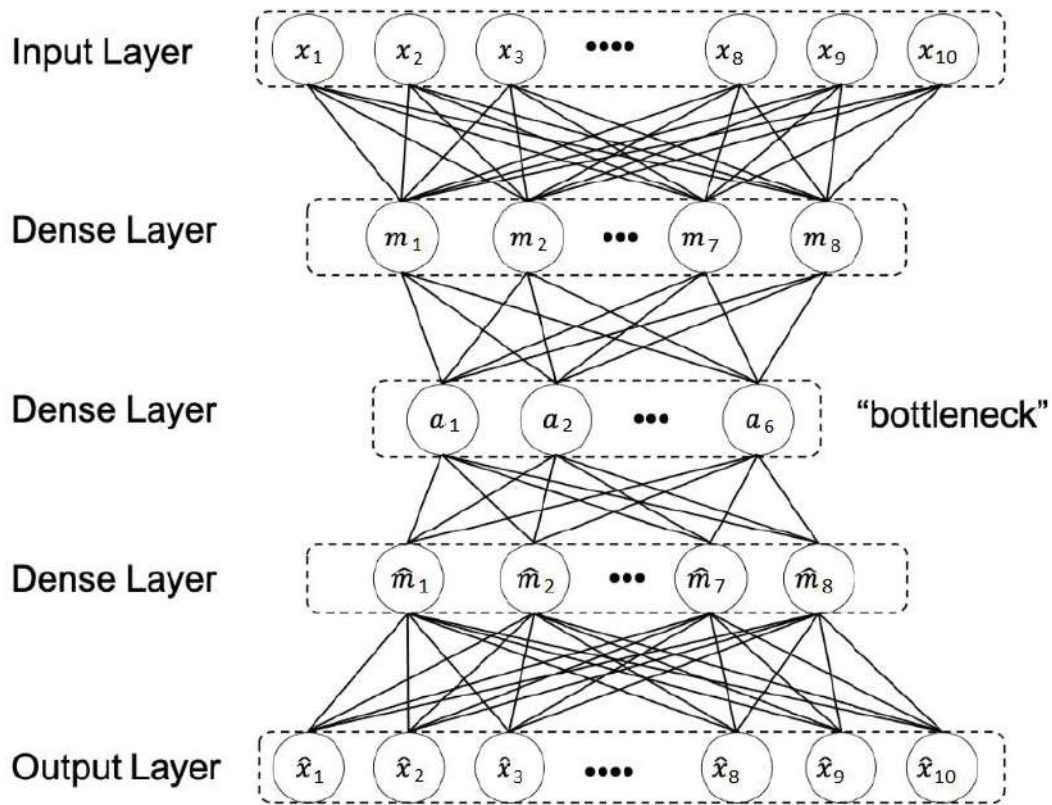


Figure 3: Autoencoder structure

Before feeding the data into the neural network, min-max normalization is applied to each feature individually. This normalization method scales each feature to a range $[0, 1]$ [27]. By applying min-max normalization, all features are brought to the same scale and are constrained within the desired range, promoting stable and efficient learning processes. The autoencoder model is designed with an input layer that receives data in the shape of a 10-variate representation. The model consists of a series of fully connected layers for both encoding and decoding operations. The encoding phase compresses the input data into a lower-dimensional representation, gradually reducing the dimensionality from 10 to 8 and then further down to 6, forming a bottleneck in the network architecture [28]. This bottleneck layer restricts the model's capacity to capture the most salient features of the input data, forcing it to learn a compact representation that captures essential information. Each encoding layer utilizes rectified linear

unit (ReLU) activation functions, which introduce non-linearity to the model and facilitate the extraction of complex features from the input data. Following the encoding layers, the decoding phase reverses the process, reconstructing the original input dimensions. The decoded layers expand the dimensionality back to 8 and finally, to the original 10 dimensions. Similar to the encoding layers, ReLU activation functions are employed in the decoding layers, preserving the non-linear relationships learned during the encoding phase. The final layer of the model uses a sigmoid activation function, ensuring that the output values are within the range [0, 1]. This activation function is commonly used for binary classification tasks and reconstruction problems, providing smooth and interpretable output. The autoencoder structure (fig. 3) is similar to [29].

To train the model, the Adam optimizer is utilized with binary cross-entropy loss, a common choice for reconstruction tasks aiming to minimize the difference between the input and reconstructed data. Adam optimizer is an adaptive learning rate optimization approach that combines the advantages of both AdaGrad and RMSProp [30]. It dynamically adjusts the learning rate during training, allowing for faster convergence and improved performance. Binary cross-entropy loss, on the other hand, measures the dissimilarity between the input and the reconstructed data, particularly suitable for binary classification problems where each instance belongs to one of two classes. This configuration ensures efficient training of the autoencoder model, optimizing its ability to capture the underlying patterns in the data while minimizing reconstruction error.

The training process spans 100 epochs, with a batch size of 8 instances per batch. Shuffling the data at each epoch introduces randomness and prevents the model from memorizing the training data order, aiding in generalization.

4.3. Evaluation metrics

Various evaluation metrics are employed to assess the effectiveness of created models in recognition of the target instances or anomalies. Commonly used metrics include accuracy, precision, recall (sensitivity), specificity, and the $F1$ score [31, 32]. While accuracy provides a broad perspective on correctness, precision, recall, specificity, and the $F1$ score offers nuanced insights into the model's efficacy in detecting anomalies.

The evaluation metrics are derived from a confusion matrix, which offers four distinct measures. True Positive (TP) represents instances where the model correctly identifies an anomaly as such. False Positive (FP) instances occur when the model inaccurately labels a target instance as an anomaly. True Negative (TN) denotes instances where the model accurately identifies target instances. Lastly, False Negative (FN) instances arise when the model incorrectly labels target instances as normal.

Accuracy acts as a gauge for overall correctness, derived from the ratio of correctly classified instances to the total number of instances :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

Precision, on the other hand, quantifies the proportion of correctly identified anomalies among all instances classified as anomalies. This metric is calculated as the ratio of true positives to the sum of true positives and false positives:

$$Precision = \frac{TP}{TP + FP}.$$

Recall, also termed sensitivity, represents the proportion of actual anomalies correctly identified by the model. It serves as a pivotal indicator of the model's ability to detect all anomalies within the dataset:

$$Recall = \frac{TP}{TP + FN}.$$

Specificity provides complementary insights by measuring the proportion of true negatives and correctly predicted target instances, among all actual target instances. It highlights the model's capacity to accurately identify target instances while minimizing false alarms.

$$Specificity = \frac{TN}{TN + FP}.$$

The $F1$ score is the harmonic mean of precision and recall:

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}.$$

5. Results

Table 1 presents a comparison of recognition performance of prediction ellipsoid for non-Gaussian data (PENGD) (1), prediction ellipsoid for normalized data (PEND) (7), one-class support vector machine (OCSVM), isolation forest (IF) and autoencoder (AE).

Table 1
Comparison of models

Model	Precision	Recall	Specificity	F1 score	Accuracy
PENGD	0.9598	0.9550	0.9200	0.9574	0.9433
PEND	0.9848	0.9750	0.9700	0.9793	0.9733
OCSVM	0.8878	0.9100	0.7700	0.8988	0.8633
IF	0.9393	0.9300	0.8800	0.9347	0.9133
AE	0.9366	0.9600	0.8700	0.9481	0.9300

In general, all developed models have good efficiency. The OCSVM exhibited the lowest performance among the models evaluated, with scores generally lower across precision, recall, specificity, *F1* score, and accuracy. The IF demonstrated competitive performance across most metrics, falling between the scores of OCSVM and the other models. This indicates that IF may offer a decent compromise between simplicity and performance, making it a viable choice depending on the specific requirements of the application.

AE showed performance comparable to PENGD, with scores across various metrics similar to or slightly lower than PENGD. However, by employing normalization techniques, PEND achieved notable enhancements in performance compared to both AE and PENGD. This underscores the importance of normalizing transformation in enhancing prediction ellipsoid models for recognition, particularly in scenarios involving non-Gaussian data.

6. Discussion

As evidenced by the results, the prediction ellipsoid for normalized data consistently outperformed machine learning models such as OCSVM, IF, and AE. Compared to the prediction ellipsoid for non-Gaussian data, the application of a ten-variate Box-Cox transformation significantly improved the model. This highlights the importance of applying a normalizing transformation to data with a non-Gaussian distribution. The choice of an appropriate normalization technique is far from trivial and exerts a profound influence on the efficacy and robustness of the created model. Univariate normalization methods often fall short of capturing the intricate relationships between features, leading to suboptimal results. In contrast, multivariate normalization techniques offer a more comprehensive approach by considering the correlations and interdependencies between multiple features. By applying the ten-variate Box-Cox transformation, the prediction ellipsoid can account for correlations between features. This approach enables the model to more accurately represent the underlying data distribution, leading to enhanced robustness and efficiency.

It should be noted that a crucial aspect in the construction of a prediction ellipsoid is the definition of the confidence level. In our study, a confidence level of 0.005 was selected, aligning with its widespread usage in outlier detection tasks [33].

The main advantage of using the prediction ellipsoid for normalized data lies in its ability to enhance prediction accuracy and reliability. Normalizing the data not only facilitates the development of more accurate prediction ellipsoid but also ensures they capture the underlying patterns and relationships within the data more effectively, leading to more reliable and precise predictions. Additionally, by transforming the data to data with the Gaussian distribution, the prediction ellipsoid gains increased robustness and resilience.

The application of prediction ellipsoid for normalized data has some disadvantages. Firstly, a substantial dataset size is often required to robustly study the model's performance, typically necessitating a minimum of 100 instances to yield meaningful insights. Additionally, the process of identifying an appropriate normalizing transformation can be intricate and resource-intensive, particularly in cases where the underlying data distribution is complex or poorly understood. The last one is the necessity of selecting a confidence level, which can have a significant impact on the interpretation and reliability of the prediction ellipsoid.

It's important to acknowledge the inherent limitations of the prediction ellipsoid model. One notable constraint is its inherent design to recognize and delineate prediction regions for individual instances rather than accommodating multiple instances simultaneously. While the model excels in providing localized predictions for individual data points, its applicability to scenarios involving collective analysis or batch processing may be limited, warranting careful consideration in practical deployment scenarios.

Future research could involve exploring alternative normalizing transformations, such as the multivariate Johnson transformation. In addition, it's important to consider the impact of model complexity and feature richness on performance. While the models evaluated in this study demonstrate commendable results with the available features, there's a clear indication that employing more complex models and incorporating a broader range of features could yield even better outcomes. This emphasizes the necessity of advancing both model architecture and feature selection techniques to effectively address the intricacies present in the data.

7. Conclusions

The study examined various one-class classification methods in the context of face recognition. Various models were constructed including prediction ellipsoid for normalized data and machine learning algorithms such as OCSVM, IF, and AE. These methodologies aim to unveil anomalies within a dataset by apprehending its underlying structure, often gleaned from instances representing the target class.

The application of the ten-variate Box-Cox transformation significantly enhanced the performance of the prediction ellipsoid, which led to a 3% increase in recognition accuracy. Consequently, the model surpassed the performance of the considered machine learning methods. This underscores the significance of applying normalization techniques to non-Gaussian data. The primary advantage of utilizing the prediction ellipsoid for normalized data is its capacity to enhance prediction accuracy and reliability. Normalizing the data not only facilitates the development of more precise prediction ellipsoids but also ensures they capture the underlying patterns.

The main disadvantage of the developed model lies in the complexity and resource-intensive nature of determining the appropriate normalizing transformation. This challenge is particularly pronounced in cases where the underlying data distribution is complex or poorly understood, especially when dealing with high-dimensional data.

An essential aspect of employing the method involves determining the confidence level, with a confidence level of 0.005 utilized in the study.

Future research could explore the impact of employing alternative normalizing methods, such as the multivariate Johnson transformation, to further enhance model performance. Additionally, deeper investigations into the reasons behind the observed performance differences among models and the potential benefits of incorporating more complex models with a broader array of features could contribute to the advancement of one-class classification methodologies in face recognition tasks.

References

- [1] Y. Kortli, M. Jridi, A. Al Falou, M. Atri, Face Recognition Systems: A Survey. *Sensors*. 2020 20(2):342. doi:10.3390/s20020342.
- [2] H. Marques, L. Swersky, J. Sander, R. Campello, A. Zimek, On the evaluation of outlier detection and one-class classification: a comparative study of algorithms, model selection,

- and ensembles. *Data Min Knowl Disc* 37, 1473–1517, 2023. doi: 10.1007/s10618-023-00931-x.
- [3] V. Bezerra V, V. da Costa, S. Barbon Junior, R. Miani, B. Zarpelão, IoTDS: A One-Class Classification Approach to Detect Botnets in Internet of Things Devices. *Sensors*. 2019; 19(14):3188. doi:10.3390/s19143188.
 - [4] S. Kim, D. Park, J. Jung, Evaluation of One-Class Classifiers for Fault Detection: Mahalanobis Classifiers and the Mahalanobis-Taguchi System. *Processes*. 2021; 9(8):1450. doi:10.3390/pr9081450.
 - [5] A. Nassif, M. Talib, Q. Nasir, F. Dakalbab, Machine learning for anomaly detection: A systematic review, in: *IEEE Access*, 9, 78658-78700. 2021. doi:10.1109/ACCESS.2021.3083060.
 - [6] L. Li, X. Mu, S. Li and H. Peng, A Review of Face Recognition Technology, in: *IEEE Access*, vol. 8, pp. 139110-139120, 2020. doi: 10.1109/ACCESS.2020.3011028.
 - [7] P. Perera, P. Oza, V. Patel, One-Class Classification: A Survey. *ArXiv*. 2021. doi: 10.48550/arXiv.2101.03064.
 - [8] N. Seliya, A. Abdollah Zadeh, M. Taghi, A literature review on one-class classification and its potential applications in big data. *Journal of Big Data* 8, 2021: 1-31. doi: 10.1186/s40537-021-00514-x.
 - [9] N. Damer, J. H. Grebe, S. Zienert, F. Kirchbuchner, A. Kuijper, On the Generalization of Detecting Face Morphing Attacks as Anomalies: Novelty vs. Outlier Detection, 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), Tampa, FL, USA, 2019, pp. 1-5. doi: 10.1109/BTAS46853.2019.9185995.
 - [10] H. Xu, G. Pang, Y. Wang, Y. Wang, Deep Isolation Forest for Anomaly Detection, in: *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12591-12604, 1 Dec. 2023, doi: 10.1109/TKDE.2023.3270293
 - [11] T. Vaiyapuri, A. Binbusayyis, Application of deep autoencoder as an one-class classifier for unsupervised network intrusion detection: a comparative evaluation. *PeerJ Computer Science* 6:e327, 2020. doi:10.7717/peerj-cs.327.
 - [12] P. Oza and V. M. Patel, Active Authentication using an Autoencoder regularized CNN-based One-Class Classifier,"2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 2019, pp. 1-8. doi: 10.1109/FG.2019.8756525.
 - [13] S. Prykhodko, A. Prykhodko, I. Shutko, Estimating the Size of Web Apps Created Using the CakePHP Framework by Nonlinear Regression Models with Three Predictors, in: 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT). Vol. 1. IEEE, 2021. doi: 10.1109/CSIT52700.2021.9648680
 - [14] S. Prykhodko, L. Makarova, K. Prykhodko, A. Pukhalevych, Application of Transformed Prediction Ellipsoids for Outlier Detection in Multivariate Non-Gaussian Data, in: 2020 IEEE 15th Inter-national Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), pp 359-362, 2020. doi: 10.1109/TCSET49122.2020.235454.
 - [15] J. Avanija, K. Madhavi, G. Sunitha, S. Sangapu, S. Raju, Facial Expression Recognition using Convolutional Neural Network, in: 2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR), pp. 1-7, 2022. doi: 10.1109/ICAITPR51569.2022.9844221.
 - [16] T. Etherington, Mahalanobis distances for ecological niche modelling and outlier detection: implications of sample size, error, and bias for selecting and parameterising a multivariate location and scatter method. *PeerJ*, 9. 2021. doi: 10.7717/peerj.11436.
 - [17] R. Ghiasi, M. A. Khan, D. Sorrentino, C. Diaine, A. Malekjafarian, An unsupervised anomaly detection framework for onboard monitoring of railway track geometrical defects using one-class support vector machine. *Engineering Applications of Artificial Intelligence*, 133, 108167, 2024. doi:10.1016/j.engappai.2024.108167.
 - [18] S. Todkar, V. Baltazart, A. Ihamouten, X. Dérobert, D. Guilbert, One-class SVM based outlier detection strategy to detect thin interlayer debondings within pavement structures using Ground Penetrating Radar data. *Journal of Applied Geophysics*, 192, 104392, 2021. doi: 10.1016/j.jappgeo.2021.104392.

- [19] T. Duong, NS. Vo, L. Nguyen, QT. Vien, VD. Nguyen, Anomaly Detection Using One-Class SVM for Logs of Juniper Router Devices. *Industrial Networks and Intelligent Systems. INISCOM 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 293. Springer, Cham. 2019. doi:10.1007/978-3-030-30149-1_24.
- [20] J. Lesouple, C. Baudoin, M. Spigai, JY. Tournieret, Generalized isolation forest for anomaly detection, *Pattern Recognition Letters*, Volume 149, 2021, Pages 109-119, ISSN 0167-8655. doi:10.1016/j.patrec.2021.05.022.
- [21] Y. Chabchoub, M. U. Togbe, A. Boly, R. Chiky, An In-Depth Study and Improvement of Isolation Forest, in: *IEEE Access*, vol. 10, pp. 10219-10237, 2022. doi: 10.1109/ACCESS.2022.3144425.
- [22] S. Imashev, Extended isolation forest–application to outlier detection in geomagnetic data, in: *IOP Conference Series: Earth and Environmental Science*. Vol. 929, No. 1, p. 012022. 2021, November. doi: 10.1088/1755-1315/929/1/012022.
- [23] M. U. Togbe, M. Barry, A. Boly, Y. Chabchoub, R. Chiky, J. Montiel, T.V. Tran, Anomaly detection for data streams based on isolation forest using scikit-multiflow, in: *Computational Science and Its Applications–ICCSA 2020: 20th International Conference, Cagliari, Italy, July 1–4, 2020, Proceedings, Part IV 20* (pp. 15-30). Springer International Publishing. doi: 10.1007/978-3-030-58811-3_2.
- [24] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, F. Sabrina, Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset, in: *IEEE Access*, vol. 9, pp. 140136-140146, 2021, doi: 10.1109/ACCESS.2021.3116612.
- [25] Z. Wang, Y-J. Cha, Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage. *Structural Health Monitoring*. 2021; 20(1):406-425. doi:10.1177/1475921720934051.
- [26] O-C. Novac, MC. Chirodea, CM. Novac, N. Bizon, M. Oproescu, OP. Stan, CE. Gordan, Analysis of the Application Efficiency of TensorFlow and PyTorch in Convolutional Neural Network. *Sensors*. 2022; 22(22):8872. doi:10.3390/s22228872.
- [27] B. Min, J. Yoo, S. Kim, D. Shin, D. Shin, Network Anomaly Detection Using Memory-Augmented Deep Autoencoder, in: *IEEE Access*, vol. 9, pp. 104695-104706, 2021. doi: 10.1109/ACCESS.2021.3100087
- [28] M. Sewak, S. K. Sahay, H. Rathore, An overview of deep learning architecture of deep neural networks and autoencoders. *Journal of Computational and Theoretical Nanoscience*, 17(1), 182-188. 2020. doi: 10.1166/jctn.2020.8648.
- [29] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, F. Sabrina, Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset, in: *IEEE Access*, vol. 9, pp. 140136-140146, 2021. doi: 10.1109/ACCESS.2021.3116612.
- [30] I. H. Kartowisastro, J. Latupapua, A Comparison of Adaptive Moment Estimation (Adam) and RMSProp Optimisation Techniques for Wildlife Animal Classification Using Convolutional Neural Networks. *Revue d'Intelligence Artificielle*, Vol. 37, No. 4, pp. 1023-1030. 2023. doi: 10.18280/ria.370424.
- [31] W. Hilal, S. A. Gadsden, J. Yawney, Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. *Expert Systems with Applications*. Volume 193, 2022, 116429, ISSN 0957-4174. doi:10.1016/j.eswa.2021.116429.
- [32] S. Nurmaini, A. Darmawahyuni, AN. Sakti Mukti, MN. Rachmatullah, F. Firdaus, B. Tutuko, Deep Learning-Based Stacked Denoising and Autoencoder for ECG Heartbeat Classification. *Electronics*. 2020; 9(1):135. doi:10.3390/electronics9010135.
- [33] S. Prykhodko, N. Prykhodko, L. Makarova, K. Pugachenko, Detecting outliers in multivariate non-Gaussian data on the basis of normalizing transformations, in: *Proceedings of the First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, IEEE, 2017, pp. 846–849. doi:10.1109/UKRCON.2017.8100366.

Development of an embedded operating system based on the Linux kernel for SoC FPGA

Yurii Herman¹, Oleh Krulikovskiy^{1,2}, Serhii Haliuk¹ and Sergey Subbotin³

¹Yuriy Fedkovych Chernivtsi National University, Department of radioengineering and information security, St. Storozhynetska, 101, Chernivtsi, 58012, Ukraine

²Ștefan cel Mare University of Suceava, Str. Universității 13, Suceava, 720229, România

³National University "Zaporizhzhia Polytechnic", St. Zhukovsky, 64, building 3, Zaporizhzhia, 69063, Ukraine

Abstract

In this paper, we create a distribution of the embedded operating system for Cyclone V SoC FPGA platforms based on the Linux kernel. A comparison of well-known open-source tools for creating embedded operating systems is demonstrated. A step-by-step example of embedded OS synthesis using custom scripts and a simplified pipeline is given, which increases the adaptability of the target system. The possibility of adding hardware-oriented tools for interaction between SoC and FPGA is demonstrated. This makes it possible to create a wide range of hardware applications with remote access. The proposed approach is also vendor-independent and can be applied to other FPGA SoCs.

The final system, although not significantly different in resource requirements from Yocto, is more adaptable and can be ported to the Yocto base if necessary. This allows us to make the most of the Full Custom approach, ensuring an optimal balance between development efficiency, responsiveness to changes, and system resource requirements.

Keywords

FPGA SoC, embedded operating systems, Linux kernel, system flexibility, extensibility, drivers, user space, dynamic development

1. Introduction

The compliance of modern computing platforms with Moore's Law requires developers to use new architectural approaches during development. One of these is a variety of systems on a chip (SoC). Among the wide variety of SoCs for real-time systems, SoC FPGAs (Field-Programmable Gate Array) have become widely used [1]. SoC FPGAs are integrated circuits that combine the functionality of a microprocessor core (CPU) with the capabilities of a programmable logic gate array (FPGA) on a single chip. This is an important class of microelectronics characterized by high programming flexibility, high-performance computing, and efficient use of resources [2-4]. In real-time embedded systems, FPGAs play a key role due to their versatility and ability to solve tasks in signal processing, controllers, image processing, and more [5]. Even though FPGAs provide outstanding characteristics (Flexible programming, Integration, Performance, Scalability), using them in their pure form requires high costs and qualifications in the development of modern applications.

Accordingly, to reduce the complexity of development, all real-time embedded applications must interact with the embedded operating system [6]. This is because it is difficult to manually create an optimized system without an OS. After all, when tested, such a system will not work better than similar solutions that use an OS [7]. Also, this approach allows you to better focus on system-level optimization when developing applications based on SoC FPGA. Because the operating system will be maximally optimized for the architecture of the microprocessor. The hardware is often fully optimized for the FPGA architecture. At the same time, communication between the software and hardware parts will be carried out through optimized bridges [8].

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ yurii.herman@chnu.edu.ua (Yu. Herman); o.krulikovskiy@chnu.edu.ua (O. Krulikovskiy); s.haliuk@chnu.edu.ua (S. Haliuk); subbotin@zntu.edu.ua (S. Subbotin)

ORCID iD 0009-0003-2473-7365 (Yu. Herman); 0000-0001-5995-6857 (O. Krulikovskiy); 0000-0003-3836-2675 (S. Haliuk); 0000-0001-5814-8268 (S. Subbotin)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

An embedded OS is a specialized operating system for computer systems embedded in other systems. These OS are computer systems designed for a specific purpose, to increase functionality and reliability for a specific task [9].

An important difference between most embedded operating systems and desktop operating systems is that the application, including the operating system, is usually statically compiled into a single package. Speaking of the Linux kernel, it is worth noting that embedded OSes based on it have features that positively distinguish them from specialized OSes. Resource efficiency is achieved at the cost of losing some functionality or granularity that larger computer operating systems provide, including features that may not be used by the specialized systems on which they run. Depending on the multitasking method, this type of OS is often thought of as a real-time operating system, or RTOS. Embedded systems most often use real-time operating systems. QNX, FreeRTOS, and VxWorks are the most widely used embedded operating systems today [10-12].

Most embedded operating systems are proprietary and closed to modification. The field of embedded development is very complex and highly demanding for engineers. Open Source Software (OSS) allows the creation of task and platform-oriented OSes without the need to implement low-level layers from scratch. In the field of SoC FPGA and embedded OSes, two systems for assembly and construction are attracting the most attention Yocto Project + Poky [13] and Buildroot [14]. These tools are representatives of the FPGA supported and used by large companies (Intel, AMD). Using them allows the use of the entire FPGA library and the experience of professionals from open-source communities [15].

Accordingly, the work aims to compare the existing approaches to building an embedded OS using open-source tools using the example of the DE10-Nano board [16] with SoC FPGA Cyclone V. Since the process of synthesizing an embedded OS, except for some points, is vendor-independent and will be similar for other SoC FPGAs.

The paper is organized as follows. A comparison of different OS development approaches and their impact on the efficiency of the FPGA SoC system is presented in section 2. The step-by-step process of building an embedded OS with hardware-specific extensions for FPGA SoC is presented in section 3. The development of a user space system and conclusions are presented in sections 4 and 5 respectively.

2. Comparison of different OS development approaches and their impact on the efficiency of the FPGA SoC system

As already mentioned above, the two most widely used tools for developing an embedded OS distribution are Yocto Project + Poky and Buildroot. A third alternative is development using custom scripting.

2.1. Yocto Project

A Linux distribution is a set of software packages and the ways they interact. There are hundreds of Linux distributions available. Most of them are not designed for embedded systems, lack the flexibility needed to achieve target sizes and change functionality, and are not suitable for systems with limited resources.

The Yocto project is a Linux distribution factory that uses several other open-source projects [15]. Yocto Project, on the other hand, is a distribution creation tool in its own right. It allows you to create a Linux distribution designed for a specific system.

2.2. BuildRoot

Unlike Yocto, BuildRoot uses more general technologies, which negatively affects its flexibility. BuildRoot is a set of make files and patch aggregation that simplifies and automates the process of creating a complete and bootable Linux environment for an embedded system while using cross-compilation to create multiple target platforms on a single Linux-based host machine. Buildroot can automatically assemble the necessary cross-compilation tools, create the root file system, compile the Linux kernel image, and generate the bootloader for the target system, or

perform any independent combination of these actions. For example, an already installed cross-compilation tool can be used independently, and Buildroot only creates the root file system[17].

Buildroot is primarily designed for use in small or embedded systems based on a variety of computer and instruction set architectures (ISAs), including x86, ARM, MIPS, and PowerPC.

2.3. Full Custom

Unlike the options described above, the Full Custom approach allows for a multifunctional system that can be updated and modified on the fly and is much more flexible in the development and prototyping process [18]. When building the OS itself, we can make changes to the software package itself, which will be standard in it, and can choose the type of Linux wrapper depending on the needs of the project (Arch Linux, Debian, etc.) [19]. This option is also capable of providing access to external resources and packages for on-the-fly installation, which greatly simplifies the process of iterating in development. Full Custom, unlike the previous approaches, is not characterized by the convenience of initial development and requires the entire image generation process. We need to assemble all parts of the OS ourselves, starting with the kernel and ending with the file system configuration.

It is a more comprehensive option that gives us the ability to control every element of the resulting OS and assemble the software and tool packages needed to get the system up and running, and if necessary, easily adapt to updating these packages from a remote source. Another significant positive factor is the ability to control what exactly will be updated in the next iteration of the distribution.

However, it should be kept in mind that that approach's adoption has higher requirements for the engineer, as the process itself needs to be fully controlled.

2.4. Comparison of Yocto, BuildRoot, and Full Custom

Overall, both frameworks can produce the same resulting distribution: a root file system image for an embedded device.

Buildroot trying to be as simple to use as possible. Its main tool is made to be as compact and user-friendly as possible, so engineers can catch up with all the tools and start developing as fast as possible. Specific features are built by extensions, allowing to create products using standard tools that are included in Buildroot distribution. The OS image created by Buildroot is usually oriented to be minimalistic, making builds quick and allowing you to create a generic system rather than one tailored to a specific use case.

In contrast, Yocto is much more robust and supports a wide range of embedded systems. To make that possible, configuration is defined in scripts (called 'recipes') that declare what software to build and how to build it, while allowing to gather recipes into layers, which can be used are collections of recipes written and maintained by the development community or by the distribution author himself [20].

Full custom is based on the step-by-step creation of a distribution, with full control of each element. Initially, this is not a fully automated process that relies heavily on the knowledge of an engineer. However, this process can be easily adapted to automation and allows you to build a process that is necessary specifically for a particular task.

The result of Buildroot is an image of the root file system, nothing more. Every time you need to update the system, the whole product will be rebuilt from scratch to avoid discrepancies.

The result of Yocto is a "distribution". More precisely a fully done OS with specific tools, configuration and packages. It is worth mentioning that integration with external package providers so the system can be modified while booted, requires development and is not pre-built. Yocto supports development on the board by providing SDK, to avoid continuous rebuilds of distribution for each modification.

Builds created with Buildroot and Yocto are also different in configuration. Buildroot stores configuration in a main file that can be edited using the kernel's kconfig tool (e.g. xconfig or menuconfig). That file basically contains all the configuration for built OS, including user space part.

In the case of the Full Custom approach, the result is both the final image and all its components, which can be easily modified depending on the needs of the project.

3. Building an embedded OS for FPGA SoC

3.1. Building Boot-Loader

As was mentioned above, the process of building a Linux-based distribution consists of a couple of steps (see Fig. 1).

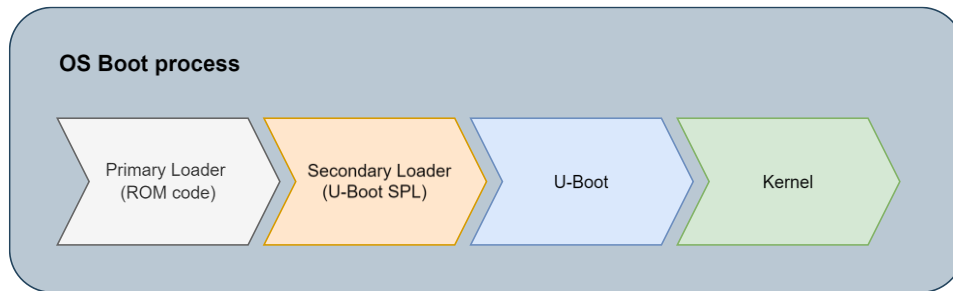


Figure 1: OS start steps diagram

The boot-loader is a crucial one because all the next parts of the OS are started by it and U-Boot handles all basic system configuration. We are using U-Boot because it is the most used and proven one, and so it usually is familiar to engineers while having an extensive knowledge base available. To start with, we can get publicly available source code and choose a specific build version, as shown in Fig. 2.

```
git clone https://github.com/u-boot/u-boot.git
cd u-boot
# We need to get list of all available u-boot versions
git tag
# And select desired one
git checkout $TAG
```

Figure 2: Getting U-Boot source code

Generally, the default configuration is enough to start with, but if we do need to customize something deeply, all configuration headers are open. For example, we can modify data and variables that are stored in the U-Boot environment or add new ones. When U-Boot starts running the kernel, it reads most of the configuration from the environment and runs scripts/code based on it.

For example, Fig. 3 demonstrates how we can anchor our OS to only a specific MAC address by modifying the header file.

```
// configs/socfpga_common.h
#define CONFIG_EXTRA_ENV_SETTINGS \
    "fdtfile=" CONFIG_DEFAULT_FDT_FILE "\0" \
    "bootm_size=0xa000000\0" \
    "kernel_addr_r=__stringify(CONFIG_SYS_LOAD_ADDR)" "\0" \
    "fdt_addr_r=0x02000000\0" \
    "scriptaddr=0x02100000\0" \
    "pxefile_addr_r=0x02200000\0" \
    "ramdisk_addr_r=0x02300000\0" \
    "socfpga_legacy_reset_compat=1\0" \
    BOOTENV
#endif
// We are able to add variables into block above, for e.g
// "ethaddr=static:mac:address\0" \
// Where \0 serves as mark for end of the string line
```

Figure 3: Environment configuration for boot

Most of the settings for U-Boot are stored in those header files, and they are used in compilation, which means that specific releases are hard-linked to those header files. We can

change some of those variables (e.g. we can set the ethernet address while U-Boot is starting). Compilation for U-Boot is straightforward (see Fig. 4) after the modification of headers is done.

```
# We are globally selecting architecture to build for
export ARCH=arm
# And setting compiler
export CROSS_COMPILE=/usr/bin/arm-linux-gnueabi-
# We need to compile default configuration into binary first
# socfpga_de10_nano_defconfig - is preset for it
# Plenty of them are stored in configs folder
# ls -l configs/socfpga*
make socfpga_de10_nano_defconfig
# We can change default platform configuration using
menuconfig
make menuconfig
# To get resulting boot-loader we are running make
# where -j is used to run compilation for in parallel
make -j 24
```

Figure 4: Script for generation of binary file

Once compilation completes we can find out the U-Boot distribution by searching for file `u-boot-with-spl.sfp`. We will use that binary file when all the OS will be compressed in one image, as the launching point for the whole system.

3.2. Selecting the configuration and setting the Linux kernel parameters.

When building an OS, we need to go through several steps:

1. Get Preloader is a binary file provided by the manufacturer, in our case Intel/Altera. It performs some basic configuration before passing it to the bootloader.
2. Compile Bootloader - software that performs hardware initialization before passing it to the kernel for OS initialization.
3. Configure Kernel - The kernel is the heart of the OS and contains all the information about the hardware on the board.
4. Build RootFS - Create a root file system that contains the basic packages and dependencies.

When developing embedded systems, one of the key steps is to select the configuration and configure the Linux kernel options. This process includes choosing a basic RootFS configuration and selecting the kernel that best suits our needs [21].

In our case, the basic configuration of RootFS is based on Arch Linux, and the kernel chosen is altera-opensource (as shown in Fig.5). The reason for choosing the Altera-opensource kernel was the ability to program the FPGA at startup or directly from the operating system, which is supported only by this Altera kernel modification.

```
# Here we are using an Altera fork of Linux Kernel
git clone https://github.com/altera-opensource/linux-socfpga.git
# github.com/torvalds/linux can be used too
cd linux-socfpga
# We need to get a full list of versions that are available
git branch -a

# Use the branch you prefer.
git checkout $BRANCH
```

Figure 5: Kernel selection script

Therefore, we will use this particular fork in the future, after compiling it. Configuring Linux kernel parameters is an important step in the process of developing embedded systems. For this purpose, special tools, such as *menuconfig*, as it specified in Fig. 6, are used to configure various

parameters that determine the behavior and functionality of the kernel. Here are some basic steps to configure Linux kernel settings

```
# We should specify target architecture and compiler before
# compilation is started
export CROSS_COMPILE=/usr/bin/arm-linux-gnueabi-
export ARCH=arm
# First and foremost we need to generate default configuration
make socfpga_defconfig
# That configuration can be tailored using menuconfig tool
make menuconfig
# When settings are made and finished we are going to compile
# kernel image
make LOCALVERSION=zImage -j 24
# LOCALVERSION allows us to specify how image should be named
```

Figure 6: Build process of kernel configuration tool

To do this, we need to enable the 'Overlay filesystem support' and 'Userspace-driven configuration filesystem' in the kernel during configuration (see Fig. 7).

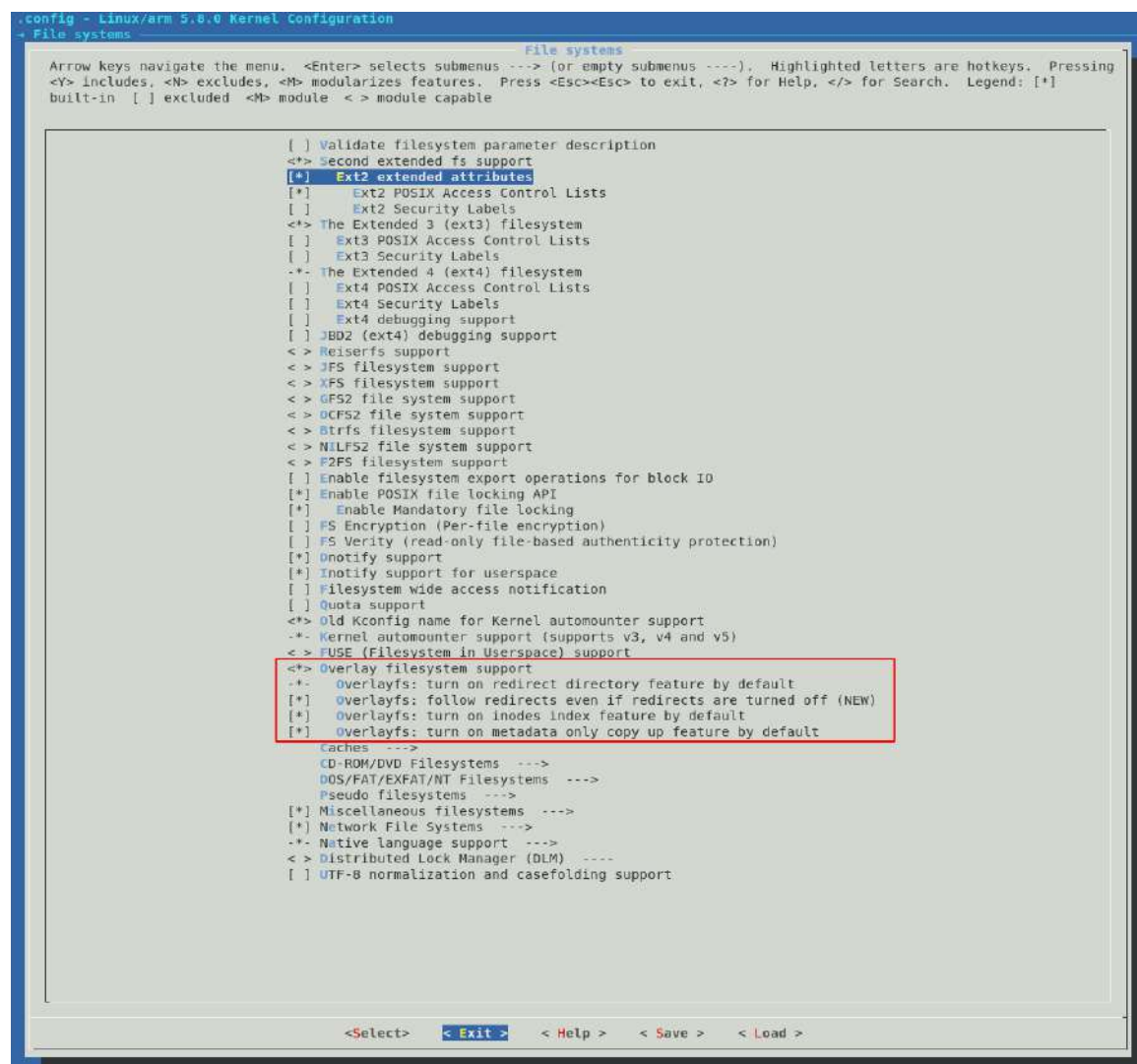


Figure 7: Configuration menu

In the future, we can change the configuration depending on our needs, but as an initial configuration, our version is sufficient.

After we've done with configuration we can build the image of said kernel using the steps from Fig. 8.

```
# When kernel properties are built and configured we are going
# to compile kernel image
make LOCALVERSION=zImage -j 24
# LOCALVERSION allows us to specify how image should be named
```

Figure 8: Building the kernel image

As a result of compilation, we will get a binary zImage file, which will be used as a main Kernel executable. That binary file can be replaced separately from all other elements of system, allowing engineers to freely modify only the needed part of the distribution.

3.3. Flexibility and extensibility of the system based on the embedded operating system.

The flexibility and extensibility of a system based on an embedded operating system are important aspects, as they determine the ability to adapt and expand the system's functionality in the future. Let's take a closer look at these aspects:

- System flexibility:
 - 1. Configuration:
Embedded operating systems typically have a flexible architecture that allows for customization of parameters and options to meet the requirements of a particular application.
 - b. Modularity:
The use of a modular architecture makes it easy to add and remove functional modules, which makes the system more flexible in changing requirements and project development.
- System extensibility:
 - 1. Support for new devices:
Embedded operating systems have mechanisms to dynamically add and support new devices without the need to recompile or flash the kernel.
 - b. Extensibility of functionality:
The system can easily add new features and services without significant changes to the source code or architecture.
- Means for expansion:
 - 1. Dynamic libraries and modules:
Using dynamic libraries and modules allows you to add new features without rebooting your system or recompiling your applications.
 - b. Network services:
Implementation of network services that allow dynamic application loading and system configuration over the network.

3.4. The impact of the embedded OS on the flexibility and extensibility of FPGA SoC designs.

The use of an embedded operating system (OS) such as Linux in FPGA system-on-chips (SoC FPGA) significantly increases the flexibility and extensibility of projects. Due to the configurability of Linux, developers can customize the system to meet specific project requirements, including the choice of supported devices and functionality. The modular architecture of Linux allows you to dynamically add and remove features through the kernel or load modules while the system is running, which allows you to quickly expand functionality without a complete system reboot. In addition, Linux supports FPGA software upgrades without the need to shut down the system, which makes it easy to adapt to changes and implement new features and capabilities without much effort.

In the case of Full Custom Linux, we have access to many external modules, which means we can quickly reconfigure the entire system or its specific parts.

Since the basic distribution includes a set of tools for working with FPGA, you can make changes to the firmware directly from the OS, as well as read the FPGA configuration and perform basic control over it. Similarly, we can read and write from the HPS-to-FPGA registers.

To increase the flexibility of the system and simplify the development process, a separate set of packages has been built in. Using these tools allows you to automate interaction, create interfaces, and reduce the time required to update the FPGA circuitry and structure.

Table. 1

List of installed utilities [22]

Name	Functionality
FPGA-status	Reading the status of an FPGA structure
FPGA-readMSEL	Reading FPGA configuration mode via MSEL-Bit
FPGA-reset	Reset the FPGA structure and delete the current configuration
FPGA-writeConfig	Writing a new FPGA configuration
FPGA-readBridge	Read addresses (32-bit register) from HPS-to-FPGA Bridge, Lightweight-HPS-to-FPGA Bridge, or MPU (HPS) memory
FPGA-writeBridge	Writing addresses (32-bit register) to HPS-to-FPGA Bridge, Lightweight-HPS-to-FPGA Bridge, or MPU (HPS) memory
FPGA-gpiRead	Reading FPGA general purpose registers (GPI)
FPGA-gpoWrite	Writing FPGA general purpose registers (GPOs)

In addition to the above package, it is also worth adding the ability of the OS to read **.rbf** files when the kernel is running and update the FPGA part with a new version of the schematic. This set of functionality has a positive impact on the flexibility of the entire system, simplifies development, and allows you to implement dynamic processes while iterating on a project.

Access to external resources and repositories allows you to expand the system with the help of specific packages, both those provided by vendors and those published by the system developers. Using the embedded OS allows you to flexibly respond to changes in the environment, tasks, or development vector. It also makes it possible to implement more complex systems and combine them, expand their functionality, and provide access to external resources.

3.5. Tools and strategies to maximize system flexibility.

To achieve maximum flexibility in an embedded system, various tools and strategies can be used:

Modular architecture: Developing a system based on a modular architecture allows you to divide the system into small modules that can be independently developed, tested, and updated. This provides flexibility to make changes and extend functionality.

Use of configuration files: configuration files allow you to change the FPGA configuration, including changing logic, I/O, and other parameters without the need to completely recompile the code.

Use of IP cores: FPGA IP cores are off-the-shelf modules that can be used to implement various functional blocks. The use of IP cores allows you to quickly and efficiently integrate the desired functionality without the need for development.

Memory Management Unit (MMU): The MMU allows you to dynamically manage memory access, which provides flexibility in the use of memory resources and allows you to use memory efficiently for different tasks.

Use of virtual devices: Virtual devices provide an interface between the FPGA and external components or systems, allowing for easier integration with other systems.

The use of tools included in the OS package allows for faster and easier interaction with the FPGA. For example, the FPGA-status and FPGA-reset utilities allow you to read the status and delete an existing configuration, making the system flexible in management and debugging. Also, FPGA-writeConfig allows you to change the configuration, which allows you to quickly adapt the system to new requirements or work scenarios. In addition, the FPGA-readBridge and FPGA-writeBridge utilities extend the capabilities of interaction between the software and hardware environment, and FPGA-gpiRead and FPGA-gpoWrite provide the ability to interact with the system through general-purpose input register signals.

This package makes the system more flexible and adaptable to changes in requirements and use cases.

Similarly, access to external repositories allows you to make changes externally, using both an "over-the-air" update and a rolling release approach, which reduces the time it takes to deliver an update to the system. The support for configuring the FPGA from the FAT partition of the embedded OS allows us to have a stable startup configuration and, if necessary, return to it.

Since the system can update the configuration during operation, we can use a modular approach and targeted development of certain modules for functionality or to solve certain tasks facing the project.

Adding to this the ability to cross-communicate between FPGA and OS parts, we can freely operate memory and utilize configurable approaches based on data exchange between system elements. Taken together, the use of the tools described above allows you to build your strategies for expanding and maintaining the designed system.

4. Development of a user space system

Developing the user space in a Linux-based embedded operating system (OS) for an FPGA SoC involves creating software that runs directly on the FPGA SoC. This user space can include various components such as applications, services, drivers, and other software modules that provide the required system functionality [23]. Here are some key aspects of the third point:

1. Application development:

Create applications that interact with the FPGA SoC through various interfaces such as UART, SPI, I2C, Ethernet, etc. These applications can perform data processing, control devices, and provide a user interface to interact with the system.

2. Service development:

Create services that run in the background and provide certain functions or event processing. For example, services can manage network connections, collect and analyze data, and interact with other system components.

3. Driver development:

Create drivers to interact with FPGA SoC hardware components such as peripherals, storage, sensors, etc. These drivers allow the user space to access and interact with hardware resources.

4. Integration with FPGA resources:

The use of FPGA resources to implement specific functions or data processing. This can include the design and integration of logic blocks, signal processing, hardware accelerators, and other elements into the system's user space.

5. Testing and debugging:

Once the software is developed, it is important to test and debug it to ensure that it works correctly, to identify and eliminate errors, and to optimize system performance and reliability.

4.1. Create and configure custom components.

To create a user space, we need to configure and enable RootFS properly. Rootfs is a special instance of ramfs always present on Linux 2.6 systems. It is used as a location inside the Linux kernel, as a list with directories of the file system the place to search. RamFs is also a pointer to the points for starting the operating system, but it is focused more on working with RAM. Most systems simply mount another file system on top of it and ignore it, because the ramfs instance takes up very little space. Rootfs is a file system. In Linux, all file systems have a mount point, which is the directory where the mounted file system connects to the root file system. This is the difference between them, RamFs are the older ancestor of RootFs, but with a focus on placing a small amount of RAM.

The peculiarity of our approach is the use of symlinks (a virtual link to a specific directory that can be located in any other directory), which allows us to split the elements of the distribution while keeping it unified for virtual access.

One of the main differences in creating RootFS for Archlinux ARM is that we need to first create an SD card image, mount it, and then create our RootFS [24]. The build process can be summarized as a step-by-step script (see Fig.9)

```
# We need to start by getting core fs files and turn them into binaries
git clone https://aur.archlinux.org/qemu-arm-static.git

cd qemu-arm-static

makepkg -si

cd..

# Here we are getting filestructure
wget http://sg.mirror.archlinuxarm.org/os/ArchLinuxARM-armv7-latest.tar.gz

cd sdcard/ext4

# For decompression we need to utilize root users permissions
sudo su
tar xf ../ArchLinuxARM-armv7-latest.tar.gz .

cd ..

# To work with emulated rootfs we need to inject core binaries there
sudo mkdir -p ext4/usr/bin
sudo cp /usr/bin/qemu-arm-static ext4/usr/bin

# After preparations are done we can initiate session in newly created FS and begin
# customization
sudo arch-chroot ext4 /bin/bash
```

Figure 9: RootFS creation script

Following the scenario above, we will get a functional file system that can be modified depending on the needs and configured to meet the project's requirements. Such file systems do not take up many resources and do not put a significant burden on the entire scheme, since they contain minimal components and basic libraries for their extension. However, if necessary, they can be converted into a user-level system with all its pros and cons.

As a result, we will obtain the generalized OS which consists of several parts that are interconnected with each other (see Fig.10)

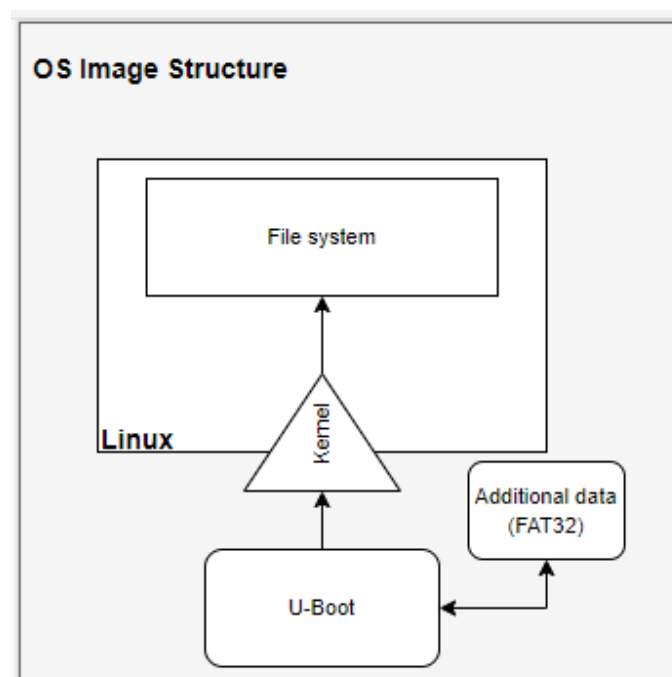


Figure 10: Embedded OS Image Structure

That composition means that we can update the specific part of the Image and the existence of the FAT32 additional component allows us to hook the default configuration that will be used as

a starter, or even a recovery point. Also, we can interact with additional data part from U-Boot, if required.

4.2. Interaction with FPGA SoC system resources from the user space level.

Interacting with FPGA SoC system resources from the user space level of the embedded operating system (OS) can be done using various tools and utilities, including table-based utilities such as FPGA-status, FPGA-reset, FPGA-writeConfig, FPGA-readBridge, FPGA-writeBridge, FPGA-gpiRead, FPGA-gpoWrite, and the devmem tool.

FPGA-reset: Allows you to clear the FPGA configuration to the basic level

FPGA-writeConfig: This utility allows you to write the FPGA configuration from the user space level.

FPGA-readBridge and FPGA-writeBridge: These tools will allow you to interact with bridges and interfaces between FPGA and other system components from the user space level.

FPGA-gpiRead and FPGA-gpoWrite: These utilities allow you to read input signals and write FPGA output signals from the user space level.

devmem: Allows you to read and write data to the device memory from the user space level. This can be useful for accessing various FPGA SoC resources, such as configuration registers, control registers, and others.

In general, the tools already in place allow integration and interaction with various aspects of the FPGA SoC from the OS user space level, making them important tools for simplifying development and interaction with FPGA. In addition to the above capabilities, the user can also use the direct memory access (DMA) option to quickly and efficiently read and write data to the FPGA SoC device memory from the user space level. This enables efficient use of memory resources and increases system performance. In addition, using the embedded Linux-based operating system, the user can easily use packages from external repositories. This extends the development capabilities as it allows the use of off-the-shelf solutions and libraries, which greatly facilitates the development, support, and expansion of the functionality of embedded systems based on FPGA SoC with a Linux environment [25]. In general, the available tools allow you to build interaction through HPS-to-FPGA and direct memory access, which simplifies the development of systems that can process and exchange data with FPGA. Knowing the addresses of the registers or the bus communication protocol, we can directly read and write binary data to this register using most programming languages or scripting tools (shell, as an example).

4.3. Using libraries to facilitate code development and optimization.

There are many advantages to using libraries for development on the embedded Linux for FPGA SoC with access to external repositories. Firstly, it allows us to significantly reduce development time, as we can use ready-made solutions to interact with hardware and software resources such as GPIO, SPI, UART, etc. Libraries also contribute to code optimization, as they provide an efficient interface to interact with these resources without the need to write your own code from scratch.

Furthermore, the use of external resources allows you to access a wide range of libraries and modules that extend the system's functionality. For example, we can include libraries for networking, data processing, creating a graphical interface, and much more. This helps to make the project more flexible and adaptive to changes in requirements or new features. In addition, by knowing which libraries and modules are required for your project, it is possible to create a standard package that can be included in a new release of the embedded OS. This simplifies the deployment and support of the system, as the standard package contains all the necessary components for the new version to work on the circuit.

Prototyping directly on the FPGA significantly reduces development time and makes it easier to understand how a system might behave in real-world conditions. Instead of using simulations or emulation environments, we can work directly with the FPGA, which allows us to identify and fix bugs faster and test different system scenarios more efficiently.

Thanks to our architecture, we can directly obtain test versions of the software code and check and modify them dynamically. This ensures a fast development cycle as we can work on new

features and modules efficiently. We can also update the FPGA circuit to suit our project without rebuilding the entire system using the tools described above. This allows us to respond to changes in requirements or identified issues and quickly make the necessary changes to the system hardware without stopping the software.

5. Conclusions

In the summary of the Linux kernel-based OS distribution for embedded systems based on FPGA SoC compared to Yocto and Buildroot, we concluded that using the Full Custom system with a set of built-in tools significantly speeds up development and simplifies prototyping. Based on this information, we built an OS distribution focused on rapid adaptation and simplification of project development on the FPGA SoC platform.

The resulting system has the necessary components built in to work on projects and the ability to dynamically add and remove libraries and modules. This approach greatly facilitates the process of code development and optimization. In addition, the availability of utilities that allow direct interaction with the FPGA system allows you to get test versions of the software code, check and modify them dynamically, and update the FPGA circuit according to the project requirements. This ensures efficient and fast system development based on the embedded Linux OS for FPGA SoC.

The final system, although not significantly different in resource requirements from Yocto, is more adaptable and can be ported to the Yocto base if necessary. This allows us to make the most of the Full Custom approach, ensuring an optimal balance between development efficiency, responsiveness to changes, and system resource requirements.

Acknowledgments

Oleh Krulikovskyi thanks the Intel company for the equipment provided within the Intel FPGA Academic Program.

References

- [1] Ross K. Snider, Advanced Digital System Design using SoC FPGAs, Springer Cham, 2023. <https://doi.org/10.1007/978-3-031-15416-4>
- [2] Arifur Rahman, FPGA Based Design and Applications (1st. ed.). Springer Publishing Company, Incorporated, 2008.
- [3] Andres Upegui, Andrea Guerrieri and Laurent Gantel (Ed.), Special Issue Reprint Applications Enabled by FPGA-Based Technology, Electronics, MDPI, Basel, 2023. doi.org/10.3390/books978-3-0365-8785-1
- [4] Dheeraj Punia (December 7, 2023) FPGA Design, Architecture and Applications URL:<https://www.logic-fruit.com/blog/fpga/fpga-design-architecture-and-applications/>
- [5] Fuming Sun, Xiaoying Li, Qin Wang and Chunlin Tang, FPGA-based embedded system design, in: Proceedings of the 2008 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2008, Macao, 2008. doi: 10.1109/APCCAS.2008.4746128.
- [6] Mastura D. Marieska, Paul G. Hariyanto, M. Firda Fauzan, Achmad Imam Kistijantoro, and Afwarman Manaf, On Performance of Kernel Based and Embedded Real-Time Operating System: Benchmarking and Analysis, in: Proceedings of the International Conference on Advanced Computer Science and Information System. ICACSIS 2011, Mercure Convention Centre, Jakarta, Indonesia, 2011 .
- [7] Bare-Metal, RTOS, or Linux? Optimize Real-Time Performance with Altera SoCs URL: https://people.ece.cornell.edu/land/courses/ece5760/DE1_SOC/wp-01245-optimize-real-time-performance-with-altera-socs.pdf.
- [8] Cyclone®V Hard Processor System Technical Reference Manual (cv_5v4 2023.08.28). URL: <https://www.intel.com/programmable/technical-pdfs/683126.pdf>.
- [9] Building embedded Linux for the terasic de10-nano URL: https://bitlog.it/20170820_building_embedded_linux_for_the_terasic_de10-nano.html

- [10] BlackBerry QNX: Real-Time OS and Software for Embedded Systems URL: <https://blackberry.qnx.com/en>
- [11] FreeRTOS™ Real-time operating system for microcontrollers URL: <https://www.freertos.org/>
- [12] VxWorks: The Leading RTOS for the Intelligent Edge. URL: <https://www.windriver.com/products/vxworks>
- [13] The Yocto Project It's not an embedded Linux distribution, it creates a custom one for you. URL: <https://www.yoctoproject.org/>
- [14] The Buildroot user manual URL: <https://buildroot.org/downloads/manual/manual.html>
- [15] Welcome to OpenEmbedded URL: https://www.openembedded.org/wiki/Main_Page
- [16] Terrasic DE10-Nano Manual 2018
- [17] Yocto Project Documentation 2024 URL: <https://www.yoctoproject.org/docs/>.
- [18] Embedded Linux System Design and Development / P.Raghavan Amol Lad Sriram Neelakandan Taylor & Francis 2019
- [19] Linux From Scratch Gerard Beekmans URL: <https://www.linuxfromscratch.org/lfs/>
- [20] Alex Gonzalez. Embedded Linux Development Using Yocto Project / Chris Simmonds. – BIRMINGHAM - MUMBAI: PACKT, 2018.
- [21] What is the Linux kernel? (2019, February 27). RedHat. URL: <https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel>
- [22] Robin Sebastian RSTools URL: <https://github.com/robseb/rstools>
- [23] Abbott, D. Linux for Embedded and Real-Time Applications; Butterworth-Heinemann: Oxford, UK, 2003.
- [24] Rocketboards WIKI URL: <https://www.rocketboards.org/foswiki/Documentation/WebHome>
- [25] Li Y, Matsubara Y, Takada H (2018) A comparative analysis of RTOS and Linux scalability on an embedded many-core processor. J Inf Process 26:225–236

A Method of Control and Operational Diagnostics of Data Errors Presented in a Non-positional Number System in Residual Classes

Alina Yanko¹, Victor Krasnobayev² and Oleg Kruk¹

¹ National University «Yuri Kondratyuk Poltava Polytechnic», Pershotravneva Avenue 24, Poltava, 36011, Ukraine

² N. Karazin Kharkiv National University, Svobody sq., 4, Kharkiv, 61022, Ukraine

Abstract

The basis of modern infocommunication systems is computer means of data transmission and processing. Therefore, one of the alternative means of achieving maximum efficiency in the functioning of the infocommunication systems when processing data in real time is to improve, first of all, such characteristics of computer data processing systems (CDPS) as the reliability and performance of information processing, as well as the fault tolerance of its functioning.

Currently, the quality of implementation of information processing procedures is largely determined by the selected mathematical model for organizing the information processing process in the CDPS. Therefore, research and finding ways to solve the problem of increasing the reliability of real-time CDPS, without reducing the productivity of processing large data arrays based on new mathematical models, is an urgent task. One of the possible innovative ways of solving the formulated problem is the usage of a non-positional number system in residual classes (NPNSRC) to create CDPS. The versatility of NPNSRC codes is explained not only by their high correcting abilities, arithmetic and the ability to fight against error packets, but also by their adaptability to flexible changes in correcting properties, without changing the coding method. The article discusses issues related to the control and diagnosis of data errors in the CDPS operating in the NPNSRC. The main attention is paid to the consideration of a method for quickly diagnosing data errors in the NPNSRC. Reducing diagnostic time increases the efficiency of diagnosing solitary errors in a non-positional code structure in the NPNSR. A specific example of the implementation of a method for diagnosing data errors in the NPNSRC is given.

Keywords

Computer data processing system, diagnosing data errors, non-positional code structure, non-positional number system in residual classes, number projection, orthogonal basis.

1. Introduction

An analysis of existing methods for increasing fault tolerance has shown that the best and most widely used in practice today are two methods: redundancy and control, diagnostics with further restoration of the CDPS functionality. When choosing control and diagnostics methods, the main attention should be paid to the ability of this control and diagnostics method to detect errors, as well as the amount of equipment and time spent on control [1-3].

The principles of control of non-positional code structures in the NPNSRC are the same as the principles of control in the positional number system (PNS), while taking into account the principles of formation of the NPNSRC and the influence of properties of the NPNSRC on the structure of the CDPS [4], let us note, in a general way, the principles of data control in non-positional code structures: the principle of reliability of control; the principle of continuity of control; the principle of operational control.

The most effectiveness from the usage of the NPNSRC is accomplished in cases when the realized algorithms comprise arithmetic operations such as addition, multiplication and subtraction [5]. However, in a CDPS operating in the NPNSRC, in addition to the above arithmetic operations, it is necessary to carry out so-called non-modular (positional) operations [6-8]. Such operations include control operations and correction (diagnosis and correction) of

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ al9_yanko@ukr.net (A. Yanko); v.a.krasnobayev@gmail.com (V. Krasnobayev); olegkruk1975@gmail.com (O. Kruk)

ORCID 0000-0003-2876-9316 (A. Yanko); 0000-0001-5192-9918 (V. Krasnobayev); 0009-0004-4241-2676 (O. Kruk)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

data errors. Data control, diagnosis and correction operations, compared to arithmetic operations in the NPNSRC, require significant time for their implementation. The need to implement control (monitoring) operations, diagnosis and correction of data reduce the overall effectiveness of the usage of the NPNSRC in real-time CDPS. When processing data in real time, the considerable time required for the implementation of control, diagnostic, and error correction procedures calls into question the feasibility of using NPNSRC as a general-purpose CDPS. The need to ensure the high efficiency of the functioning of the CDPS in the NPNSRC requires the development and implementation of methods of operational data control, diagnostics and correction, other than methods employed in ordinary binary PNS [9].

Within the framework of the concept of development of fast-acting and reliable CDPS presented in the NPNSRC, the urgent task is the development and application of methods and means of operational data control, diagnosis and error correction. In the article, the main focus is devoted to data diagnostics process in the NPNSRC.

The purpose of the research is to develop a method for controlling (monitoring) and operational diagnostics errors in data presented in the NPNSRC, using the orthogonal basis of partial sets of bases (modules).

2. Data correction process in the non-positional number system in residual classes

The correction process (detection and correction) of errors in the information code structure $\tilde{D}$ of data consists of the following main stages [4, 9]:

- data control (monitoring) (the process of detecting the presence of an error in $\tilde{D} = (d_1 \parallel d_2 \parallel \dots \parallel \tilde{d}_j \parallel \dots \parallel d_k)$, presented in the NPNSRC);
- data diagnostics (localization of error locations with a given diagnostic depth);
- error correction in the non-positional code structure (recovery of distorted residues $\{\tilde{d}_j\}$ ($j = \overline{1, k}$) of the incorrect number $\tilde{D}$ and obtaining the correct number D).

The number $D = (d_1 \parallel d_2 \parallel \dots \parallel d_{j-1} \parallel d_j \parallel d_{j+1} \parallel \dots \parallel d_k)$ in without redundant NPNSRC is represented by a set of residues $\{d_j\}$ ($j = \overline{1, k}$) according to the selected system of information bases (modules) $\{f_j\}$ in the numerical interval $[0, L)$, where $L = \prod_{j=1}^k f_j$ is overall amount of information code words [9]. In this case, the greatest common divisor of any two NPNSRC bases is equal to $(d_i, d_j) = 1$; $i, j = \overline{1, k}$ ($i \neq j$).

In order for the non-positional code structure in the NPNSRC to have the necessary corrective abilities, it is required that it contain sufficient information redundancy. First, the extant information redundancy in the original structure of the non-positional code should be determined and quantified [10]. Secondly, when tasked with providing data with additional corrective capabilities, introduce additional (artificial) information redundancy (apply the information redundancy method) by introducing additional (control) bases $\{f_c\}$ NPNSRCS [11].

Without loss of generality of reasoning, when tasked with providing data in the NPNSRCS with additional corrective capabilities, we will assume that only one additional control bases $f_c = f_{k+1}$ is added to the k information bases, which is coprime with any of the k existing information bases [4, 9]. In this case, the non-positional code structure $D = (d_1 \parallel d_2 \parallel \dots \parallel d_j \parallel \dots \parallel d_k \parallel d_{k+1})$ in the NPNSRC is represented by a set of $\{f_i\}$ ($i = \overline{1, k+1}$) bases in the full (working) numerical interval $[0, L_1)$, where $L_1 = L \cdot f_{k+1}$ is the overall amount of code words for this NPNSRC with one control base [9].

It is known [4, 12] that for non-positional code structure in the NPNSRC the minimum code distance is defined by the expression $V_{\min} = c + 1$, where c is the number of control bases used

in the non-positional code structure in the NPNSRC, i.e. minimum code distance depends both on the number of control bases and on the size of each of it.

If for the control bases $\{f_{z_j}\}$ the condition $\prod_{j=1}^g f_{z_j} \leq f_c$ is satisfied, then the introduction into the system of the NPNSRC bases of one control base $f_c = f_{k+1}$ is equivalent to the presence of g control bases $f_{z_1}, f_{z_2}, \dots, f_{z_g}$. Taking into account the fact that all numbers taking part in data processing in the CDPS, along with outcome of the operation are in the interval $[0, L)$, then it is clearly that if as outcome of data processing the final outcome D is obtained and at the same time $D \geq L$, this means that the resulting number $\tilde{D}$ is distorted (incorrect). Thus, if $D < L$, then the conclusion is that the number D is correct, and if $D \geq L$, then the number $\tilde{D}$ is incorrect. In this case, only solitary errors (only in one $\{d_j\}$ of the number D) are assumed, or a packet of errors no longer than $s = [\log_2(f_j - 1)] + 1$ binary digits in one residue modulo f_j .

All existing methods for monitoring data in the NPNSRC are based on this principle of comparing the value of the number D with the value $[0, L)$ of the information numerical interval. Note that the comparison principle is also used in the development of diagnostic and error correction methods. In the future, in this article, we will consider the method of operational (quick) diagnosis.

3. Control and diagnostics of data in the non-positional number system in residual classes

In [4] there are a number of scientific statements, the outcomes of the proof of which underlie for methods for controlling and diagnosing data errors presented in the NPNSRC. It should be reminded that in what follows only a solitary error is assumed (in one residue d_j ($j = \overline{1, k+1}$) of the number $D = (d_1 \| d_2 \| \dots \| d_j \| \dots \| d_k \| d_{k+1})$ presented in the NPNSRC).

Let the number $D = (d_1 \| d_2 \| \dots \| d_j \| \dots \| d_k \| d_{k+1})$ being checked be given in the NPNSRC with informational $\{f_j\}$ ($j = \overline{1, k}$) and one control base $f_c = f_{k+1}$. It is necessary, firstly, to control (determine the correctness) of the number D , and, secondly, to diagnose the residues $\{d_j\}$ ($j = \overline{1, k+1}$) of the number D , i.e. determine distorted (or undistorted) residues.

Data controlling and diagnostics are carried out sequentially in two stages.

First stage. Method for controlling (monitoring) data of non-positional code structure $D = (d_1 \| d_2 \| \dots \| d_{j-1} \| d_j \| d_{j+1} \| \dots \| d_k \| d_{k+1})$, which from the following algorithm of actions:

1. Determine the values of the orthogonal basis B_j ($j = \overline{1, k+1}$) for the complete system of bases (modules) $\{f_j\}$ NPNSRC:

$$B_j = \frac{e_j \cdot L_1}{f_j}, \quad (1)$$

where e_j is weight of the orthogonal basis B_j .

2. Using the system of orthogonal basis B_j , the original number D in the NPNSRC is represented in the PNS [13]:

$$D_{PNS} = \left(\sum_{j=1}^{k+1} d_j \cdot B_j \right) \bmod L_1. \quad (2)$$

3. Carry out positional comparison operations between the values of D_{PNS} and L . If the comparison result showed that $D_{PNS} < L$, then number D is correct. If $D_{PNS} \geq L$, then the

number $\tilde{D}$ is considered incorrect if only one of the residues $\{d_j\}$ of the number D is distorted.

Second stage. Method for diagnosing the residues $\{d_j\}$ ($j=1, \overline{k+1}$) of the code structure $\tilde{D}$ of data, based on the use of the obtained results of the following statement.

Statement. Let in an ordered ($f_j < f_{j+1}$) NPNSRC with k information and one control base $f_c = f_{k+1}$ the number $D = (d_1 \parallel d_2 \parallel \dots \parallel d_j \parallel \dots \parallel d_k \parallel d_{k+1})$ satisfy the following condition:

$$L = \frac{L_1}{f_{k+1}} = L_{k+1} < D < \bar{L}_i, \quad (3)$$

where $L = \prod_{j=1}^k f_j$ is overall amount of information code words in the NPNSRC (including only information bases);

$L_1 = \prod_{j=1}^{k+1} f_j$ is overall amount of all code words in the NPNSRC (including information bases and control base);

L_{k+1} is overall amount code words with one control base $f_c = f_{k+1}$;

$\bar{L}_i = \prod_{\substack{p=1, \\ p \neq i}}^{k+1} f_p$ is overall amount of code words for excluding the base f_i , that is

$$\bar{L}_i = f_1 \cdot f_2 \cdot \dots \cdot f_{i-1} \cdot f_{i+1} \cdot \dots \cdot f_{k+1}.$$

Then the residues of the number D are not distorted (correct) if only a solitary error (in one residue $\tilde{d}_j$) is possible. The second stage of the developed method will be considered in more detail using a specific example.

4. An example of the application of the control and operational diagnostics method

Let's consider an example of using the control and diagnostic method in the NPNSRC for a one-byte ($l = 1$) machine word (8 binary digits) CDPS. In this case, a complete NPNSRC with one control base is specified by information $f_1=3, f_2=4, f_3=5, f_4=7$ and control $f_c = f_{k+1} = f_5 = 11$ bases. At the same time, the requirements for unambiguous representation of code words in a given information numeric $[0, L)$ range are ensured.

For a given NPNSRC we can calculate: $L_1 = \prod_{j=1}^{k+1} f_j = f_1 \cdot f_2 \cdot f_3 \cdot f_4 \cdot f_{k+1} = 3 \cdot 4 \cdot 5 \cdot 7 \cdot 11 = 4620$

– overall amount of code words in this NPNSRC; $L = \prod_{j=1}^k f_j = f_1 \cdot f_2 \cdot f_3 \cdot f_4 = 3 \cdot 4 \cdot 5 \cdot 7 = 420$ –

overall amount of information code words in this NPNSRC. In this case, the full (working) $[0, L_1)$ and informational $[0, L)$ numerical ranges of numbers are defined, respectively, as $[0, 4620)$ and $[0, 420)$. All possible partial sets of the NPNSRC bases for a one-byte ($l = 1$)

CDPS are presented in Table 1, where $\bar{L}_i = \frac{L_1}{f_i}$, f_i is a base that is not included in the given complete system of the NPNSRC bases.

For example, consider the process of finding $\bar{L}_i$, since the complete system of bases of the considered NPNSRC consists of five bases: $f_1=3, f_2=4, f_3=5, f_4=7$ and $f_5 = 11$, the base that is not included in the first line ($i=1$) of Table 1 is $f_i=3$, so we divide overall amount of code

words in this NPNSRC $L_1 = 4620$ by this base: $\bar{L}_1 = \frac{4620}{3} = 1540$ or can get the same result by multiplying all sets of bases NPNSRC in the first line ($i = 1$) of Table 1: $\bar{L}_1 = 4 \cdot 5 \cdot 7 \cdot 11 = 1540$.

Table 1

Set of partial operating bases NPNSRC ($l = 1$) [14]

$i \backslash j$	f_1	f_2	f_3	f_4	$\bar{L}_i$
1	4	5	7	11	1540
2	3	5	7	11	1155
3	3	4	7	11	924
4	3	4	5	11	660
5	3	4	5	7	420

Let, in the process of data processing, in lieu the correct $D = (1 \parallel 0 \parallel 0 \parallel 2 \parallel 1)$ ($D_{PNS} = 100 < L = 420$) result of the operation a number of the form $\tilde{D} = (0 \parallel 0 \parallel 0 \parallel 2 \parallel 1)$, where $\tilde{D}_{PNS} = 3180 > L = 420$. It is necessary to verify the correctness of the number $\tilde{D}$ and diagnose its residues $\{d_j\}$ ($j = \overline{1,5}$).

First stage.

1. Let's define all possible orthogonal basis $B_j (j = \overline{1,5})$ using formula (1) for the complete system of bases $f_1 = 3, f_2 = 4, f_3 = 5, f_4 = 7$ and $f_5 = 11$ NPNSRC [14]:

$$\begin{cases} B_1 = (1, 0, 0, 0, 0) = 1540, e_1 = 1, \\ B_2 = (0, 1, 0, 0, 0) = 3465, e_2 = 3, \\ B_3 = (0, 0, 1, 0, 0) = 3696, e_3 = 4, \\ B_4 = (0, 0, 0, 1, 0) = 2640, e_4 = 4, \\ B_5 = (0, 0, 0, 0, 1) = 2520, e_5 = 6. \end{cases}$$

2. Using the values of orthogonal basis $B_j (j = \overline{1,5})$, let's determine the value of number $\tilde{D}$ in the NPNSRC is represented in the PNS according to formula (2):

$$\tilde{D}_{PNS} = (0 \cdot 1540 + 0 \cdot 3465 + 0 \cdot 3696 + 2 \cdot 2640 + 1 \cdot 2520) \bmod L_1 = 7800 \bmod 4620 = 3180.$$

3. Let's compare the obtained number of $\tilde{D}_{PNS}$ and the value $L = 420$. Since $\tilde{D}_{PNS} = 3180 > L = 420$, we make conclusion that the received $\tilde{D}$ is incorrect by $\{d_j\}$ of the correct number $D = (1 \parallel 0 \parallel 0 \parallel 2 \parallel 1)$.

Second stage.

1. Let's determine the values of partial orthogonal basis B_{ji} for each of the 5 possible sets of the NPNSRC bases [15]. In general, the value of partial orthogonal basis B_{ji} is determined based on the following comparison [16, 17]:

$$B_{ji} = \frac{\bar{L}_i \cdot e_{ji}}{f_j} \equiv 1 \pmod{f_j}, \quad (4)$$

where $e_{ji} \equiv \overline{1, f_j - 1}$ – weight of the orthogonal basis B_{ji} .

So, for $j = 4$ and $i = 5$ we have:

$$\begin{cases} B_{1i} = (1, 0, 0, 0), \\ B_{2i} = (0, 1, 0, 0), \\ B_{3i} = (0, 0, 1, 0), \\ B_{4i} = (0, 0, 0, 1). \end{cases}$$

Let's determine the values of B_{j1} for the first ($i=1$) set of bases: $f_1=4, f_2=5, f_3=7$ and $f_4=11$ (see Table 1). In this case $\bar{L}_1 = \prod_{j=1}^4 f_j = 4 \cdot 5 \cdot 7 \cdot 11 = 1540$ ($L=420$). We determine the values of the partial orthogonal basis based on the known relationship [18-20].

Determine the value of $B_{11} = \frac{\bar{L}_1 \cdot e_{11}}{f_1}$. In this case, we use $f_1=4$. In this case, $\frac{\bar{L}_1}{f_1} = \frac{1540}{4} = 385$, $e_{11} \equiv \overline{1, f_1 - 1} = \overline{1, 4 - 1} = \overline{1, 3}$. Let's compose possible values of B_{11} depending on possible e_{11} :

$$\begin{cases} 1 \cdot 385 \equiv 1(\text{mod } 4), \\ 2 \cdot 385 \equiv 2(\text{mod } 4), \\ 3 \cdot 385 \equiv 3(\text{mod } 4). \end{cases}$$

In this case, to fulfill the condition (4) we have that $B_{11} = 1 \cdot 385 = 385$.

Let's determine the value of $B_{21} = \frac{\bar{L}_1 \cdot e_{21}}{f_2}$. In this case $f_2=5$, $\frac{\bar{L}_1}{f_2} = \frac{1540}{5} = 308$, $e_{21} \equiv \overline{1, f_2 - 1} = \overline{1, 5 - 1} = \overline{1, 4}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 308 \equiv 3(\text{mod } 5), \\ 2 \cdot 308 \equiv 1(\text{mod } 5), \end{cases} \quad \begin{cases} 3 \cdot 308 \equiv 4(\text{mod } 5), \\ 4 \cdot 308 \equiv 2(\text{mod } 5). \end{cases}$$

In this case, $B_{21} = 2 \cdot 308 = 616$.

Let's determine the value of $B_{31} = \frac{\bar{L}_1 \cdot e_{31}}{f_3}$. In this case $f_3=7$, $\frac{\bar{L}_1}{f_3} = \frac{1540}{7} = 220$, $e_{31} \equiv \overline{1, f_3 - 1} = \overline{1, 7 - 1} = \overline{1, 6}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 220 \equiv 3(\text{mod } 7), \\ 2 \cdot 220 \equiv 6(\text{mod } 7), \\ 3 \cdot 220 \equiv 2(\text{mod } 7), \end{cases} \quad \begin{cases} 4 \cdot 220 \equiv 5(\text{mod } 7), \\ 5 \cdot 220 \equiv 1(\text{mod } 7), \\ 6 \cdot 220 \equiv 4(\text{mod } 7). \end{cases}$$

In this case, $B_{31} = 5 \cdot 220 = 1100$.

Let's determine the value of $B_{41} = \frac{\bar{L}_1 \cdot e_{41}}{f_4}$. In this case $f_4=11$, $\frac{\bar{L}_1}{f_4} = \frac{1540}{11} = 140$, $e_{41} \equiv \overline{1, f_4 - 1} = \overline{1, 11 - 1} = \overline{1, 10}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 140 \equiv 8(\text{mod } 11), \\ 2 \cdot 140 \equiv 5(\text{mod } 11), \\ 3 \cdot 140 \equiv 2(\text{mod } 11), \\ 4 \cdot 140 \equiv 10(\text{mod } 11), \\ 5 \cdot 140 \equiv 7(\text{mod } 11), \end{cases} \quad \begin{cases} 6 \cdot 140 \equiv 4(\text{mod } 11), \\ 7 \cdot 140 \equiv 1(\text{mod } 11), \\ 8 \cdot 140 \equiv 9(\text{mod } 11), \\ 9 \cdot 140 \equiv 6(\text{mod } 11), \\ 10 \cdot 140 \equiv 3(\text{mod } 11). \end{cases}$$

In this case, $B_{41} = 7 \cdot 140 = 980$.

Let's determine the values of B_{j2} for the second ($i=2$) set of bases: $f_1=3, f_2=5, f_3=7$ and $f_4=11$ (see Table 1). In this case $\bar{L}_2 = \prod_{j=1}^4 f_j = 3 \cdot 5 \cdot 7 \cdot 11 = 1155$ ($L=420$).

Let's determine the value of $B_{12} = \frac{\bar{L}_2 \cdot e_{12}}{f_1}$. In this case $f_1=3$, $\frac{\bar{L}_2}{f_1} = \frac{1155}{3} = 385$, $e_{12} \equiv \overline{1, f_1 - 1} = \overline{1, 3 - 1} = \overline{1, 2}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 385 \equiv 1(\text{mod } 3), \\ 2 \cdot 385 \equiv 2(\text{mod } 3). \end{cases}$$

In this case, $B_{12} = 1 \cdot 385 = 385$.

Let's determine the value of $B_{22} = \frac{\bar{L}_2 \cdot e_{22}}{f_2}$. In this case $f_2 = 5$, $\frac{\bar{L}_2}{f_2} = \frac{1155}{5} = 231$, $e_{22} \equiv \overline{1, f_2 - 1} = \overline{1, 5 - 1} = \overline{1, 4}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 231 \equiv 1(\text{mod } 5), \\ 2 \cdot 231 \equiv 2(\text{mod } 5), \end{cases} \quad \begin{cases} 3 \cdot 231 \equiv 3(\text{mod } 5), \\ 4 \cdot 231 \equiv 4(\text{mod } 5). \end{cases}$$

In this case, $B_{22} = 1 \cdot 231 = 231$.

Let's determine the value of $B_{32} = \frac{\bar{L}_2 \cdot e_{32}}{f_3}$. In this case $f_3 = 7$, $\frac{\bar{L}_2}{f_3} = \frac{1155}{7} = 165$, $e_{32} \equiv \overline{1, f_3 - 1} = \overline{1, 7 - 1} = \overline{1, 6}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 165 \equiv 4(\text{mod } 7), \\ 2 \cdot 165 \equiv 1(\text{mod } 7), \\ 3 \cdot 165 \equiv 5(\text{mod } 7), \end{cases} \quad \begin{cases} 4 \cdot 165 \equiv 2(\text{mod } 7), \\ 5 \cdot 165 \equiv 6(\text{mod } 7), \\ 6 \cdot 165 \equiv 3(\text{mod } 7). \end{cases}$$

In this case, $B_{32} = 2 \cdot 165 = 330$.

Let's determine the value of $B_{42} = \frac{\bar{L}_2 \cdot e_{42}}{f_4}$. In this case $f_4 = 11$, $\frac{\bar{L}_2}{f_4} = \frac{1155}{11} = 105$, $e_{42} \equiv \overline{1, f_4 - 1} = \overline{1, 11 - 1} = \overline{1, 10}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 105 \equiv 6(\text{mod } 11), \\ 2 \cdot 105 \equiv 1(\text{mod } 11), \\ 3 \cdot 105 \equiv 7(\text{mod } 11), \\ 4 \cdot 105 \equiv 2(\text{mod } 11), \\ 5 \cdot 105 \equiv 8(\text{mod } 11), \end{cases} \quad \begin{cases} 6 \cdot 105 \equiv 3(\text{mod } 11), \\ 7 \cdot 105 \equiv 9(\text{mod } 11), \\ 8 \cdot 105 \equiv 4(\text{mod } 11), \\ 9 \cdot 105 \equiv 10(\text{mod } 11), \\ 10 \cdot 105 \equiv 5(\text{mod } 11). \end{cases}$$

In this case, $B_{42} = 2 \cdot 105 = 210$.

Let's determine the values of B_{j3} for the third ($i=3$) set of bases: $f_1=3$, $f_2=4$, $f_3=7$ and $f_4=11$ (see Table 1). In this case $\bar{L}_3 = \prod_{j=1}^4 f_j = 3 \cdot 4 \cdot 7 \cdot 11 = 924$ ($L = 420$).

Let's determine the value of $B_{13} = \frac{\bar{L}_3 \cdot e_{13}}{f_1}$. In this case $f_1 = 3$, $\frac{\bar{L}_3}{f_1} = \frac{924}{3} = 308$, $e_{13} \equiv \overline{1, f_1 - 1} = \overline{1, 3 - 1} = \overline{1, 2}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 308 \equiv 2(\text{mod } 3), \\ 2 \cdot 308 \equiv 1(\text{mod } 3). \end{cases}$$

In this case, $B_{13} = 2 \cdot 308 = 616$.

Let's determine the value of $B_{23} = \frac{\bar{L}_3 \cdot e_{23}}{f_2}$. In this case $f_2 = 4$, $\frac{\bar{L}_3}{f_2} = \frac{924}{4} = 231$, $e_{23} \equiv \overline{1, f_2 - 1} = \overline{1, 4 - 1} = \overline{1, 3}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 231 \equiv 3(\text{mod } 4), \\ 2 \cdot 231 \equiv 2(\text{mod } 4), \\ 3 \cdot 231 \equiv 1(\text{mod } 4). \end{cases}$$

In this case, $B_{23} = 3 \cdot 231 = 693$.

Let's determine the value of $B_{33} = \frac{\bar{L}_3 \cdot e_{33}}{f_3}$. In this case $f_3 = 7$, $\frac{\bar{L}_3}{f_3} = \frac{924}{7} = 132$, $e_{33} \equiv \overline{1, f_3 - 1} = \overline{1, 7 - 1} = \overline{1, 6}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 132 \equiv 6(\text{mod } 7), \\ 2 \cdot 132 \equiv 5(\text{mod } 7), \\ 3 \cdot 132 \equiv 4(\text{mod } 7), \end{cases} \quad \begin{cases} 4 \cdot 132 \equiv 3(\text{mod } 7), \\ 5 \cdot 132 \equiv 2(\text{mod } 7), \\ 6 \cdot 132 \equiv 1(\text{mod } 7). \end{cases}$$

In this case, $B_{33} = 6 \cdot 132 = 792$.

Let's determine the value of $B_{43} = \frac{\bar{L}_2 \cdot e_{43}}{f_4}$. In this case $f_4 = 11$, $\frac{\bar{L}_2}{f_4} = \frac{924}{11} = 84$,

$e_{43} \equiv \overline{1, f_4 - 1} = \overline{1, 11 - 1} = \overline{1, 10}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 84 \equiv 7(\text{mod } 11), \\ 2 \cdot 84 \equiv 3(\text{mod } 11), \\ 3 \cdot 84 \equiv 10(\text{mod } 11), \\ 4 \cdot 84 \equiv 6(\text{mod } 11), \\ 5 \cdot 84 \equiv 3(\text{mod } 11), \end{cases} \quad \begin{cases} 6 \cdot 84 \equiv 9(\text{mod } 11), \\ 7 \cdot 84 \equiv 5(\text{mod } 11), \\ 8 \cdot 84 \equiv 1(\text{mod } 11), \\ 9 \cdot 84 \equiv 8(\text{mod } 11), \\ 10 \cdot 84 \equiv 4(\text{mod } 11). \end{cases}$$

In this case, $B_{43} = 8 \cdot 84 = 672$.

Let's determine the values of B_{j4} for the fourth ($i = 4$) set of bases: $f_1 = 3$, $f_2 = 4$, $f_3 = 5$ and

$f_4 = 11$ (see Table 1). In this case $\bar{L}_4 = \prod_{j=1}^4 f_j = 3 \cdot 4 \cdot 5 \cdot 11 = 660$ ($L = 420$).

Let's determine the value of $B_{14} = \frac{\bar{L}_4 \cdot e_{14}}{f_1}$. In this case $f_1 = 3$, $\frac{\bar{L}_4}{f_1} = \frac{660}{3} = 220$,

$e_{14} \equiv \overline{1, f_1 - 1} = \overline{1, 3 - 1} = \overline{1, 2}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 220 \equiv 1(\text{mod } 3), \\ 2 \cdot 220 \equiv 2(\text{mod } 3). \end{cases}$$

In this case, $B_{14} = 1 \cdot 220 = 220$.

Let's determine the value of $B_{24} = \frac{\bar{L}_4 \cdot e_{24}}{f_2}$. In this case $f_2 = 4$, $\frac{\bar{L}_4}{f_2} = \frac{660}{4} = 165$,

$e_{24} \equiv \overline{1, f_2 - 1} = \overline{1, 4 - 1} = \overline{1, 3}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 165 \equiv 1(\text{mod } 4), \\ 2 \cdot 165 \equiv 2(\text{mod } 4), \\ 3 \cdot 165 \equiv 3(\text{mod } 4). \end{cases}$$

In this case, $B_{24} = 1 \cdot 165 = 165$.

Let's determine the value of $B_{34} = \frac{\bar{L}_4 \cdot e_{34}}{f_3}$. In this case $f_3 = 5$, $\frac{\bar{L}_4}{f_3} = \frac{660}{5} = 132$,

$e_{34} \equiv \overline{1, f_3 - 1} = \overline{1, 5 - 1} = \overline{1, 4}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 132 \equiv 2(\text{mod } 5), \\ 2 \cdot 132 \equiv 4(\text{mod } 5), \end{cases} \quad \begin{cases} 3 \cdot 132 \equiv 1(\text{mod } 5), \\ 4 \cdot 132 \equiv 3(\text{mod } 5). \end{cases}$$

In this case, $B_{34} = 3 \cdot 132 = 396$.

Let's determine the value of $B_{44} = \frac{\bar{L}_4 \cdot e_{44}}{f_4}$. In this case $f_4 = 11$, $\frac{\bar{L}_4}{f_4} = \frac{660}{11} = 60$,

$e_{44} \equiv \overline{1, f_4 - 1} = \overline{1, 11 - 1} = \overline{1, 10}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 60 \equiv 5(\text{mod } 11), \\ 2 \cdot 60 \equiv 10(\text{mod } 11), \\ 3 \cdot 60 \equiv 4(\text{mod } 11), \\ 4 \cdot 60 \equiv 9(\text{mod } 11), \\ 5 \cdot 60 \equiv 3(\text{mod } 11), \end{cases} \quad \begin{cases} 6 \cdot 60 \equiv 8(\text{mod } 11), \\ 7 \cdot 60 \equiv 2(\text{mod } 11), \\ 8 \cdot 60 \equiv 7(\text{mod } 11), \\ 9 \cdot 60 \equiv 1(\text{mod } 11), \\ 10 \cdot 60 \equiv 6(\text{mod } 11). \end{cases}$$

In this case, $B_{44} = 9 \cdot 60 = 540$.

Let's determine the values of B_{j5} for the fifth ($i = 5$) set of bases: $f_1 = 3$, $f_2 = 4$, $f_3 = 5$ and

$f_4 = 7$ (see Table 1). In this case $\bar{L}_5 = \prod_{j=1}^4 f_j = 3 \cdot 4 \cdot 5 \cdot 7 = 420$ ($L = 420$).

Let's determine the value of $B_{15} = \frac{\bar{L}_5 \cdot e_{15}}{f_1}$. In this case $f_1 = 3$, $\frac{\bar{L}_5}{f_1} = \frac{420}{3} = 140$, $e_{15} \equiv \overline{1, f_1 - 1} = \overline{1, 3 - 1} = \overline{1, 2}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 140 \equiv 2(\text{mod } 3), \\ 2 \cdot 140 \equiv 1(\text{mod } 3). \end{cases}$$

In this case, $B_{15} = 2 \cdot 140 = 280$.

Let's determine the value of $B_{25} = \frac{\bar{L}_5 \cdot e_{25}}{f_2}$. In this case $f_2 = 4$, $\frac{\bar{L}_5}{f_2} = \frac{420}{4} = 105$, $e_{25} \equiv \overline{1, f_2 - 1} = \overline{1, 4 - 1} = \overline{1, 3}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 105 \equiv 1(\text{mod } 4), \\ 2 \cdot 105 \equiv 2(\text{mod } 4), \\ 3 \cdot 105 \equiv 3(\text{mod } 4). \end{cases}$$

In this case, $B_{25} = 1 \cdot 105 = 105$.

Let's determine the value of $B_{35} = \frac{\bar{L}_5 \cdot e_{35}}{f_3}$. In this case $f_3 = 5$, $\frac{\bar{L}_5}{f_3} = \frac{420}{5} = 84$, $e_{35} \equiv \overline{1, f_3 - 1} = \overline{1, 5 - 1} = \overline{1, 4}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 84 \equiv 4(\text{mod } 5), \\ 2 \cdot 84 \equiv 3(\text{mod } 5), \end{cases} \quad \begin{cases} 3 \cdot 84 \equiv 2(\text{mod } 5), \\ 4 \cdot 84 \equiv 1(\text{mod } 5). \end{cases}$$

In this case, $B_{35} = 4 \cdot 84 = 336$.

Let's determine the value of $B_{45} = \frac{\bar{L}_5 \cdot e_{45}}{f_4}$. In this case $f_4 = 7$, $\frac{\bar{L}_5}{f_4} = \frac{420}{7} = 60$, $e_{45} \equiv \overline{1, f_4 - 1} = \overline{1, 7 - 1} = \overline{1, 6}$. Let's make a set of comparisons:

$$\begin{cases} 1 \cdot 60 \equiv 4(\text{mod } 7), \\ 2 \cdot 60 \equiv 1(\text{mod } 7), \\ 3 \cdot 60 \equiv 5(\text{mod } 7), \end{cases} \quad \begin{cases} 4 \cdot 60 \equiv 2(\text{mod } 7), \\ 5 \cdot 60 \equiv 6(\text{mod } 7), \\ 6 \cdot 60 \equiv 3(\text{mod } 7). \end{cases}$$

In this case, $B_{45} = 2 \cdot 60 = 120$.

The set of calculated partial orthogonal basis B_{ji} is given in Table 2.

Table 2
Partial orthogonal basis B_{ji} for $i = 1$ [14]

$i \backslash j$	1	2	3	4
1	385	616	1100	980
2	385	231	330	210
3	616	693	792	672
4	220	165	396	540
5	280	105	336	120

2. Let's determine the correctness of the residues of the number $\tilde{D}$. First, let's compose all possible projections $\tilde{D}_i$ of the number $\tilde{D} = (0 \| 0 \| 0 \| 2 \| 1)$:

$$\begin{cases} \tilde{D}_1 = (0, 0, 2, 1), \\ \tilde{D}_2 = (0, 0, 2, 1), \\ \tilde{D}_3 = (0, 0, 2, 1), \\ \tilde{D}_4 = (0, 0, 0, 1), \\ \tilde{D}_5 = (0, 0, 0, 2). \end{cases}$$

3. Let's represent the values of the projections $\tilde{D}_i$ in the PNS [14]:

$$\begin{aligned}\tilde{D}_{1PNS} &= (d_1 \cdot B_{11} + d_2 \cdot B_{21} + d_3 \cdot B_{31} + d_4 \cdot B_{41}) \bmod \bar{L}_1 = \\ &= (0 \cdot 385 + 0 \cdot 616 + 2 \cdot 1100 + 1 \cdot 980) \bmod 1540 = 100 < 420.\end{aligned}$$

$$\begin{aligned}\tilde{D}_{2PNS} &= (d_1 \cdot B_{12} + d_2 \cdot B_{22} + d_3 \cdot B_{32} + d_4 \cdot B_{42}) \bmod \bar{L}_2 = \\ &= (0 \cdot 385 + 0 \cdot 231 + 2 \cdot 330 + 1 \cdot 210) \bmod 1155 = 870 > 420.\end{aligned}$$

$$\begin{aligned}\tilde{D}_{3PNS} &= (d_1 \cdot B_{13} + d_2 \cdot B_{23} + d_3 \cdot B_{33} + d_4 \cdot B_{43}) \bmod \bar{L}_3 = \\ &= (0 \cdot 616 + 0 \cdot 693 + 2 \cdot 792 + 1 \cdot 672) \bmod 924 = 418 < 420.\end{aligned}$$

$$\begin{aligned}\tilde{D}_{4PNS} &= (d_1 \cdot B_{14} + d_2 \cdot B_{24} + d_3 \cdot B_{34} + d_4 \cdot B_{44}) \bmod \bar{L}_4 = \\ &= (0 \cdot 220 + 0 \cdot 165 + 2 \cdot 396 + 1 \cdot 540) \bmod 660 = 540 > 420.\end{aligned}$$

$$\begin{aligned}\tilde{D}_{5PNS} &= (d_1 \cdot B_{15} + d_2 \cdot B_{25} + d_3 \cdot B_{35} + d_4 \cdot B_{45}) \bmod \bar{L}_5 = \\ &= (0 \cdot 280 + 0 \cdot 105 + 2 \cdot 336 + 1 \cdot 120) \bmod 420 = 240 < 420.\end{aligned}$$

Thus, of all the received projections $\tilde{D}_i$ of the number $\tilde{D} = (0\|0\|0\|2\|1)$ the projections $\tilde{D}_1, \tilde{D}_3, \tilde{D}_5 < L = 420$ and the projections $\tilde{D}_2, \tilde{D}_4 > L = 420$. Therefore, the results of controlling and diagnosing the incorrect $\tilde{D}$ number state that among all five residues of the number it is the residues d_1, d_3 and d_5 may be erroneous, but the residues d_2 and d_4 are definitely not distorted.

5. Conclusions

This article improves the method of control and operational diagnostics of data errors presented in the NPNSRC, using the orthogonal basis B_{ji} of partial sets of bases (modules) NPNSRC. The values of orthogonal basis B_{ji} are formed from the complete system of bases $\{f_i\}$ ($i = \overline{1, k+1}$), and its usage provides an opportunity to organize the parallel processing of projections $\tilde{D}_i$ of numbers D of a non-positional code structure in the NPNSRC. This circumstance makes it possible to enhance efficiency of controlling and diagnosing data in the CDPS operating in the NPNSRC.

Examples of specific realization of the process of controlling (monitoring) and diagnosing errors are given. It is established that the use of the proposed method will improve the efficiency of controlling and diagnosing data errors in the data processing system operating in the NPNSRC, and will also technically simplify the procedure for processing non-positional code structures.

References

- [1] R. Wilmhof, F. Taylor, On hard errors in RNS architectures, IEEE Transactions on Acoustics, Speech, and Signal Processing 31 (1983) 772–774. doi:10.1109/TASSP.1983.1164105.
- [2] H. Krishna, K.-Y. Lin, J.-D. Sun, A coding theory approach to error control in redundant residue number systems. I. Theory and single error correction, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 39 (1992) 8–17. doi:10.1109/82.204106.
- [3] Lie-Liang Yang, L. Hanzo, Redundant residue number system based error correction codes, in: IEEE 54th Vehicular Technology Conference, Atlantic City, NJ, 2001, pp. 1472–1476. doi:10.1109/VTC.2001.956442.
- [4] I.Ya. Akushskii, D.I. Yuditskii, Machine Arithmetic in Residual Classes, Sov. Radio, Moscow, 1968.
- [5] V. Krasnobayev, A. Yanko, D. Kovalchuk, Control, An Improved Method for Performing the Arithmetic Operations of Modulo Addition of the Remainders of Numbers, in: 2023 13th

- International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 2023, pp. 1–6. doi:10.1109/DESSERT61349.2023.10416508.
- [6] G. Sobczyk, *Modular Number Systems, New Foundations in Mathematics*, Birkhäuser, Boston, MA, 2013, pp. 1–21. doi:10.1007/978-0-8176-8385-6_1.
 - [7] V. Yatskiv, T. Tsavolyk, Two-dimensional corrective codes based on modular arithmetic, in: *The Experience of Designing and Application of CAD Systems in Microelectronics*, Lviv, 2015, pp. 291–294. doi:10.1109/CADSM.2015.7230860.
 - [8] N.I. Chervyakov, A.V. Veligosh, K.T. Tyncherov, S. A. Velikikh, Use of modular coding for high-speed digital filter design, *Cybernetics and Systems Analysis* 34 (1998) 254–260. doi:10.1007/BF02742075.
 - [9] V. Krasnobayev, A. Yanko, D. Kovalchuk, Control, Diagnostics and Error Correction in the Modular Number System, in: *The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS 2023)*, Zaporizhzhia, Ukraine, 2023, pp. 199–213. doi:10.32782/cmisi/3392-17.
 - [10] P. V. A. Mohan, *Residue Number Systems: Theory and Applications*, Birkhäuser Basel, Switzerland, 2016.
 - [11] R. Liu, L. Li, Y. Yang, Performance Residual Based Fault Detection for Feedback Control Systems, *IEEE Transactions on Circuits and Systems II: Express Briefs* 68 (2021) 3291–3295. doi:10.1109/TCSII.2021.3062718.
 - [12] S. Onyshchenko, A. Yanko, A. Hlushko, Improving the efficiency of diagnosing errors in computer devices for processing economic data functioning in the class of residuals, *Eastern-European Journal of Enterprise Technologies* 5, 4 (125) 63–73. doi:https://doi.org/10.15587/1729-4061.2023.289185.
 - [13] M. Zaharchenko, M. Hadzhyiev, A. Nazarenko, N. Salmanov, N. Shvets, Comparison of Positional and Timer Coding in the System of Residual Classes, in: *2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, pp. 414–418. doi:10.1109/TCSET55632.2022.9766866.
 - [14] A. Moskalenko, V. Krasnobayev, S. Koshman, The concept of diagnostic data errors of computing systems witch functioning in the system of residue classes, *Computer science and cybersecurity (CS&CS)* 2 (6) (2017) 39–46. URL: <https://periodicals.karazin.ua/cscs/article/download/8733/9891>.
 - [15] V. Krasnobayev, A. Yanko, A. Hlushko, Information Security of the National Economy Based on an Effective Data Control Method, *Journal of International Commerce, Economics and Policy* 14 (3) (2023). doi:10.1142/S1793993323500217.
 - [16] J. James, A. Pe, A novel method for error correction using Redundant Residue Number System in digital communication systems, in: *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, 2015, pp. 1793–1798. doi:10.1109/ICACCI.2015.7275875.
 - [17] Z. Wang, E. Zheng, J. Liu, T. Guo, Adaptive Orthogonal Basis Function Detection Method for Unknown Magnetic Target Motion State, *Applied Sciences* 14 (2):902 (2024). doi:10.3390/app14020902.
 - [18] O. Laktionov, L. Lievi, A. Tretiak, M. Movin, Investigation of combined ensemble methods for diagnostics of the quality of interaction of human-machine systems, *Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu* 4 (2023) 138–143. doi:10.33271/nvngu/2023-4/138.
 - [19] L. Mu, X. Liu and L.-L. Yang, Redundant Residue Number System Coded Diffusive Molecular Communications, in: *10th International Conference on Wireless Communications and Signal Processing (WCSP)*, Hangzhou, China, 2018, pp. 1–6, doi:10.1109/WCSP.2018.8555582.
 - [20] Y. Jia, S. Wang, J. Jin, H. Long, Hang, Adaptive Orthogonal Basis Scheme for OTFS, in: *Proceedings of the 17th International Conference on Communications and Networking*, volume 500 of *Lecture Notes of the Institute for Computer Sciences*, Springer, Switzerland, 2023. doi:10.1007/978-3-031-34790-0_5.

Algorithm for Assessing the Degree of Information Security Risk of a Cyber Physical System for Controlling Underground Metal Constructions

Volodymyr Yuzevych ^{1,2}, Anatoliy Obshta ², Ivan Opirskyy ² and Oleh Harasymchuk ²

¹ Lviv Karpenko Physico-mechanical Institute of the NAS of Ukraine, 5 Naukova str., Lviv, 79060, Ukraine

² Lviv Polytechnic National University, 12 Stepana Bandery str., Lviv, 79013, Ukraine

Abstract

A procedure for monitoring the functioning of the "underground metal structure (UMC) - aggressive environment (AE)" system has been developed, taking into account the methods of monitoring information on the detection and localisation of product leaks, methods for assessing the life of the corresponding cyber-physical system, as well as optimisation criteria related to the life of metal structural elements, risk assessment units and cryptographic information security system.

A system for assessing the value of information for the procedure of integral diagnosis of the CPS cyber-physical system has been developed, taking into account the division of the components of the value of the diagnosis D into two parts, where one part corresponds to incidents and the other to vulnerabilities, taking into account the Vaadin, Spring, AnyLogic, OptQuest frameworks

A compromise function has been developed that can be used to ensure the functioning of a cyber-physical system with given values of risk, the value of information about incidents, the value of information related to CPS vulnerabilities, as well as the strength of structural elements and parameters characterizing the quality of the cryptography algorithm.

The proposed approach uses elements of graph theory and takes into account the interdependencies of vulnerabilities. The result obtained with the help of criterion ratios makes it possible to: propose a new methodology for assessing the degree of risk of information security of CFS, taking into account the index of probability of a successful attack on the system and the index of adjustment, which provides feedback. Some results have been obtained: modelling of the main processes and links in the CFS; a procedure for identifying vulnerabilities in the physical space (at the sensor level), in cyberspace, and in the communication environment; an algorithm for calculating the degree of information security risk for the CFS using the neural network method and taking into account the quality functionality and quality criterion.

Keywords

Underground metal structures (UMC), cyber-physical system (CFS), optimization algorithm, quality criterion, neural networks, risk level, information security, protection system, sensors, communication environment.

1. Introduction

In the process of a comprehensive analysis of the functioning of the underground metal structures (UMC) system, it is necessary to take into account modern technologies for selecting, storing and processing information obtained by the cyber-physical system (CFS) in the process of control and the corresponding information security risks related to corrosion defects. In this context, the research subject is the procedure for assessing the level of information security risk of a cyber-physical system (CFS) for underground metal structures (UMC), which is implemented in the form of an approach that considers the interdependence of vulnerabilities.

This work aims to develop a methodology for assessing the functioning of the underground metal structures system and the degree of information security risk of the Cyber Physical System (CFS), which would ensure the reliability, confidentiality, completeness, value, and authenticity

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ yuzevych@ukr.net (V. Yuzevych); obshta2002@gmail.com (A. Obshta); iopirsky@gmail.com (I. Opirskyy), garasymchuk@ukr.net (O. Harasymchuk),

ORCID 0000-0001-5244-1850 (V. Yuzevych); 0000-0001-5151-312X (A. Obshta); 0000-0002-8461-8996 (J. X. Ceur), 0000-0002-8742-8872 (O. Harasymchuk),



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

of information, as well as reduce the ambiguity of determining informative parameters. By the goal, the following tasks need to be performed:

1a) to develop an approach to the design of a monitoring complex for the underground metal structure (UMC) - aggressive environment (AE) system, taking into account information monitoring methods, methods for assessing the life of the relevant cyber-physical system, and optimisation criteria;

2b) to develop a system for assessing the value of information for the procedure of integral diagnosis of the CPS cyber-physical system, taking into account the division of the components of the value of the diagnosis D into two parts, where one part C_{Inz} (D) corresponds to incidents, and the second C_{Vnz} (D) - to vulnerabilities:

3c) to develop a compromise function that can be used to ensure the functioning of a cyber-physical system with the specified values of risk, the value of information about incidents, the value of information about vulnerabilities related to CPS vulnerabilities, as well as the strength of structural elements and a high-quality cryptography algorithm.

UMCs are often subject to terrorist attacks and vandalism [1], which can lead to environmental problems of environmental pollution, hydrocarbon decomposition, and significant economic losses. It is necessary to control (monitor) UMCs, but the relevant process requires significant material costs and is also often dangerous. Therefore, it is advisable to use remote wireless monitoring of the UMC, taking into account the threshold values of informative parameters, in particular: temperature, relative humidity pressure around the UMC, dew point of the environment, carbon monoxide volume, number of liquefied petroleum gas leaks, human movement around the facility, fire and smoke [2]. There are safety regulations and documents, safety guidelines, and industry standards and regulations, compliance with which increases the likelihood of preventing system security failures, the ability to protect the engineering facility from an incident, and ensures the integrity of UMC structural elements [3].

Wireless sensor networks, together with microcontrollers, sensor devices, and communication interfaces, allow users to measure and select the necessary information, as well as respond to phenomena in the monitored UMC environment [4].

The relevance of UMC system risk research is related to the issues of hardware and software information protection. In this context, two important factors should be noted. Firstly, the risks of CFS hardware components are related to the fact that these components are not always certified and this affects the energy consumption and performance in the final version [5]. Secondly, fuzzy logic and neural networks (NNs) are applied to the risks of UMC software, as the combination of these two methods provides the highest efficiency and adaptability to non-numerical data [6]. It should be noted that the relevance of this type of research is also because IT risk may also be the risk of loss that originates from computer software malfunction, such as a manufacturer's software license expiration or glitches, and the ways it affects corporate activities [7, 8].

2. A model for diagnosing underground metal constructions concerning the quality criterion

We will build a diagram of the process of diagnosing an underground metal structure taking into account the quality criterion in the BPWin Process Modeller program using the IDEF0 module. Information about BPWin Process Modeller and the IDEF0 module is given in [9, 10]. The corresponding model diagram is shown in Fig. 1, which displays information for monitoring underground metal structures.

The input to the monitoring system (Fig. 1) is the information denoted by the set of parameters $M(P)$ and obtained using sensors for UMC [11-16]: electric current density, voltage, polarisation potential, soil moisture, and temperature. To verify the reliability of the input parameters, we use regulatory documents (standards, regulations, instructions). To organise the information on $M(P)$ for UMC, we use databases, knowledge bases, and appropriate algorithms (Fig. 1).

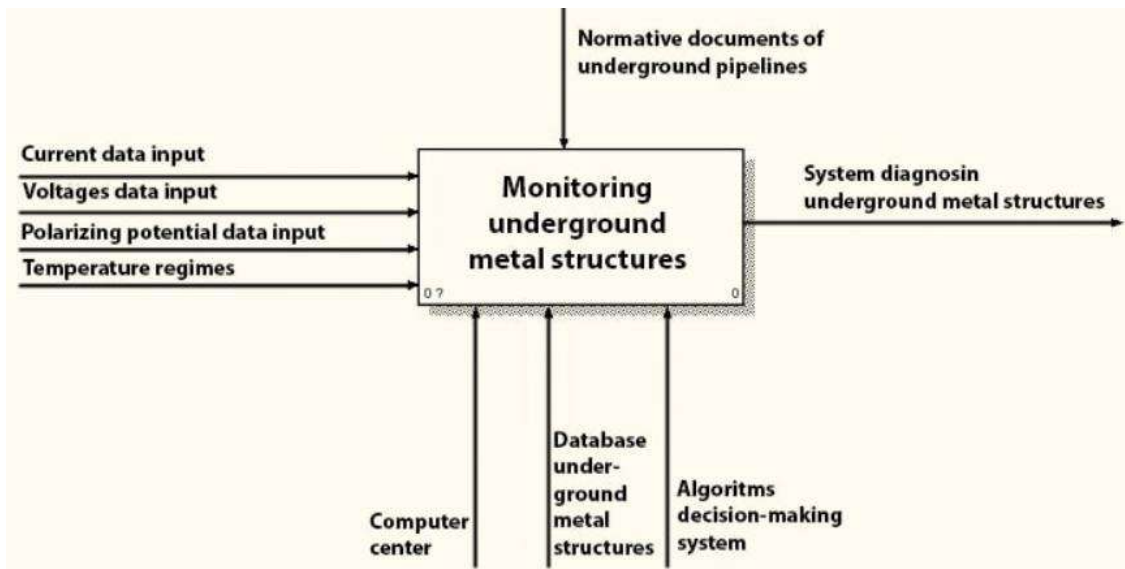


Figure 1: Monitoring system for underground metal constructions

The decomposition scheme of the model presented in Fig. 1 is shown in Fig. 2. To implement the model decomposition procedure, we used the relevant principles similarly to those in [11, 12, 17].

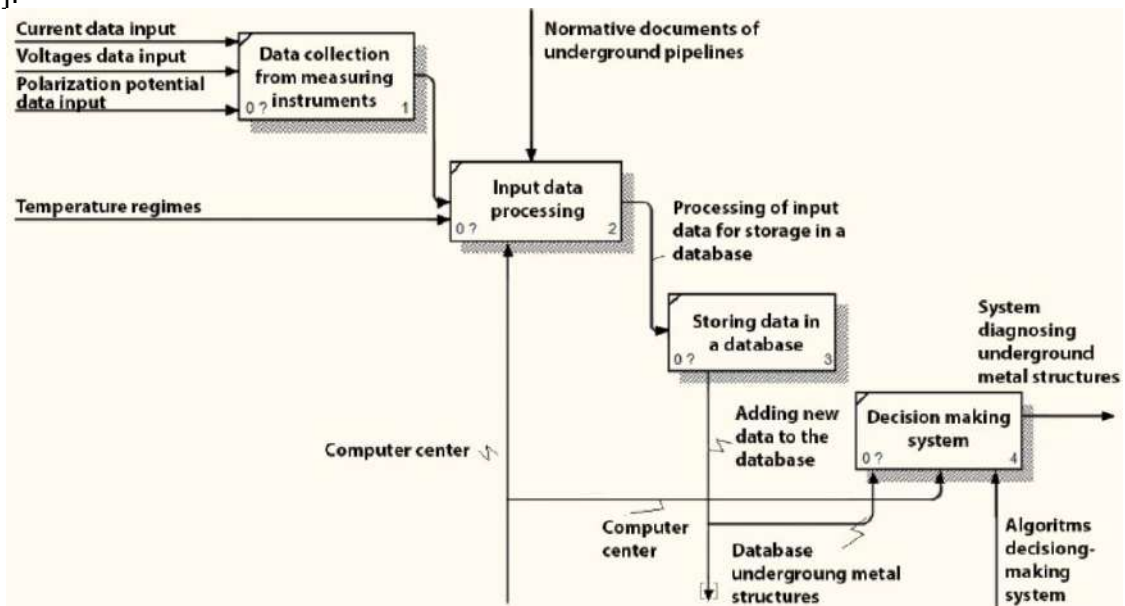


Figure 2: Decomposition monitoring model for underground metal constructions

Decomposition is a scientific method that uses the structure of a problem and allows replacing the solution of one large problem with a series of smaller problems, taking into account data availability, clarity, and the level of model complexity [17]. To organize the information about $M(P)$ for UMC in the decomposition process, we use databases, knowledge bases, as well as strength and quality criteria for each individual task. The complexity and categoricity of decomposition tasks are assessed by seven indicators of the component tasks similarly to [17]. This allows us to specify the quantitative description of the model, taking into account clear criteria and the peculiarities of the fact that structures are located underground. Such specification reduces the uncertainty and ambiguity of qualitative judgments, facilitating the assessment of the tasks of the model elements. The appropriate distribution of the categories of tasks of the UMC monitoring model is based on the experience of the authors of the scientific article [17] and the experience of the authors of this publication.

The block diagram of the new system, which corresponds to the technology of monitoring information on electrophysical parameters for underground metal structures with regard to the aggressive environment, is shown in Fig. 3.

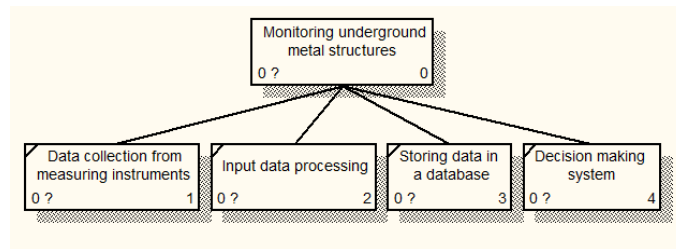


Figure 3: Block diagram of information monitoring for the system “underground metal construction (UMS) – aggressive environment (AE)”

Monitoring of the "UMC - Aggressive Environment (AE)" system details information on data acquisition from measuring devices (currents, voltages, soil moisture, temperature), processing of input data, data storage in the database, and decision-making in the context of optimizing informative parameters and physical characteristics of contacting media (soil, dielectric coating, metal).

A block diagram of the cyber-physical system (CPS) and the corresponding decision-making system for the operation of the CPS is shown in Fig. 4.

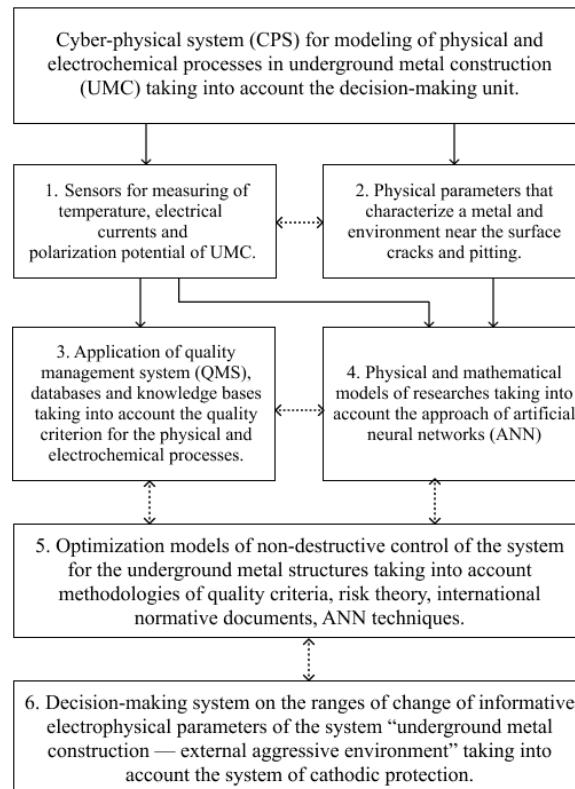


Figure 4: Block diagram of a cyber-physical system (CPS) for modelling the electrophysical parameters of the system "underground metal structure – external aggressive environment"

The block diagram in Fig. 4. contains 6 blocks, of which blocks 1-2 characterize the information and measurement system (IMS) for measuring electrophysical parameters. Blocks 3 and 4 are the basis of the information processing system for the CPS, obtained by sensors and non-destructive testing devices. Units 3 and 4 are used to organize and clarify the information received by the information monitoring system (Fig. 3). The methods of functioning of blocks 1-4, taking into account the methods of using neural networks, are described in detail in articles [13-16].

We will introduce an integral indicator of the effectiveness of the ER functioning of the underground pipeline monitoring system similar to [17] with the addition of the parameters of the quality management system (QMS) [15, 16] and the cryptographic information protection system [20, 21]:

$$ER = f(F(R), M(P), F(Z), F(Q), F(It, Pw)) \Rightarrow \text{opt} \quad (1)$$

Here $F(R)$ is the effectiveness of the monitoring system and the corresponding algorithm in the context of the risk-based corrosion model R ; $F(Z)$ is the effectiveness of the information security structural units; $F(Q)$ is the effectiveness of the algorithm taking into account the QMS; $F(It, Pw)$ is a function of performance and the effectiveness of the algorithm in the context of personnel functioning; It is the index of creativity, qualification and loyalty of employees; Pw is a set of parameters that characterizes the behaviour of personnel, including qualitative and quantitative factors Pw_i . ($i = 1, 2, \dots, n_s$; n_s is the total number of parameters of the corresponding model).

To optimize the information flows $J_n(M(P), Pw_i)$ in the cyber-physical system monitoring system and improve the corrosion protection system of underground metal structures, we use the quality functionality with feedback similar to [14]:

$$\Psi(J_n(M(P), Pw_i), R, FB(M(P), Pw_i)) = \int_{t_0}^{t_k} f(\bar{y}, \bar{u}, \bar{\xi}) dt \Rightarrow \text{opt} \quad (2)$$

where $\bar{y}$ - vector of specified influences ($y_j(t)$ - components of the vector, $j = 1, 2, \dots, n_t$); $\bar{u}$ - vector of controls; $\bar{\xi}$ - vector of uncertain disturbances; $[t_0, t_k]$ - time interval in which the process is considered (formation of optimal values of information and financial flows $J_n(Pw_i)$; $n = 1, 2, \dots, m$; $i = 1, 2, \dots, n_s$; m - total number of parameters); $f(\bar{y}, \bar{u}, \bar{\xi})$ - a function that reflects the quality indicator; $FB(M(P), Pw_i)$ - a function that characterizes the feedback between the flows $J_n(M(P), Pw_i)$, taking into account the regulatory documents on information security and expert opinions. Here, the symbol opt corresponds to the condition of optimality of the functional (2).

It is worth noting that relation (2) is written in the form of a quality criterion for the corresponding cyber-physical system (CPS). It is worth noting that the inverse relationship $FB(M(P), Pw_i)$ is related to the risks R and the conditions of operation of the cryptographic information security system. To optimize the risks, we will take into account the following factors [14, 20, 21]: quality and reliability, information capacity, and risk factors R associated with the software that provides cryptographic encoding.

For the functioning of the algorithm for assessing the degree of risk of CPS information security, i.e., for the control of underground metal structures, we use the structural elements of the Vaadin, Spring, AnyLogic, and OptQuest frameworks, which together provide a reliable database that ensures effective protection of CPS, as well as the cathodic protection system metal structures (CPSMS) [22-25].

Vaadin offers a server-oriented architecture based on Java Enterprise Edition (JEE) [25]. Using JEE allows you to execute the bulk of the program logic in the context of the server, and also allows you to interact with the user, taking into account the capabilities of desktop applications [25]. To display user interface (UI) elements and interact with the server on the client side, Vaadin uses its own set of web components [25]. An important feature of the Vaadin framework is that the interaction between the server and the browser is fully automated, and this connection allows developers to create web interfaces efficiently and quickly without having to manually code HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) elements [24].

The Vaadin Framework simplifies the process of creating and maintaining high-quality web user interfaces and is structured as a server API, a client API, a set of UI components on both sides, a theme engine for designing the interface, and a data model that allows you to link server components directly to data [25].

The server part of the Vaadin application runs on any servlet or portlet container, accepts HTTP requests from the user and interprets them as events of a specific session [25]. Events associated with client interface components are delivered to those event listeners that appear in the application and the corresponding UI logic is accompanied by changes to the UI components on the server side, this procedure provides the process of obtaining an image with a model response for display in a web browser [25].

The Vaadin Framework corresponds to two models of web application development: for the client (browser) and the server side, respectively, and the client layer of the Vaadin interface includes two types of components: basic UI components and application components [25].

The UI component library of the Vaadin framework contains a set of pre-built, tested, and well-documented user interface (UI) components that can be easily reused in the UI of a product [25]. This ensures that the final product has a user-friendly appearance, promotes efficiency and scalability in its operation, and allows users to perform actions and monitor processes and events [25]. The noted components of the Vaadin interface are quite general and can be introduced into various programs since they do not violate the logic of the computer program and can be successfully used to control R risks.

The purpose of the Spring Boot project is to simplify the creation of Spring-based programs and to create web applications with minimal effort from designers in terms of configuration and code writing [23]. The Spring framework provides comprehensive support for web application infrastructure for developing Java programs [23].

Spring Boot is focused on extending the Spring environment with Spring, Spring Boot, and Spring Cloud components, eliminates some of the template configurations required to set up a Spring application, and provides a fairly fast and efficient system for developing various partial tasks that are useful for cybersecurity [23].

Spring Framework is a universal, widespread application framework that focuses on the corporate development of complex programs in the Java environment and provides a modular approach, as well as allows for the implementation of aspect-oriented programming (AOP) dependencies and rational data access [23].

Aspect-oriented programming (AOP) complements object-oriented programming (OOP) by specifying information about the program structure [23]. The key unit of modularity in OOP is the class, while in AOP the unit of modularity is the aspect, and the set of aspects and classes makes it possible to implement modules in procedures such as transaction management that combine several types and objects [23]. In the literature, problems such as transaction management with AOP are related to the structure of AOP and are often called cross-cutting problems [23].

That is, AOP can be a kind of tool for solving problems that are not directly related to Spring logic but allows you to modify the behaviour of existing Spring code without changing its functionality [23]. Spring has its own AOP framework that complements OOP by introducing a new technique for achieving modularity and greater code cleanliness that is conceptually easy to understand and allows you to effectively solve most practical problems in enterprise Java applications [23].

Since Spring AOP is implemented in the object-oriented Java programming language, there is no need for a special compilation procedure. When using Spring AOP, there is no need to manage the class loader hierarchy, because Spring AOP creates the conditions for use in a Servlet container or application server [23].

Spring AOP primarily supports method composition points and in this case, provides recommendations for method execution on Spring components [23]. Field capture operations (sensitive information) are often not implemented, although support for field capture can be added without breaking the core Spring AOP APIs [23]. If the user needs to advise access to the field and update the connection points, then it is worth considering an additional task in the AspectJ programming language [23].

Spring AOP's approach to AOP differs from most other AOP frameworks [23]. The goal in this context is not to provide the complete AOP implementation (although Spring AOP is quite capable of doing so); rather, it is to ensure tight integration between the AOP implementation and Spring IoC to help provide effective algorithms for solving typical cybersecurity problems in enterprise applications [23].

The functionality of the AOP Spring Framework is usually used in conjunction with the Spring IoC (Inversion of Control) container, and the characteristics of the corresponding implementation of the cybersecurity program are configured using the bean-component definition syntax, and the corresponding approach allows to implement effective "auto proxy" capabilities and this technique provides a significant difference from other AOP implementations [23]. The experience of the conducted research shows that Spring AOP provides a rational and successful solution to most of the tasks in enterprise Java programs that are subject to AOP. However, the Spring IoC

container is independent of AOP, that is, the user does not need to use AOP if he does not want to [23]. AOP does not compete with AspectJ to provide comprehensive problem-solving with AOP, and at the same time complements Spring IoC to provide sufficiently efficient solutions to applied cybersecurity problems with middleware [23].

It is known that proxy-based frameworks such as Spring AOP and full-scale frameworks such as AspectJ are valuable and that they complement each other rather than compete [23]. Spring seamlessly integrates Spring AOP and IoC with AspectJ to provide the full capabilities of AOP in a coherent Spring-based application architecture [23]. This integration does not affect Spring AOP: that is, Spring AOP remains backwards compatible [23].

Spring Boot is built on top of the traditional Spring Framework, which is widely used to develop REST APIs and is characterized by standalone applications with a built-in server, standalone productivity applications, simplified configuration, and selective defaults [23].

The Spring Framework Inversion of Control (IoC) components codify formalized design patterns as first-class objects that users can integrate into their applications, provide a formalized means of combining different components, and build on this basis fully working applications that are ready to use [23]. The Spring Framework always offers choices and the user has the freedom to make an informed decision about which option is best suited for their particular use case or scenario, and modifies formalized design patterns as separate cybersecurity objects that the user can integrate into their applications [23].

AnyLogic integrates the leading OptQuest optimizer and is based on sophisticated analytical algorithms that allow finding optimal parameter values when solving cybersecurity problems using modeling tools and help users make informed decisions [24].

The system for protecting a metal structure element in an aggressive soil environment, taking into account the Vaadin, Spring, AnyLogic, and OptQuest frameworks, will be linked to the $R_S(R)$ resource for the safe operation of a material with a damaged insulating coating in a corrosive environment, and the corresponding ratios will be written in a form similar to that in [25]:

$$R_S(R) = R_w \cdot w(R); \quad R_w(R) = K_w \cdot R_p(R). \quad (3)$$

Here, $R_p(R)$, and $w(R)$ are the design and relative service life of the metal structure material in the air, respectively; $K_w = K_w(N_p, N_s)$ is the coefficient of influence of the aggressive environment on the durability of the metal structure material; N_p, N_s is the durability of the metal structure material in the air and aggressive environment; $R_w = R_w(M(P), Pw_i)$ is the service life of the metal structure material with a damaged insulation coating underground under the condition of contact with the soil. The influence of an aggressive environment on the parameters characterizing the service life of a metal structure is characterized using the methods given in [26, 27]. The effect of an aggressive environment on the service life and protection system of a metal structure is characterized using the methods given in [21, 28].

The effectiveness of UMC protection depends on the intensity of corrosion processes on the surface. Relevant engineering objects are protected by coatings and cathodic protection equipment (CPE). The UMC surface is covered with defects such as cracks and pittings. The corrosion inhibition coefficient and information security risks of the relevant cyber-physical system (CPS) depend on the state of the defects [29, 30].

Let us consider the diagnostic value of information $C_{Di}(k_j)$ for the CPS system, which includes UMC, CPE, and the sensor system [13-16, 29]. The diagnostic value $C_{Di}(k_j)$ for a feature k_j for a state (diagnosis) D_i is the amount of information about incidents and vulnerabilities (including frameworks) (incidents (In) and vulnerabilities (Vn)) contributed by all variants of the corresponding feature implementation in establishing the corresponding state [29]. We propose to write the expression $C_{Di}(k_j)$ for the m -bit feature in the same way as in [29]:

$$C_{Di}(k_j, D_i) = \sum_{s=1}^m P(k_{js} / D_i) \log \frac{P(k_{js} / D_i)}{\sum_{i=1}^n P(D_i) \cdot P(k_{js} / D_i)} = C_{In}(D_i) + C_{Vn}(D_i). \quad (4)$$

Here, in (4), the feature k_j has several values k_{js} for a given object, and these values together form the realization of the feature k_j ; $P(k_{js}/D_i)$ is the probability of the state (diagnosis) D_i , provided that the feature k_j has received the value k_{js} ; $P(D_i)$ is the a priori probability of the diagnostic state.

Let us write down the ratio of the integral value of information $C_D(D, CPS)$ for the procedure of integral diagnosis $D=D(R)$ of the CPS system, taking into account the division into two components $C_{Inz}(D)$, $C_{Vnz}(D)$, where the $C_{Inz}(D)$ component corresponds to incidents and $C_{Vnz}(D)$ to vulnerabilities:

$$C_D(D, CPS) = C_{Inz}(D, R) + C_{Vnz}(D, R). \quad (5)$$

To ensure a high level of security of a cyber-physical system (CPS), we use a cryptography algorithm (ACR), which is characterized by quality, stability, and reliability. It is formed using elements of the AES and RSA algorithms, as well as a system for testing, encrypting, and decrypting information using software that operates with the NetBeans program [31]. The proposed approach (5) takes into account the interdependencies of vulnerabilities and uses elements of graph theory. The method of randomization and encryption based on the method of matrix operations (MORE) was used to train neural networks using encrypted data [32]. It was found that the use of a neural network approach, as well as training a neural network using MORE, improves the accuracy, running time, and performance of CPS, as well as the appropriate level of privacy compared to other state-of-the-art methods [32, 33]. The quality of a cryptography algorithm $Q_{Cz}(R_A, T_C, S_C, A_R)$ is characterized by four parameters [31, 32]: reliability R_A , time complexity T_C , spatial complexity S_C , accuracy A_R , i.e., the level of ensuring the correct result.

Let's write down the criterion relations for the functional dependencies $C_{Inz}(D)$, $C_{Vnz}(D)$, $Q_{Cz}(R_A, T_C, S_C, A_R)$, as well as for the structural strength S_S :

$$\begin{aligned} C_{Inz}(D, R) &\Rightarrow \min; \quad C_{Vnz}(D, R) \Rightarrow \min; \quad Q_{Cz}(R_A, T_C, S_C, A_R, R) \Rightarrow \max; \\ S_S(*) &= S_S(W_S, \sigma_S, K_{CV}, W_{PL}, K_{1C}, R) \Rightarrow \max. \end{aligned} \quad (6)$$

Here, W_S , σ_S are the surface energy and tension of the structural material, respectively; K_{CV} is the impact toughness of the material; W_{PL} is the surface plastic strain energy of the material; K_{1C} is the fracture toughness of the material [33].

Let us convert the functional dependencies $C_{Inz}(D)$, $C_{Vnz}(D)$, $Q_{Cz}(R_A, T_C, S_C, A_R)$, $S_S(*)$ to a dimensionless form, in particular:

$$C_{In}(D) = C_{Inz}(D) / (C_{Inz}(D))_{CP}; \quad C_{Vn}(D) = C_{Vnz}(D) / (C_{Vnz}(D))_{CP}; \quad (7)$$

$$Q_C(*, D) = Q_C(R_A, T_C, S_C, A_R) = Q_{Cz}(R_A, T_C, S_C, A_R) / (Q_{Cz}(R_A, T_C, S_C, A_R))_{CP}; \quad (8)$$

$$S_S(*, D) = S_S(W_S, \sigma_S, K_{CV}, W_{PL}, K_{1C}),$$

$$S_S(*, D) = S_S(W_S, \sigma_S, K_{CV}, W_{PL}, K_{1C}) / (S_S(W_S, \sigma_S, K_{CV}, W_{PL}, K_{1C}))_{CP}. \quad (9)$$

Here $(C_{Inz}(D))_{CP}$, $(C_{Vnz}(D))_{CP}$, $(Q_{Cz}(R_A, T_C, S_C, A_R))_{CP}$, $(S_S(W_S, \sigma_S, K_{CV}, W_{PL}, K_{1C}))_{CP}$ – values of the relevant functions at critical (special) points. These can be, for example, the maximum possible values of the relevant factors.

Since the factors (functions) $C_{In}(D)$, $C_{Vn}(D)$, $Q_C(*, D)$, $S_S(*, D)$ in relations (7)-(9) are dimensionless, then they can be used to formulate a trade-off function $F_Q(D, R)$ (Trade-off function) and, with its help, the optimization criterion, since the first two of them should tend to decrease in terms of safety, and the third and fourth in absolute value should increase:

$$F_Q(D, R) = \alpha \cdot C_{In}(D) + \beta \cdot C_{Vn}(D) + \gamma \cdot Q_C(*, D) + \delta \cdot S_S(*, D) \Rightarrow opt, \quad (10)$$

where $C_{In}(D)$, $C_{Vn}(D)$, $Q_C(*, D)$, $S_S(*, D)$ – functional dependencies that we establish based on the experiment; α , β , γ , δ – weighting coefficients, which are determined on the basis of a computational experiment, taking into account information of the type presented in Fig. 4, as well as criteria (1), (2), (5). The result obtained using (10) makes it possible to: propose a new

methodology for assessing the degree of risk of CFS information security, taking into account the index of the probability of a successful attack on the system and the index of adjustment, which provides feedback.

As a result, taking into account the quality criteria (2), (5), parameters characterizing the resource (3), and the corresponding optimization algorithm based on the trade-off function (10), it is possible to assess the risk degree R of the metal structure and increase the efficiency of the CPS protection system against unauthorized influences and threats.

Example. The results of testing the leak detection system (LDS) on the linear part of the main oil pipeline "section 43" station "5C" - station "1K" from 0 km to 231 km were used [1]. The tests were carried out with full verification of the system operation algorithms, by the design decisions, by organising oil product discharges from the pipeline according to the approved programme and testing methodology of the IHS [1]. A partial result of a series of tests conducted using pressure sensors according to the methodology [1] is shown in Table 2. To check the performance of the IED, oil product discharges were organised at 2-3 points of the protected section of the oil pipeline "section 43", 0-231 km, with different intensity, leakage development time and for different modes of operation of the pipeline [1].

In the course of the work, the following were performed [1]: verification of functional requirements for the IED; verification of the IED characteristics; checking the system for stability; and checking the accuracy and reliability of leakage detection within the protected area.

At the time of testing, the complete set of software and hardware was provided, as well as the completeness of permits and technical documentation following norms and standards for all supplied equipment [1]. Before the start of the tests, the readiness of the material, technical and metrological support facilities was checked to ensure that the conditions and test modes were created under the programme [1]. The tests were carried out to determine the appropriate size of the minimum recorded leakage for a given oil pipeline in the following modes of operation [1]: maximum, minimum, and one of the intermediate modes.

The tests were carried out at each of the modes of operation of the oil pipeline with the following leakage development time [1]: up to 1 s, - from 1 to 10 s, more than 10 s.

The tests were carried out in series of 2-3 showers for each leak, taking into account the following conditions [1]: 4 series of experiments were carried out; the first - three experiments, the second - three, the third - two, the fourth - three experiments. The mean values and standard deviations for each series of experiments were calculated.

Table 1
Abbreviated table of system test results [1]

Tests Results	Date and Time	The duration of the leak ΔT , s	The size of the leak Q , l	The magnitude of the leak ΔP , mPa (out/in)	The fact of detecting a leak	Actual coordinate of selection, km	System leakage coordinate, km	Error _k , m
	1	2	3	4	5	6	7	8
1	12.02 12:00	1	1	0.02/*	Yes	23.4	23.55	150
2	12.02 12:10	1	2	0.03/*	Yes	23.4	23.15	250
3	12.02 12:20	1	2	0.03/*	Yes	23.4	23.2	200
4	12.02 13:30	10	15	0.07/0.024	Yes	56.1	56.4	300
5	12.02 13:40	10	15	0.07/0.024	Yes	56.1	56.35	250

6	12. 02 10 13:50	15	0.07/0.024	Yes	56.1	56.4	300
7	12. 02 60 14:00	80	0.12/0.04	Yes	120.7	120.3	400
8	12. 02 60 14:10	90	0.12/0.04	Yes	120.7	120.2	500
9	15. 02 1 12:00	1	0.02/*	Yes	140.4	140.1	300
10	15.02 1 12:10	2	0.03/*	Yes	140.4	140	400
11	15. 02 1 12:20	1.5	0.03/*	Yes	140.4	140.2	200

Average (mean) value of Error= 295,45 m.

So, if we sum all errors in determining the leak coordinate and divide them by the total number of tests, we should get an average (mean) value, which should be less than $\pm 300\text{m}$ ($k = 1, 2, \dots, 11$).

A refined leakage test was carried out using the BVS-K non-contact current meter and the VPP-M polarisation potential meter for non-destructive testing of metal surfaces according to the methodology presented in [13-16,34]. As a result, the average error value $\text{Errors}_2 = 2\text{ m}$ was obtained. The relative value of $\text{Errors}_3 = 2/300 = 0.007$. It is irrational to control leaks only with the help of BVS-K and VPP-M devices. It is more rational to conduct leakage control based on pressure sensors at the first stage and obtain results similar to those in Table 1, and at the second stage, it is possible to clarify the locations of leaks (depressurisation) using remote monitoring devices BVS-K and VPP-M.

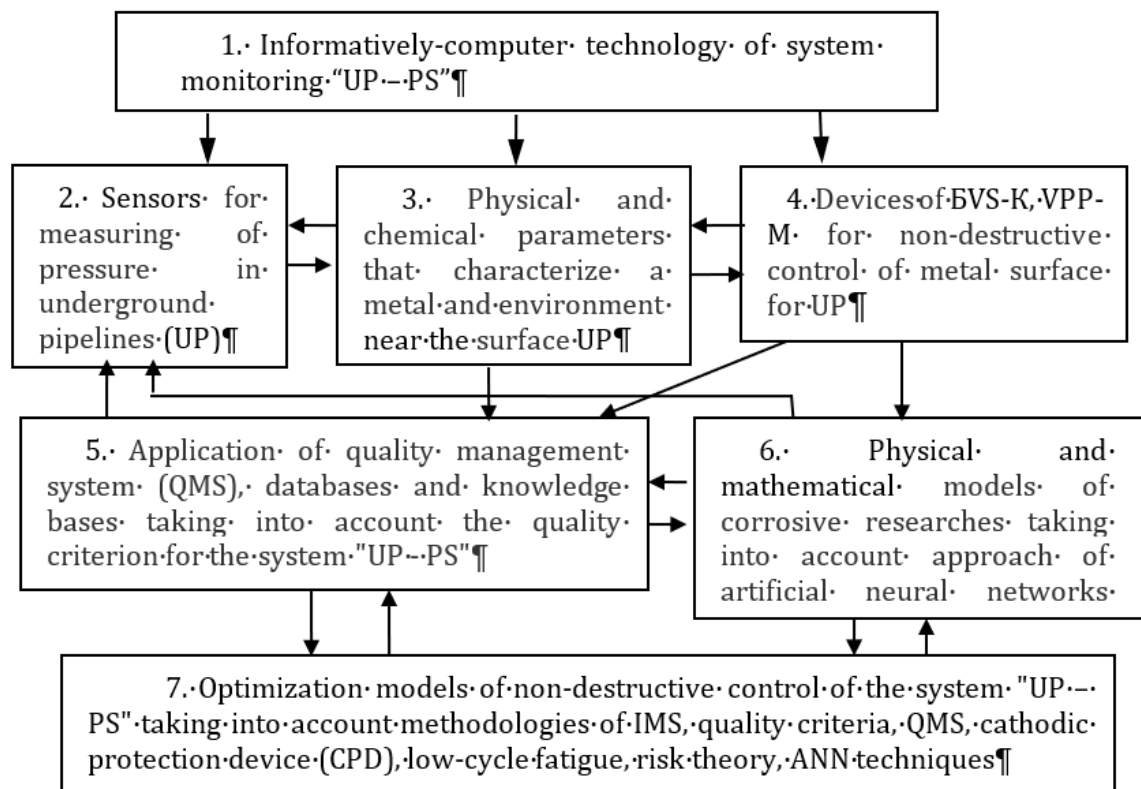


Figure 5: Block diagram of informatively-computer technology for underground pipelines (UP) taking into account the informatively-measuring system (IMS) (2, 3, 4) for control of corrosive processes [34]

In actual pipeline operation conditions, it is necessary to take into account the possibility of corrosion defects such as pitting, cavities, and cracks, which change their size over time and eventually become fracture sites. We describe their evolution using relations (1)-(10), taking into

account the monitoring system, which is based on a cyber-physical system (CPS) for modelling the electrophysical parameters of the system "underground metal structure - external aggressive environment", taking into account the risk assessment unit and the cryptographic information security system (Fig. 4).

To diagnose defects on the surface of the pipe and places of depressurisation (leakage) of the pipeline using: pressure sensors and non-destructive testing devices, which measure potentials and corrosion currents, an information and computer technology has been developed, the structural diagram of which is presented in [34] (Fig. 5) for the underground pipeline (UP) - pumping station (PS) system.

The scheme in Fig. 5 contains 7 blocks, of which blocks 2, 3, 4 correspond to the information and measurement system (IMS) for measurements. Blocks 5, 6, 7 are the basis of the system for processing information received by sensors and non-destructive testing devices (IMS). The methods of operation of units 5, 6, 7 are partially described in articles [13-16, 34].

The scheme in Fig. 5 contains 7 blocks from which blocks 2, 3, 4 answer the informative-measuring system (IMS) for realisation of measurement. Blocks 5, 6, 7 are the basis of the information processing system received by sensors and non-destructive control devices (IMS). Methodologies of functioning of blocks 5, 6, 7 are partially described in articles [13-16, 34].

For the cyber-physical system (CPS) (Fig. 4) and information and computer technology (Fig. 5), we use the Vaadin, Spring, AnyLogic, OptQuest frameworks, as well as the principles of cryptographic encoding and decoding of information

Conclusions

The basics of the method and a variant of the methodology for improving the efficiency of the CPS protection system of an underground metal structure against unauthorized influences and threats, taking into account the corrosive effect on the life and strength of the material, quality criteria (2, 5) and the corresponding optimization algorithm based on the trade-off function and the Vaadin, Spring, AnyLogic, OptQuest frameworks, are presented.

1) An approach to monitoring the system "underground metal structure (UMC) - aggressive environment (AE)" was developed, taking into account information monitoring methods, methods for assessing the life of the corresponding cyber-physical system, as well as optimization criteria related to the life of metal structural elements, risk assessment units, and cryptographic information security system.

2) A system for assessing the value of information for the procedure of integral diagnosis of a cyber-physical system CPS was developed, taking into account the division of the components of the value of the diagnosis D into two parts, where one part $C_{inz}(D)$ corresponds to incidents, and the second $C_{vnz}(D)$ - to vulnerabilities, taking into account the Vaadin, Spring, AnyLogic, OptQuest frameworks.

3) A compromise function has been developed that can be used to ensure the functioning of a cyber-physical system with given values of risk, the value of information about incidents, the value of information related to CPS vulnerabilities, as well as the strength of structural elements, and parameters that characterize the quality of the cryptography algorithm.

Using the trade-off function, options for methods of improving the efficiency of the CPS protection system, and the initial values of the parameters $M(P)$, specific partial tasks can be formulated and the corresponding risk values can be determined.

For the functioning of the optimization algorithm for assessing the degree of risk of information security of a cyber-physical system and, in this context, to ensure the conditions for controlling the protection system of underground metal structures, we use the structural elements of the Vaadin, Spring, AnyLogic, OptQuest frameworks, as well as the principles of cryptographic encoding and decoding of information.

References

- [1] Anatoliy Obshta, Yurii Biliak, Vladyslav Shugai. Cyber-Physical System for Diagnostic Along the Controlled Section of the Oil Pipeline // Advances in Cyber-Physical Systems Vol. 8, Num. 1, 2023 pp. 66-73 DOI: 10.23939/acps2023.01.066; Language: EN.

- [2] Ngonadi I. V., Ajiroghene S. Remote Pipeline Monitoring Security System // International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 2021. Vol. 7, No. 6. P. 135-145. <https://doi.org/10.32628/CSEIT217631>.
- [3] Pipeline Security and Incident Recovery Protocol Plan / Transportation Security Administration. U. S. Department of Homeland Security. 2010. 58 p.
- [4] Pipeline Infrastructure Vandalism Monitoring using Wireless Sensor Networks Technique / Y. Kefas, A. O. Sunday, A. M. Bashir and A. A. Abraham // International Journal of Computer Science and Telecommunications. Vol. 8, No. 2, 2017. P. 39-42.
- [5] Lerner, L. W. Farag, M. M., Patterson, C. D.: Run-time prediction and preemption of configuration attacks on embedded process controllers. In: SecurIT '12: Proceedings of the First International Conference on Security of Internet of Things, August 2012, P. 135–144. (2012). <https://doi.org/10.1145/2490428.2490447>.
- [6] Lee, M.-C.: Information Security Risk Analysis Methods and Research Trends: AHP and Fuzzy Comprehensive Method. International Journal of Computer Science & Information Technology (IJCSIT). 2014. 6(1), P. 29-45. DOI:10.5121/ijcsit.2014.6103.
- [7] Biro M., Mashkoo A., Sametinger J., Seker R.: Software Safety and Security Risk Mitigation in Cyber-physical Systems. IEEE Software. 2018. 35(1). P. 24–29.
- [8] Hu F.: Cyber-Physical Systems: Integrated Computing and Engineering Design. New York: CRC Press. 2018. 398 p.
- [9] Design and validation of a C++ code generator from abstract state machines specifications / S Bonfanti, A Gargantini, A Mashkoo // Journal of Software: Evolution and Process. 2020. 32 (2), e2205.
- [10] Bendoly E., Rosenzweig E. D., Stratman J. K. The efficient use of enterprise information for strategic advantage: A data envelopment analysis // Journal of Operations Management, 2009. No. 27. P. 310-323.
- [11] Construction Site Monitoring Data Processing Based on Detecting Anomalies and Improved Variational Mode Decomposition / Y. Shaoa, T. Anb, Y. Qic, W. Liu // Proceedings of the 2023 5th International Conference on Structural Seismic and Civil Engineering Research (ICSSER 2023), Atlantic Highlights in Engineering 24. 2023. P. 258-269. https://doi.org/10.2991/978-94-6463-312-2_27.
- [12] Ahmed, M., Mahmood, A. N., and Hu, J. A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 2016. 60: P. 19-31. DOI: 10.1016/j.jnca.2015.11.016.
- [13] Yuzevych, L., Skrynkovskyy, R., Yuzevych, V., Lozovan, V., Pawlowski, G., Yasinskyi, M., Ogirko, I.: Improving the diagnostics of underground pipelines at oil-and-gas enterprises based on determining hydrogen exponent (ph) of the soil media applying neural networks. Eastern-European Journal of Enterprise Technologies. 2019. 4(5 (100)), P. 56–64. doi: <https://doi.org/10.15587/1729-4061.2019.174488>.
- [14] Lozovan, V., Skrynkovskyy, R., Yuzevych, V., Yasinskyi, M., Pawlowski, G.: Forming the toolset for development of a system to control quality of operation of underground pipelines by oil and gas enterprises with the use of neural networks. Eastern-European Journal of Enterprise Technologies. 2019. 2(5 (98)), P. 41–48. doi: <http://dx.doi.org/10.15587/1729-4061.2019.161484>.
- [15] Yuzevych, L., Yankovska, L., Sopilnyk, L., Yuzevych, V., Skrynkovskyy, R., Koman, B., Yasinska-Damri, L., Heorhiadi, N., Dzhalala, R., Yasinskyi, M.: Improvement of the toolset for diagnosing underground pipelines of oil and gas enterprises considering changes in internal working pressure. Eastern-European Journal of Enterprise Technologies. 2019. 6(5 (102)), P. 23–29. DOI: 10.15587/1729-4061.2019.184247.
- [16] Lozovan V., Dzhalala R., Skrynkovskyy R., Yuzevych V.. Detection of specific features in the functioning of a system for the anti-corrosion protection of underground pipelines at oil and gas enterprises using neural networks // Eastern-European Journal of Enterprise Technologies. 2019. Vol. 1, No. 5 (97). P. 20–27. doi: <https://doi.org/10.15587/1729-4061.2019.154999>.
- [17]. Chen Z., Pouliot J., Hubert F. Task decomposition and level of complexity to select the content of underground utility network model // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-4/W4-2022 17th 3D GeoInfo Conference, 19–21 October 2022, Sydney, Australia. P. 21-27. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-21-2022>, 2022.

- [17] Agmon, N.; Kordova, S.; Shoval, S. Global Quality Management System (G-QMS) in Systems of Systems (SoS)—Aspects of Definition, Structure and Model. *Systems* 2022, 10, 99. <https://doi.org/10.3390/systems10040099>.
- [18] International Organization of Standardization. ISO 9004:2018 Quality Management—Quality of an Organization—Guidance to Achieve Sustained Success; ISO: Geneva, Switzerland, 2018.
- [19] Wicik R., Borowski M. Cryptographic protection of classified information in military radio communication faced with threats from quantum computers // *Proc. SPIE 11442, Radioelectronic Systems Conference 2019, 114420Q* (11 February 2020); 8 p. doi: 10.1117/12.2565467.
- [20] Encryption Technology in Information System Security / S. Suo, W. Xi, T. Cai, G. Jian, H. Yao, J. Li // 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019). *Advances in Computer Science Research*, 2019. Vol. 87. P. 495-499.
- [21] Mattila T. Building a complete fullstack software development environment. Turku University of Applied Sciences. 2018. 112 p.
- [22] Spring Framework Reference Documentation. <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/index.html>.
- [23] The official site of the AnyLogic company. Access mode: <https://www.anylogic.com/>.
- [24] vaadin/docs: Official documentation for Vaadin and Hilla. <https://github.com/vaadin/docs>.
- [25] Wilds N. Corrosion under insulation. In: *Energy Trends in Oil and Gas Corrosion Research and Technologies*. Duxford, UK: Woodhead Publishing, 2017: P. 409–429.
- [26] Eltai E. O., Musharavati F., Mahdi E. Severity of corrosion under insulation (CUI) to structures and strategies to detect it // *Corros Rev* 2019; 37(6): P. 553–564. <https://doi.org/10.1515/corrrev-2018-0102>.
- [27] Furrer F. J. Safety and Security of Cyber-Physical Systems. Engineering dependable Software using Principle-based Development. Springer Vieweg Wiesbaden. 2022. 537 p. <https://doi.org/10.1007/978-3-658-37182-1>.
- [28] Yuzevych V., Skrynkovskyy R., Koman B. Intelligent analysis of data systems for defects in underground gas pipeline // *Proceedings of the 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. Lviv, Ukraine. August 21-25, 2018. P. 134-138. DOI: 10.1109/dsmp.2018.8478560
- [29] X. Lyu, Y. Ding, S.-H. Yang, Safety and security risk assessment in cyber-physical systems // *IET Cyber-Physical Systems: Theory & Applications*. 2019, Vol. 4, No. 3. P. 221-232.
- [31]. Pattanavichai S. Program for Simulation and Testing of Apply Cryptography of Advance Encryption Standard (AES) Algorithm with Rivest-Shamir-Adleman (RSA) Algorithm for Good Performance // *International Journal of Electronics and Telecommunications*, 2022, Vol. 68, No. 3, P. 475-481. Doi: 10.24425/ijet.2022.141263.
- [32]. Enhanced Security in Cloud Computing Using Neural Network and Encryption / M. U. Sana, Z. Li, F. Javaid, H. B. Liaqat, and M. U. Ali // *IEEE Acces*, 2021, Vol. 9, October. P. 145785-145799.
- [33]. Yuzevych, V. M., Lozovan, V. P. Influence of Mechanical Stresses on the Propagation of Corrosion Cracks in Pipeline Walls // *Materials Science*, 2022. Vol. 57, No. 4. P. 539-548; <https://doi.org/10.1007/s11003-022-00576-z>. DOI 10.1007/s11003-022-00576-z.
- [34]. Yuzevych, V., Horbonos, F., Rogalskyi, R., Yemchenko, I., & Yasinskyi, M. (2020). Determination of the Place Depressurization of Underground Pipelines in the Monitoring of Oil and Gas Enterprises. *International Journal of Recent Technology and Engineering (IJRTE)*, 2020. Vol. 9, No. 1. P. 2274–2281. <http://doi.org/10.5281/zenodo.3841287>.

Electricity Forecasting Software for Microgrid Energy Management System

Yuliia Parfenenko ¹, Vira Shendryk ¹, Yevhen Kholiavka ¹, Oleh Miroshnichenko ¹

¹Sumy State University, 116 Kharkivska Street, Sumy, 40007, Ukraine

Abstract

This article outlines the development of a forecasting software tailored for managing energy in microgrids. It discusses the creation of forecasting models using linear regression and random forest techniques. The software's purpose is to employ these models, handle input data, and store forecast results in a structured manner within a database. This meticulous organization enables efficient processing and timely decision-making in microgrid energy management.

The study primarily focuses on building a power demand forecasting subsystem utilizing random forest and linear regression models. It covers various stages, including project planning and execution, with special emphasis on designing a robust database structure for effective data management. Additionally, the implementation involves an activation module crucial for analyzing and forecasting power grid parameters, enhancing the forecasting process's efficiency.

Through rigorous analysis, the Random Forest Model emerges as the primary forecasting tool. The trained forecasting models are stored in S3 cloud storage. Following model selection, the Forecasting Software's architecture were designed, incorporating essential components for seamless microgrid energy management system operation.

The developed electricity forecasting software is used for activating pre-trained models, processing input data, and logging forecast results into the database. It ensures smooth functioning and facilitates the integration of forecasted data into decision-making processes within power grid management systems.

Keywords

Forecasting, electricity, random forest, linear regression, machine learning, energy management

1. Introduction

Hybrid energy networks are becoming a key element of the transition to a more sustainable and efficient energy future [1]. Hybrid systems combine various energy sources, such as renewable (solar, wind, hydropower, etc.) and traditional energy sources, forming complex networks that meet the needs of consumers. However, solving the problems associated with managing such complex systems requires advanced methods and tools, particularly software modules for decision support.

The management of the hybrid energy network is faced with a number of complex tasks, including forecasting production [2] and energy consumption, optimization of system operation modes, and ensuring the stability of the energy balance. One of the key aspects of management is the ability to accurately forecast changes in energy production and consumption based on available data.

The relevance of developing a software module to support decision-making in managing a hybrid power grid is obvious due to the increasing complexity and volume of processed data. Using forecasting models and analytical tools [3] can greatly facilitate the process of making optimal decisions, increasing energy systems' efficiency and stability.

One of the key aspects of managing such a system is accurate forecasting of energy consumption based on available data. In the context of this study, the software will be responsible

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ yuliya_p@cs.sumdu.edu.ua (Yu. Parfenenko); v.shendryk@cs.sumdu.edu.ua (V. Shendryk); yevhen.kholyavka@itp.sumdu.edu.ua (Ye. Kholiavka);

ORCID 0000-0003-4377-5132 (Yu. Parfenenko); 0000-0001-8325-3115 (V. Shendryk); 0000-0002-9841-7909 (Ye. Kholiavka); 0009-0005-6761-6269 (O. Miroshnichenko)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

for launching a system for forecasting electricity consumption volumes, which is a critical element for effective hybrid power grid management.

Forecasting models for decision support systems (DSS) should cover various time intervals, as decisions may be made both operationally and during the planning process. This is important considering the need for forecasts over short, medium, and long time periods. One of the pertinent challenges is the storage and activation of necessary models within specified time frames to obtain relevant forecasts timely.

This study aims to develop software for electricity demand forecasting, which can be used as a subsystem in a hybrid energy grid management decision support system.

Tasks include analyzing management issues in hybrid energy, selecting and developing appropriate forecasting models, integrating them into the software module, creating the module's architecture, and testing its effectiveness in practice. This approach will contribute to improving the quality of management in hybrid energy and fostering the development of more resilient and cost-effective energy systems.

This study introduces a novel approach by developing specialized software for forecasting electricity demand within hybrid energy grids. By addressing the complexities of managing such systems through advanced forecasting models and decision support tools, it aims to enhance operational efficiency and system stability. The focus on covering various time intervals in forecasting and integrating these models into a cohesive software module represents a pioneering effort towards optimizing hybrid energy management.

2. Literature review

Before delving into assessing the effectiveness of forecasting models for electricity consumption and developing an activation module, it is essential to evaluate various models applicable to the forecasting process thoroughly. Article [2] delves into selecting an optimal model for forecasting hourly electricity consumption, emphasizing a significant dataset. Various models, including autoregressive and machine learning models, underwent examination. The study focused on comparing model performance, particularly emphasizing autoregressive models and deep learning models like LSTM.

Ultimately, the results indicate that the LSTM model displayed superior forecast performance for large datasets compared to the autoregressive SARIMAX model. Thus, it underscored the effectiveness of using deep learning models for forecasting electricity consumption when dealing with substantial data volumes.

Article [11] underscores the importance of selecting the appropriate method for forecasting microgrid load. It suggests that using random forest (RFR) with feature selection enhances forecast accuracy, considering crucial characteristics affecting electricity consumption and providing reliable load forecasts.

The development of accurate forecasting models necessitates carefully considering different methods, particularly those accounting for time dependence and data seasonality, such as random forest and linear regression. However, the suitability of each algorithm for specific time series forecasting tasks must be meticulously assessed.

The forecasting software development is an active area of research. The studies presented in this review demonstrate significant progress in this field. The appropriate architecture, framework, and deployment tool selection depend on your specific requirements and resources.

Article [5] presents an overview of the MLflow open platform aimed at managing the life cycle of machine learning. It provides tools and capabilities for tracking experiments and managing and deploying models.

The authors systematically analyze various aspects related to MLflow and consider their impact on machine learning processes. Special attention is paid to the possibilities of registration and tracking of parameters and results of experiments, which allows proper analysis and comparison of different approaches and models. In addition, model versioning tools are considered, which is important to ensure stability and reliability in development and deployment.

The paper also discusses aspects of model deployment, which is a key stage in the machine learning lifecycle. MLflow provides a user-friendly interface for deploying and managing models

in a production environment, facilitating rapid implementation and use of trained models in practical applications.

Kubeflow is an open-source platform designed to create and deploy machine learning pipelines on Kubernetes. In this article, the authors thoroughly describe Kubeflow's architecture and illustrate how the platform enables developers to build and scale machine learning pipelines using the various tools available in its ecosystem.

Kubeflow architecture includes a number of components that are designed for different aspects of machine learning, such as data preparation, model training, validation of results, and deployment of models to a production environment. The article's authors examine these components and their interaction, explaining how they can be used to create complete machine-learning pipelines.

One of the main advantages of Kubeflow is its integration with Kubernetes, which allows you to automate the processes of deployment and management of computing resources. This makes Kubeflow a powerful tool for developing and implementing complex and large-scale machine-learning pipelines.

The article [7] presents TorchServe as a framework for deploying PyTorch models. The authors describe in detail the capabilities of this framework, including the scaling, monitoring, and model management features that make it an attractive solution for deploying models in a production environment.

TorchServe provides developers and researchers with a convenient tool to deploy and manage their PyTorch models in a production environment. The framework automates the deployment process, enabling efficient scaling, and provides real-time model monitoring and management.

One of the key strengths of TorchServe is its flexibility and ability to integrate with other tools and environments, making it a versatile solution for deploying models in various scenarios and requirements.

While TorchServe offers advantages, potential drawbacks include configuration complexity, limited model compatibility, resource costs, security concerns, and the need for ongoing support and updates.

Overall, the paper provides a detailed overview of the functionality and benefits of TorchServe and demonstrates its potential importance in the field of deploying machine learning models in production using the PyTorch framework.

Article [8] examines Amazon SageMaker Serverless Endpoint as a Serverless solution for deploying machine learning models. The authors describe in detail the advantages of using serverless approaches in the context of deploying models and provide practical examples of using Amazon SageMaker Serverless Endpoint.

The article examines the key functionalities of Amazon SageMaker Serverless Endpoint, including its ability to automatically scale as needed, monitor and manage models. The authors highlight the importance of these capabilities in ensuring efficient and reliable deployment of machine learning models in a real production environment.

In addition, the article provides practical examples of using Amazon SageMaker Serverless Endpoint, which helps readers better understand how this solution can be applied in specific machine-learning scenarios. This allows readers to evaluate the potential benefit and suitability of this solution for their own projects and tasks.

Despite its advantages, some drawbacks of Amazon SageMaker Serverless Endpoint may include limited configurability, AWS service costs, and restricted flexibility for certain complex model configurations.

The article [9] discusses TensorFlow Serving as a framework for deploying TensorFlow models to a production environment. The authors describe in detail the functionality of TensorFlow Serving, focusing on its ability to scale, monitor, and manage models.

The article provides a detailed overview of TensorFlow capabilities Serving in the context of deploying models in a production environment. Special attention is paid to scaling tools that allow easy handling of large workloads and high availability of models. In addition, the authors consider monitoring tools that help track the effectiveness and reliability of models in a production environment.

Additionally, the article also discusses the model management process, including versioning and updating. This is important to ensure stability and reliability in a production environment and to implement the latest model versions.

TensorFlow's functionality Serving, making it a valuable resource for those seeking to understand and implement efficient deployment models in a production environment.

In the context of deploying forecasting models, model activation plays a key role in ensuring their effectiveness and use in a production environment. Software activation modules ensure the fast and reliable launch of forecasting models, allowing them to be used in real-time to make important management decisions.

The work of forecasting software usually includes several main stages aimed at effectively using models for forecasting. The initial stage is the management of the activation process, where a request for a forecasting model is generated based on the user's needs. This request initiates a request to activate the model, which is then activated for forecasting.

Next, the forecasting model acquisition module executes a request to obtain the required model using the S3 API or other appropriate mechanisms [8]. After receiving the model, it is transferred to the activation module, where the activation process itself takes place, which includes internal procedures for modeling and calculating forecasts.

After activating the model, the received forecasting results are stored in the database. This step involves saving the forecast data in the appropriate database tables for later use. In addition, the recording results in the database can be subjected to additional analytical operations or used for reporting and analysis [9].

The absence of explicit drawbacks mentioned in the provided text makes it difficult to pinpoint specific limitations of TensorFlow Serving or the described forecasting model deployment process. However, potential challenges could include issues related to scalability, complexity of model management, and integration with other systems. Additionally, ensuring the reliability and accuracy of forecasting models in real-time production environments may pose technical difficulties. It's imperative to address these challenges to ensure the successful deployment and operation of the forecasting models, particularly in dynamic and high-demand scenarios. This may involve optimizing TensorFlow Serving for scalability, streamlining model management processes, enhancing integration capabilities, and implementing robust monitoring and validation mechanisms to maintain model accuracy and reliability in real-time settings.

Based on the analysis of literature sources and consideration of possible applications and frameworks for deploying forecasting models, it is noted that various tools can be utilized to address the direct task of electricity consumption forecasting. However, due to the necessity of further integrating forecasting results into a decision support system (DSS), it is expedient to develop specialized software rather than using what already exists. This will provide the integration of forecasting results into the decision support system and ensure high operational efficiency in addressing complex tasks related to energy resource management.

3. Forecasting Methodology

To collect and process data for electricity consumption forecasting, a technique to obtain a dataset from a private enterprise was used. The specified dataset contained indicators of changes in electricity consumption by the enterprise, recorded with a frequency of every hour.

The data collection method included obtaining information from the enterprise, which included the use of contractual agreements and the exchange of data between the parties. As an entity, the enterprise ensured the provision of data, including time indicators of electricity consumption.

After receiving the data, they were processed. This process involved identifying and correcting possible anomalies or errors in the data, as well as transforming and aggregating the data into a format suitable for further analysis and modeling. An analysis of the absence of missing values and their processing in case of detection was also carried out.

Subsequently, the data were subjected to a process of exploratory analysis to identify patterns, trends, seasonal variations, and other characteristics (Figure 1) that may affect electricity consumption. This analysis made it possible to better understand the internal relationship of the data and contributed to the selection of the most suitable forecasting models.

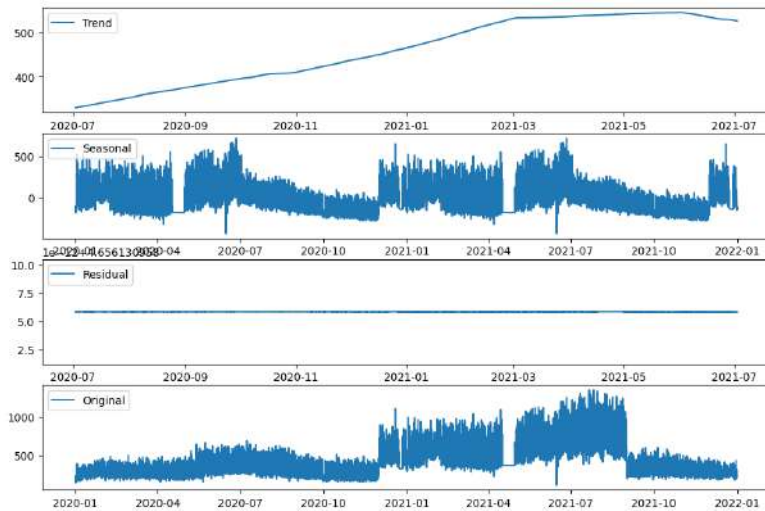


Figure 1: Time series decomposition

Observing the growing trend, seasonality, and character of the residual, which is similar to a straight line, will significantly contribute to understanding the data dynamics. The described data structure can occur in the context of various domains, including economic, technical, or social domains, where certain patterns and trends are present.

An increasing trend indicates a systematic increase in the values in the time series during the observation period. This may indicate economic development, increased demand, or other factors leading to gradual growth.

Seasonality reveals regular cyclical fluctuations in the data that repeat with the same or nearly the same amplitude and shape. However, the straight line-like nature of the residual may indicate a high level of randomness or uncertainty in the data that cannot be explained by trend or seasonality.

This scenario of data dynamics can be interpreted as the discovery of certain regularities in the development of the phenomenon [11], however, with significant uncertainty or randomness of influence in the specified time series. Such an analysis is important for understanding and forecasting the further development of data dynamics in the relevant context.

In the course of the study, a comparative analysis of the effectiveness of two different machine learning methods – linear regression and the random forest method for forecasting electricity consumption was carried out. The data used covered the period from November 2020 to May 2021 and contained information on various parameters such as year, month, week, day of the week, day, hour, and day length, as well as electricity consumption.

First, the data were prepared by dividing it into training and test sets. The training data were used to train the models, while the test data remained to evaluate their performance.

During the research, electricity demand was forecasted for different periods of time, which is determined by the "forecast" parameter. In this case, such forecast periods as 6, 12, 24, 72, and 168 hours were used.

Each of these periods has its own importance and application in the real world. For example, short-term forecasts, such as a 6-hour or 12-hour forecast, can be useful for real-time optimization of power generation or for managing household energy consumption. Long-term forecasts, such as a 24-hour, 72-hour, or 168-hour forecast, can be useful for planning and managing energy resources, such as scheduling scheduled repairs or optimizing the energy use of large industrial enterprises.

Based on the literature analysis, two main models were selected for regression analysis: linear regression and the random forest method. These models were chosen due to their widespread use, flexibility, and effectiveness in handling various data types.

Random forest is a powerful machine learning algorithm that uses an ensemble of decision trees to solve classification and regression problems [12]. One of its main advantages is the ability to automatically model complex nonlinear dependencies and consider the importance of features.

The formula for forecasting a time series using a random forest can be expressed as follows [12]:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^N f_i(x), \quad (1)$$

where: $\hat{y}$ – the forecasting value, N – the number of trees of the random forest, $f_i(x)$ – the output of the model for the i th tree.

Linear regression is a classic analysis method that uses a linear function to forecast response values based on input features [13]. It is a simple and interpretable model that is often used for econometric and forecasting problems.

The formula for linear regression in the form of a time series forecast can be presented as [13]:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n, \quad (2)$$

where: $\hat{y}$ – forecasting value, $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ – regression coefficients, $x_1, x_2, \dots, x_n$ – features of the model.

First, using the random forest method, 100 trees with a random state of 42 were generated, and then the model was trained on the training data. After that, the forecasts that were obtained were compared with the actual values of the test data. The same process was carried out for the linear regression model.

After obtaining the forecasts, both models were evaluated using various quality metrics such as mean squared error (MSE), mean absolute error (MAE), and others. These metrics made it possible to make an objective analysis of the effectiveness of each model in forecasting electricity consumption.

In measuring the effectiveness of forecasting models, such as Linear Regression and Random Forest, various metrics are used that determine the degree of their accuracy and adaptability to data [14]. The main metrics include root mean square error (Mean Squared Error, MSE), coefficient of determination (Coefficient of Determination, R-squared), average absolute error (Mean Absolute Error, MAE), and root mean square error (Root Mean Squared Error, RMSE).

The root mean square error [15] is one of the most common metrics for evaluating the accuracy of regression models. It is calculated as the average value of the squares of the difference between the observed and forecasting value. The smaller the MSE value, the better the model forecasts the data.

The coefficient of determination (R-squared) [15] measures the proportion of variation in responses explained by the model. It takes a value between 0 and 1, where 1 indicates a perfect model that fully explains the variation in the data. Using R-squared, you can compare different models and determine their ability to explain the data.

The mean absolute error measures the mean absolute value of all the differences between the observed and forecasting values. It allows you to get an idea of the average accuracy of the model, regardless of whether the forecasting values are exceeded or underestimated.

The root mean square error (RMSE) is an interpreted version of MSE, which is calculated as the square root of the root mean square error. It measures the average error in the same units as the original data, making it more interpretable.

These metrics help evaluate the performance and accuracy of Linear Regression and Random Forest models. The use of the most suitable metric depends on the specific context of the task and the requirements of the researcher [14, 15].

Based on the results obtained (Figure 2), it is possible to draw a conclusion about which model is better suited for forecasting electricity consumption in this context, which can be useful for further analysis and decision-making in the field of energy.

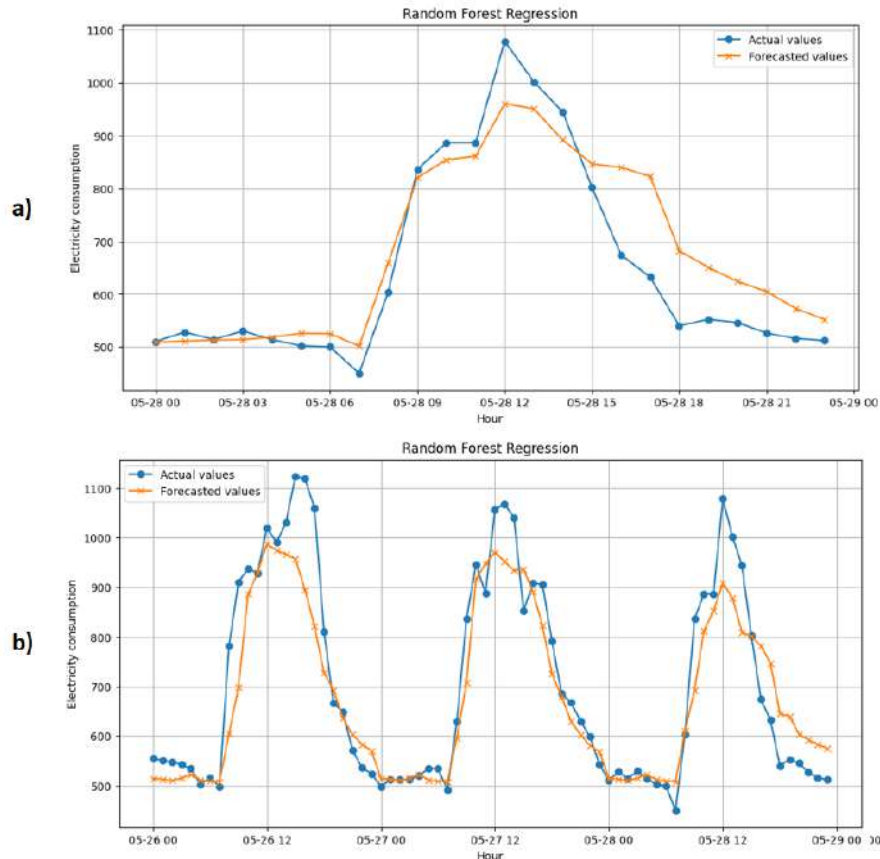


Figure 2 : a) forecasted and actual values for 24-hour-ahead forecasting using the RFR model; b) forecasted and actual values for forecasting 72 hours ahead using the RFR model

As a result of forecasting electricity consumption using random forest (RFR) and linear regression (LR) models, there is a difference in forecasting accuracy for different forecasting periods from 6 to 168 hours.

The random forest model shows low values of mean absolute error (MAE), root mean square error (RMSE), and mean percentage absolute error (MAPE) for short-term forecasts (6 and 12 h), indicating its good accuracy. However, when forecasting for longer periods (24, 72, and 168 hours), these metrics increase, indicating poorer performance of the model at longer time intervals.

Compared to the random forest model, linear regression exhibits higher values for all three metrics regardless of the forecast period. In addition, there is a tendency for errors to increase with the length of the forecast period, which may indicate the limited ability of linear regression to accurately forecast over a long period of time.

In general, the results (Table 1) indicate the superiority of the random forest model for short-term forecasting, while for long-term forecasting, it may be useful to use other models or approaches since both considered models show certain limitations in accuracy at long distances in time.

Table 1

The forecast results

Model	Metrics	6 hours	12 hours	24 hours	72 hours	168 hours
RFR	MAE	14.47	30,138	59.5	61,42	67,106
	RMSE	15,978	36,261	79.48	90,096	86,77
	MAPE	4,995	10.89	9.59	7.89	9.235
LR	MAE	84,689	164,235	131.62	173.04	177.69
	RMSE	85,225	222,036	193,419	242.64	245.41
	MAPE	18,180	20.58	16.78	20,233	21,211

In conclusion, the developed models tailored to various time intervals will be utilized to construct a subsystem for launching forecasting models within the decision support system (DSS), ensuring effective management of electricity consumption and facilitating informed decision-making processes.

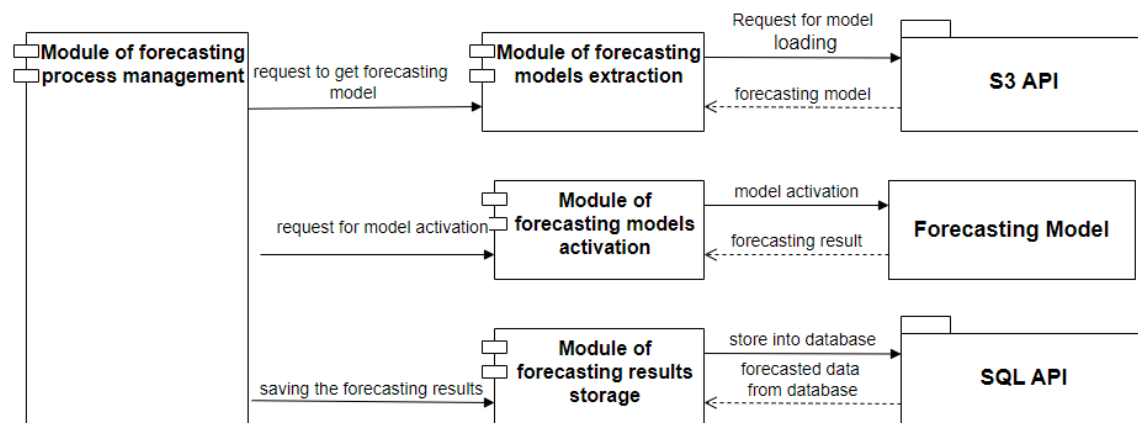
4. Forecasting software development

The forecasting software is designed to automatically run forecasting models to provide stable and accurate predictability at set time intervals, which is critical to ensuring continuity of electricity supply and efficient resource allocation.

This architecture is a complex system of management and application of forecasting models aimed at efficiently using resources and optimizing the forecasting process. In this system, there is interaction between several key modules, each of which performs a specific function within the forecasting process.

In general, this forecasting software architecture (Figure 4) is designed with the optimal use of resources in mind and provides an efficient management process and use of forecasting models. Its structured organization allows for reliable and fast implementation of forecasting with minimal loss of resources.

Figure 4 : Forecasting software architecture



The first module, namely the "Module of the forecasting process management", is responsible for coordinating and controlling the activation process of forecasting models. This module functions as an entry point for model activation requests, sending them to the forecasting model activation module.

The second "Module of forecasting models extraction" performs the task of getting the required forecasting models using the S3 API. It acts as an intermediary between the stored models and other modules of the system, providing access to the models using a defined protocol.

The "Module of forecasting models activation" is responsible for the activation process of the models obtained through the previous module. This module coordinates the initialization and loading of models for further use in the forecasting process.

The "Module of forecasting results storage" provides interaction of the system with the database, where forecasting results and other important data are stored. It writes and reads data from the database using the SQL API, ensuring reliable and efficient work with information.

The software was developed using the Python programming language due to its versatility and a rich arsenal of available libraries that facilitate the development process. Amazon S3 API ensures storage and efficient management of forecasting models, providing reliable storage and fast access to models. Using containerization via Docker provides a standardized environment for the module, while using Linux Cron allows you to automate the running process on a schedule. It actively interacts with the model repository hosted on Amazon S3 to obtain the required forecasting models.

After the forecasting models are activated and executed, the forecasting results or error information are written to the database. This process, which includes acquiring the models, running them, and storing the results or activation errors, is critical to ensuring the accuracy and reliability of the forecasting.

In the process of developing a system for activating forecasting models for managing a hybrid power grid, special attention was paid to the structuring and organization of the database. This database provides reliable and efficient storage of important data associated with each model run. After analyzing literary sources, it was decided to create two main tables in the database.

The first table, Model List, contains information about each moddwdel, including its S3 storage location, test data ID, and launch interval. The second table, Model Activation Results, is designed to track each model run and contains details of the execution process, including status, error descriptions, and results obtained.

5. Experiments

Before testing the developed software for the models, which were created in section 3 and stored in *.py files on an S3 Bucket, it is worth considering the process of automatic activation of models, which occurs through the configuration of tasks in the Linux Cron system.

Automatic activation is provided to run forecasting models, which is carried out by configuring tasks in the Linux Cron system. According to these tasks, the activation module runs at the specified time schedule, initiating a request to the model repository hosted on S3. At this stage, the necessary models are selected for further execution.

After receiving the appropriate models, the Module of forecasting models activation runs them to perform forecasting based on the incoming data. The module then processes the obtained results, including forecasted data and possible performance metrics, for further analysis.

The process of activating models includes not only starting and processing data but also registering these results in a database. Other modules of the system for analysis and development of optimal power grid management strategies subsequently use this database.

In case of errors detected during the activation of models, such errors are recorded for further analysis and possible optimization of the process in the future. This contributes to the continuous improvement of the management system and ensures its optimal functioning.

The obtained results (Figure 5, 6), stored in the database, serve as a basis for the operation of the subsequent modules of the system. This data is analyzed and used to make accurate and efficient decisions regarding power grid management. Thus, the activation module of forecasting models plays a key role in the operation and optimization of the power grid management system.

Id	ModelId	Dataset	Status	LastError	Results
15	1	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
16	2	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
17	1	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
18	2	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
19	1	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
20	2	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
21	1	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
22	2	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
23	1	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
24	2	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZp
25	1	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZpbmcu
26	2	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZpbmcu
27	1	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZpbmcu
28	2	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZpbmcu
29	1	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZpbmcu
30	2	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZpbmcu
31	1	NULL	COMPLETE	NULL	c3RyID0gcIIiUkCrATigKJEICAgICAgIMWShWtlcmFzLnNyYy5zYXZpbmcu

Figure 5: Results of forecasting models activation

Based on the testing results, errors occurred when the model files were missing and when there was a lack of access to the necessary resources required for forecasting. However, in cases where the models were stored in the repository, they were all successfully launched, confirming the effectiveness of the automatic model activation process.

Begin Transaction					Text	Filter	Sort	Column
id	model_id	Date	Forecast					
1	2	2021-05-28 00:00:00	509,16					
2	2	2021-05-28 01:00:00	511,08					
3	2	2021-05-28 02:00:00	512,88					
4	2	2021-05-28 03:00:00	513,68					
5	2	2021-05-28 04:00:00	518,44					
6	2	2021-05-28 05:00:00	525,7					
7	2	2021-05-28 06:00:00	524,94					
8	2	2021-05-28 07:00:00	501,54					
9	2	2021-05-28 08:00:00	659,16					
10	2	2021-05-28 09:00:00	820,26					
11	2	2021-05-28 10:00:00	854					
12	2	2021-05-28 11:00:00	860,72					
13	2	2021-05-28 12:00:00	959,88					
14	2	2021-05-28 13:00:00	950,54					
15	2	2021-05-28 14:00:00	891,6					
16	2	2021-05-28 15:00:00	846,06					

Figure 6: The forecasting results stored in the database

The developed software is used as a subsystem in the decision support system of microgrid energy management. The forecast data stored in the database have timeframes for convenient usage and analysis, enabling timely and effective decision-making regarding electricity management. This facilitates informed actions in the field of electric power engineering.

Conclusion

As a result of the study the software for electricity consumption was developed. It is designed to activate forecasting models that have been previously trained and stored in S3 storage. The developed forecasting software, integrated as a forecasting subsystem within the decision support system of microgrid energy management, showcases exceptional accuracy and efficiency in predicting electricity demand. The comparison between random forest method and linear regression highlights the former's superiority in forecasting accuracy. Computational experiments conducted over various time intervals demonstrate the robustness of the developed models.

In conclusion, while the successful implementation of the forecasting software and rigorous testing of forecasting models are noteworthy achievements, it's imperative to elucidate the practical applications and potential scalability beyond the S3 storage framework. The developed subsystem stands as a critical component of the decision support system, enriching decision-making processes in electricity management and offering promising avenues for optimizing energy network operations and ensuring reliable power grid management. To extend its utility to broader domains, the software can be adapted to different sectors such as city-wide power networks, industries, or residential communities. Integration with real-time data from smart meters and IoT devices can enhance performance. Scalability, renewable energy integration, and decision support capabilities are key areas for further development, ensuring effective energy management and sustainability across diverse contexts.

References

- [1] Bankruptcy Proceedings for an International Doctorate Contract within the Scope of the WEC4Ports Project: A Hybrid Wave Energy Converter for Ports. Notice (excerpt) no. 20317/2023. Official Gazette of Ukraine, Series II. October 24, 2023. URL: <https://files.diariodarepublica.pt/2s/2023/10/206000000/0017800178.pdf> (Accessed: March 18, 2024).
- [2] Shendryk, V., et al. "Modeling Techniques of Electricity Consumption Forecasting." Presented at the 9th International Conference of the Indonesian Chemical Society (ICICS) 2021: Toward a Meaningful Society, Mataram, Indonesia, 2022. URL: <https://doi.org/10.1063/5.0100123> (Accessed: March 18, 2024).

- [3] Chen, Q., et al. "Monthly Electricity Consumption Forecasting." In *Data Analytics in Power Markets*. Singapore, 2021. Pages 43–69. URL: https://doi.org/10.1007/978-981-16-4975-2_4 (Accessed: March 18, 2024).
- [4] Shendryk, V., et al. "Short-term Solar Power Generation Forecasting for Microgrid." Presented at the 2022 IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC), Kyiv, Ukraine, October 4–7, 2022. URL: <https://doi.org/10.1109/saic57818.2022.9922982> (Accessed: March 18, 2024).
- [5] Ventriglia, R. M., et al. "Applying Machine Learning to the Fuel Theft Problem in Pipelines." *Journal of Pipeline Systems Engineering and Practice*, vol. 14, no. 2, 2023. URL: <https://doi.org/10.1061/jpsea2.pseng-1374> (Accessed: March 18, 2024).
- [6] Ye, X., Liu, R., Yue, Z. "Structured Knowledge Base Q&A System Based on TorchServe Deployment." *Journal of Physics: Conference Series*, vol. 2078, no. 1, 2021, p. 012017. URL: <https://doi.org/10.1088/1742-6596/2078/1/012017> (Accessed: March 18, 2024).
- [7] "ECMC Rule-based Software Module for Defect Prediction using Support Vector Balanced Data." *International Journal of Recent Technology and Engineering*, vol. 8, no. 6, 2020, pp. 5219–5223. URL: <https://doi.org/10.35940/ijrte.f1103.038620> (Accessed: March 18, 2024).
- [8] Gadban, F., Kunkel, J. "Analyzing the Performance of the S3 Object Storage API for HPC Workloads." *Applied Sciences*, vol. 11, no. 18, 2021, p. 8540. URL: <https://doi.org/10.3390/app11188540> (Accessed: March 18, 2024).
- [9] Karunanidhi, V. "Understanding the TensorFlow Serving API: Deploying TensorFlow Models to a Web Application." Berkeley, CA, 2020. URL: https://doi.org/10.1007/978-1-4842-6699-1_7 (Accessed: March 18, 2024).
- [10] Canela, M. Á., Alegre, I., Ibarra, A. "Trend and Seasonality." In *Quantitative Methods for Management*. Cham, 2019. Pages 93–101. URL: https://doi.org/10.1007/978-3-030-17554-2_10 (Accessed: March 18, 2024).
- [11] Zheng, D., et al. "Short-term Renewable Generation and Load Forecasting in Microgrids." In *Microgrid Protection and Control*, 2021, pp. 57–96. URL: <https://doi.org/10.1016/b978-0-12-821189-2.00005-x> (Accessed: March 20, 2024).
- [12] Mantas, C. J., et al. "A Comparison of Random Forest-Based Algorithms: Random Credal Random Forest vs. Oblique Random Forest." *Soft Computing*, vol. 23, no. 21, 2018, pp. 10739–10754. URL: <https://doi.org/10.1007/s00500-018-3628-5> (Accessed: March 18, 2024).
- [13] Fahrmeir, L., et al. "Generalized Linear Models." In *Regression*. Berlin, Heidelberg, 2021, pp. 283–342. URL: https://doi.org/10.1007/978-3-662-63882-8_5 (Accessed: March 18, 2024).
- [14] Damanhuri, A., Mukarram, A. "Ideal Data to Determine Accurate Fajr Time." *Jurnal Mathematics MANTIK*, vol. 8, no. 1, 2022, pp. 28–35. URL: <https://doi.org/10.15642/mantik.2022.8.1.28-35> (Accessed: March 18, 2024).
- [15] Kuhlitz, D. A., et al. "Development of Predictive Models for Boar Semen Quality." *Theriogenology*, vol. 134, 2019, pp. 129–140. URL: <https://doi.org/10.1016/j.theriogenology.2019.05.024> (Accessed: March 18, 2024).
- [16] Barrak, A. F., Petrillo, F., Jaafar, F. "Serverless Machine Learning: A Systematic Mapping Study." *IEEE Access*, 2022, p. 1. URL: <https://doi.org/10.1109/access.2022.3206366> (Accessed: March 18, 2024).
- [17] Parfenenko, Y. V., et al. "Comparison of Short-term Forecasting Methods of Electricity Consumption in Microgrids." In *Radio Electronics, Computer Science, Control*, 2023, Issue 1, p. 14. URL: <https://doi.org/10.15588/1607-3274-2023-1-2> (Accessed: March 18, 2024).
- [18] Karyeyen, A., Çetinkaya, N. "Ensuring Energy Balance for Sudden Demand Changes in Smart Grids." *International Journal of Energy Studies*, 2023. URL: <https://doi.org/10.58559/ijes.1266296> (Accessed: March 18, 2024).
- [19] Zaharia, M. "Designing Production-Friendly Machine Learning." *Proceedings of the VLDB Endowment*, vol. 14, no. 13, 2021, p. 3420. URL: <https://doi.org/10.14778/3484224.3484241> (Accessed: March 18, 2024).
- [20] Siegel, A. F., Wagner, M. R. "Chi-Squared Analysis." In *Practical Business Statistics*. 2022, pp. 531–547. URL: <https://doi.org/10.1016/b978-0-12-820025-4.00017-8> (Accessed: March 18, 2024).

Research on Enhancing Power Generation Forecasts with Real-Time Machine Learning Models

Yulii Horichenko¹, Anzhelika Parkhomenko^{1,2}, Oleg Pozdnyakov¹, Artem Tulenkov¹, Olga Gladkova¹ and Andriy Parkhomenko¹

¹ National University Zaporizhzhia Polytechnic, 64, Zhukovskogo str., Zaporizhzhia, 69063, Ukraine

² University of Applied Sciences and Arts, 23, Otto-Hahn str., Dortmund, 44227, Germany

Abstract

Intelligent methods for optimizing energy use are rapidly gaining popularity as the world moves towards sustainable energy solutions. This study delves into enhancing the efficiency of alternative energy usage by improving of the forecasting model by integrating data collected in real time from local weather stations, leading to more accurate and localized forecasting of power generation. Significant focus is placed on the development and integration of cost-effective, custom-built measurement systems for wind speed and solar irradiation. By integrating real-time data from local meteorological stations with advanced machine learning methods, the study proposes a new approach that combines broad weather forecasts with precise local conditions to predict power generation more accurately. The research emphasizes the scientific novelty of the model, which combines real-time data from local meteorological stations with Long Short-Term Memory model. It also highlights the practical significance in improving the reliability and efficiency of alternative energy use in home automation systems.

Keywords

Home automation system, real-time machine learning, artificial neural networks, incremental learning, adaptive learning, power generation, forecasting, LSTM, OpenHAB

1. Introduction

In today's world, saving energy is becoming more and more important. Many people are looking for ways to use less electricity from the traditional power grid to save money and help the environment. One popular way to do this is by using alternative energy sources, like solar or wind power. However, the effective use of power from alternative sources requires advanced management strategies. There is a lot of discussion around this topic, so many authors have proposed their own methods for solving this problem. Existing works propose three main categories of energy optimization strategies: the optimization of electrical appliance usage schedule [1, 2, 3, 4, 5, 6, 7, 8], forecasting electricity consumption [9, 10, 11, 12, 13, 14, 15, 16, 17], and forecasting power generation [18, 19, 20]. Each of these approaches has its own advantages and limitations, which contribute to the overall goal of achieving energy efficiency from alternative sources.

The study [21] introduced a complex intelligent method for controlling energy consumption in home automation systems (HAS). This novel approach comprises two strategies: forecasting power generation and optimizing the schedule of electrical appliance usage. Consequently, homeowners can use energy more efficiently, saving money and reducing environmental impact. To facilitate this, an intelligent support subsystem was developed, providing residents with

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ gorichenko.yuliy@gmail.com (Y. Horichenko); parkhomenko.anzhelika@gmail.com (A. Parkhomenko); oleg.pozdnyakov.ua@gmail.com (O. Pozdnyakov); aetulenkov@gmail.com (A. Tulenkov); gladolechka@gmail.com (O. Gladkova); andriy2872073@gmail.com (A. Parkhomenko)

🆔 0009-0004-4506-9631 (Y. Horichenko); 0000-0002-6008-1610 (A. Parkhomenko); 0009-0006-3955-802X (O. Pozdnyakov); 0000-0003-4863-4144 (A. Tulenkov); 0000-0002-6834-2854 (O. Gladkova); 0000-0001-8265-0530 (A. Parkhomenko)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

recommendations for the efficient use of electrical appliances. While the developed Long Short-Term Memory (LSTM) model [21] showed good predictive capabilities in forecasting power generation, a significant limitation arises from its reliance on data from online weather services. The effectiveness of these forecasts is often compromised, as they may not accurately reflect the actual weather conditions where the HAS is located. This can cause issues for the developed intelligent support subsystem [21] while forecasting power generation and providing recommendations.

Therefore, the task of enhancing the accuracy of the forecasting model by incorporating real-time data is urgent.

The goal of the work is the research and practical implementation of methods and tools for real-time data collection, as well as methods for real-time machine learning.

2. Study of real-time data collection features for power generation forecasting

2.1. Analysis of management architectures for alternative energy sources

The initial phase of this study delves into how inverters are used within alternative energy sources, focusing on their architectural integration. By exploring different inverter architectures, the aim is to see how they can integrate with the power grid and other energy sources.

Inverters play a crucial role in integrating alternative energy, transforming the direct current (DC) produced by solar panels and wind turbines into the alternating current (AC) used in homes and the power grid. This transformation is key because it makes the generated electricity compatible with usual household appliances and the wider electrical system. But solar inverters do a lot more than just convert DC to AC. They also adjust the electricity to the right voltage and sine wave frequency, making sure it is safe for home use or to be sold out on the electricity grid. Furthermore, these inverters incorporate safety features, ensuring a secure energy supply [24].

It is also important to understand the different types of inverters in order to fully appreciate their role in adopting alternative energy. There are three main types: Stand-alone, Grid-tie and Hybrid inverters.

Stand-alone inverters are the heart of off-grid solar systems. They convert DC electricity from batteries, which get charged by the solar panels, into AC that can be used for a variety of needs (Fig. 1). This type is perfect for places without access to the traditional grid, providing a reliable source of power. They have features that protect batteries from overcharging and help keep the power supply steady, making them crucial for continuous energy in off-the-grid spots [24, 25].

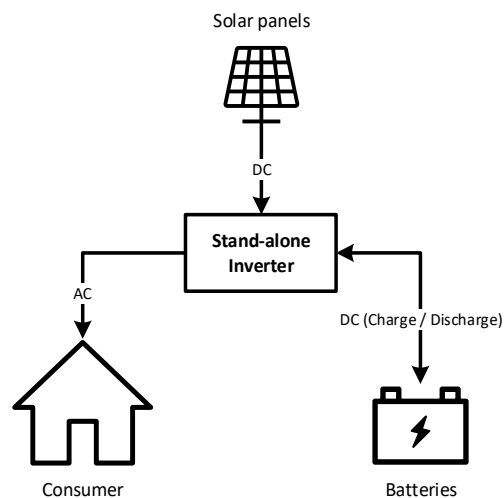


Figure 1: Stand-alone inverter configuration for off-grid scenarios

Grid-tie inverters connect solar systems to the utility grid. They transform the DC electricity from solar panels into AC that matches the external grid's requirements. A key function of these inverters is to synchronize the solar power with the grid's frequency and phase (Fig. 2). This not only lets homes use solar power but also lets them sell extra electricity back to the grid to get credits for this extra energy [24, 25].

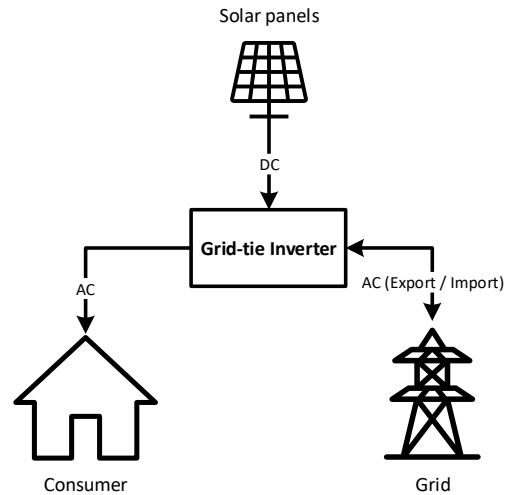


Figure 2: Grid-tie inverter integration with public electricity grid

Hybrid inverters combine the best features of both stand-alone and grid-tie inverters. They are capable of using power from solar panels, batteries and the external grid, supporting systems that aim for independence from the traditional grid while also ensuring efficient energy use (Fig. 3). These inverters allow homes to store excess solar energy for use during periods of high demand or when the grid is unavailable. This type of inverter is particularly useful in areas with unreliable grid service, as it makes energy use more efficient and reduces reliance on the grid.

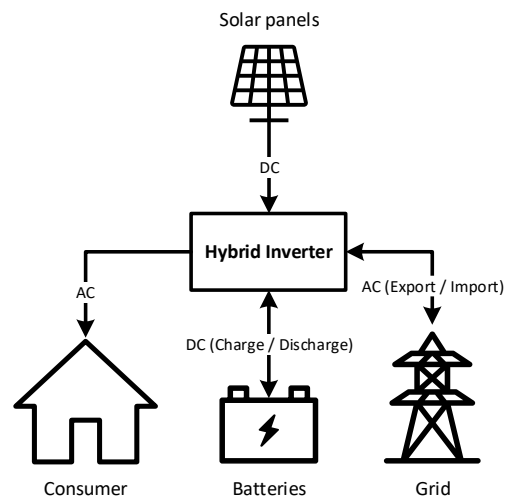


Figure 3: Hybrid inverter configuration for integrated energy management

In work [21], a real HAS was examined, with a focus on its integration with alternative energy sources and battery backup system. It has a hybrid setup, including a hybrid inverter connected to the external power grid, battery storage and renewable energy sources such as solar panels and wind turbines. These batteries play a critical role as a backup solution in scenarios where both self-generated electricity and the external grid are unavailable. Under normal conditions, they maintain a charged state, acting as a buffer and enabling efficient energy management within the system.

Additionally, this system incorporates Open Home Automation Bus (OpenHAB), a home automation software that allows homeowners to control a wide range of Internet of Things (IoT) devices. This means that people can easily control the environment in their home, making it more comfortable and able to respond to their needs more effectively.

2.2. Analysis and implementation of real-time data collection from meteorological stations

Recent study [21] has shown the correlation between weather data and power generation while implementing an LSTM forecasting model. Specifically, wind speed (measured in meters per second) and solar irradiation (measured in watts per square meter) have been identified as essential parameters for the forecasting model. To collect this data, it is necessary to install a meteorological station near the HAS. Such an installation not only ensures the availability of real-time data but also enhances the model's predictive accuracy by using local weather conditions.

A detailed review of various online stores offering meteorological stations was carried out to find the best solution. The research showed a strong preference for all-in-one weather stations. These stations can track a variety of weather information, such as wind speed and direction, temperature, solar irradiation, air pressure etc. Despite their capabilities, the primary drawback of these stations lies in their high cost, making them a less suitable option for budget limited projects. As a result, a more affordable alternative is to buy wind and solar irradiation sensors separately. Although this approach offers a smaller range of data, it still meets the essential requirements for the forecasting model.

There is a wide variety of wind sensors available on the market at affordable prices. They are designed to accurately measure wind speed and direction, making them an ideal choice for collecting the necessary wind data for a forecasting model.

To collect wind speed data in real-time, the wind speed sensor “CWT-SWC-C-RS485” was selected for its price and efficiency. This wind sensor uses the RS485 standard, which OpenHAB does not support. To resolve this, the data is sent to an ESP8266 microcontroller with “Tasmota” firmware installed, acting as a Modbus bridge. Subsequently, OpenHAB processes this information and stores it in a local MySQL database. This measurement system represents a seamless integration of the sensor with HAS, enabling advanced wind speed monitoring and data collection (Fig. 4).



Figure 4: Architecture of custom wind measurement system

The custom wind speed measurement system has now been successfully implemented and is actively collecting local wind speed data for the database. On Figure 5, a sample of the data from the database is displayed.

				Time	Value
☐				2024-01-28 23:07:46	2
☐				2024-01-28 23:07:52	1.2
☐				2024-01-28 23:07:58	1.4
☐				2024-01-28 23:08:04	2
☐				2024-01-28 23:08:09	2.1
☐				2024-01-28 23:08:16	0.5
☐				2024-01-28 23:08:28	0
☐				2024-01-28 23:08:34	1.9
☐				2024-01-28 23:08:46	1.5
☐				2024-01-28 23:08:52	1.6
☐				2024-01-28 23:08:58	0.9
☐				2024-01-28 23:09:04	0.7
☐				2024-01-28 23:09:09	0.1
☐				2024-01-28 23:09:16	0.2
☐				2024-01-28 23:09:22	0.3
☐				2024-01-28 23:09:27	1
☐				2024-01-28 23:09:34	0.9
☐				2024-01-28 23:09:40	1
☐				2024-01-28 23:09:58	1.3
☐				2024-01-28 23:10:04	0.6
☐				2024-01-28 23:10:09	0.4
☐				2024-01-28 23:10:16	0.1
☐				2024-01-28 23:10:28	0
☐				2024-01-28 23:10:33	0.5
☐				2024-01-28 23:10:40	1.3

Figure 5: Sample wind speed data collected from the local wind speed measurement system

The OpenHAB platform enables real-time monitoring of wind speed data (Fig. 6) and offers access to historical data charts (Fig. 7). This setup provides instant insights into current wind conditions and supports the analysis of wind speed trends over time. Additionally, OpenHAB collects wind speed data from online weather service OpenWeatherMap [23], which enables to compare the local weather data with online forecasts.



Figure 6: Real-time wind speed monitoring via the OpenHAB interface

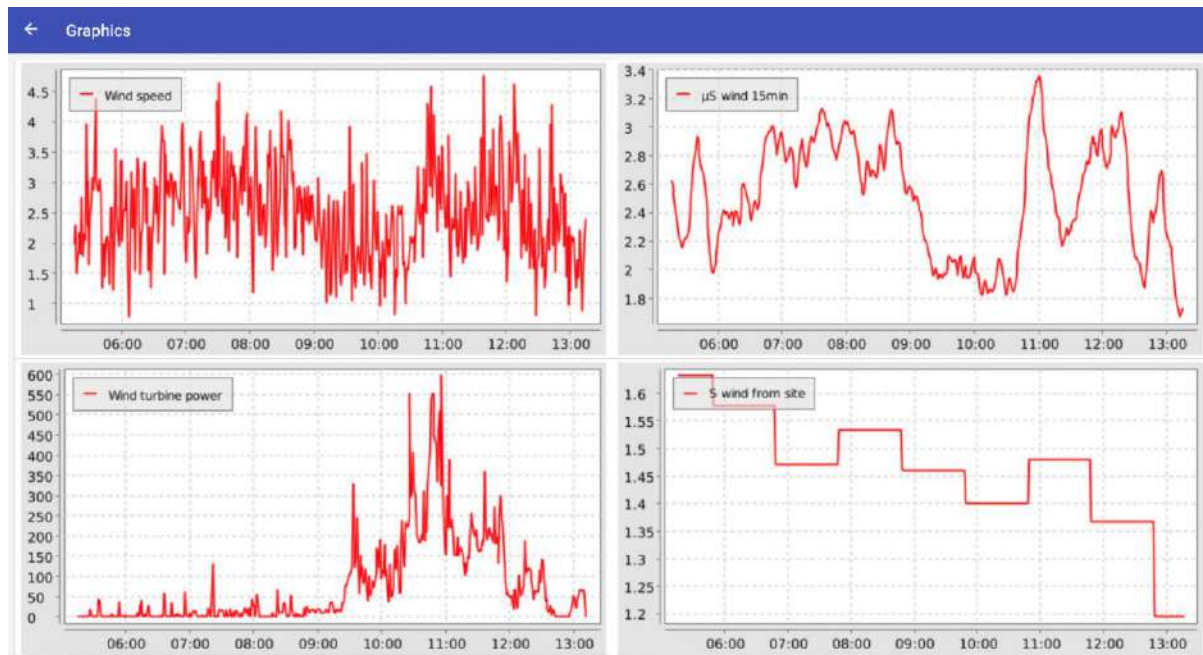


Figure 7: Historical wind speed data charts in the OpenHAB interface

Subsequent research involved analyzing data obtained from online weather service, which was used for forecasting power generation. In Figure 8, a chart shows a comparative analysis of historical wind speed data, derived from both a local meteorological station and online weather service. The local meteorological station data, denoted by the blue line with error bars, shows the mean wind speed for each hour along with the standard deviation, indicating variability within the hour. The error bars show how much the wind speed can vary over time. In contrast, the data from the online weather service OpenWeatherMap [23] is plotted as an orange line.

The comparative analysis between these two datasets revealed some major differences, suggesting that incorporating local meteorological station data could enhance the accuracy of power generation forecasts. This visual analysis not only highlights the temporal dynamics of wind speed but also serves as a crucial tool for evaluating the reliability of wind speed data from different sources. Such comparisons are essential for applications where accurate weather data is crucial for predicting power generation and providing recommendations.

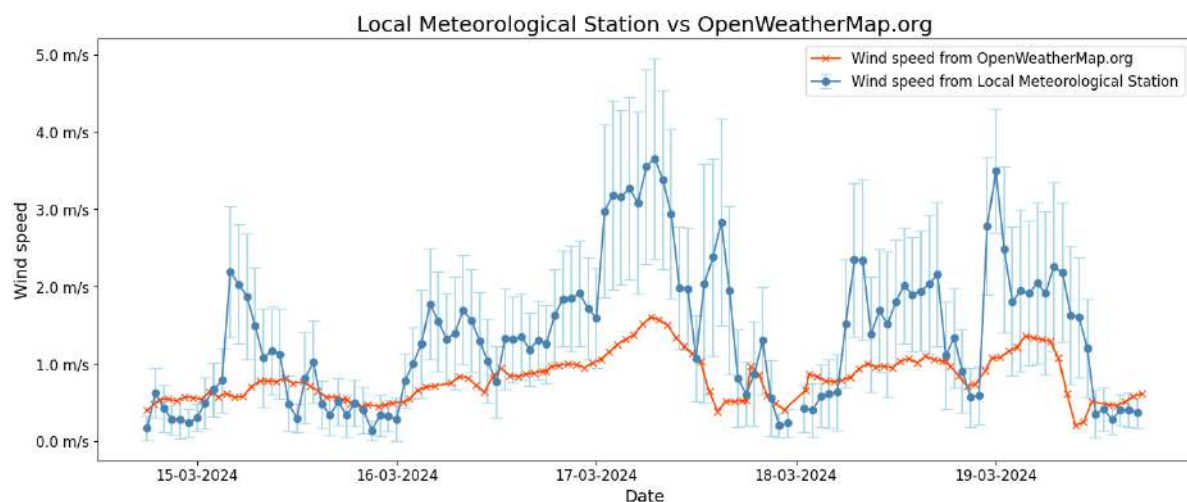


Figure 8: Comparative analysis of hourly historical wind speed data between a local meteorological station and an online weather service

During the search for a suitable solar irradiation sensor, it was discovered that the available sensors on the market are too expensive. This discovery suggested an idea for designing and building a custom solar irradiation measurement system that would meet the requirements of the study. This solution is expected to provide accurate data on solar irradiation at a lower cost than commercial sensors.

To calculate the level of solar irradiation, a measurement system was suggested that involves a solar controller charger with pulse width modulation (PWM) technology, a solar panel and a battery. By measuring the electrical output from the solar panel and considering the panel's surface area, the level of solar irradiation can be calculated (Fig. 9). The solar controller charger plays a crucial role in this process by managing the charging cycle and ensuring that the energy transfer to the battery remains within safe bounds.

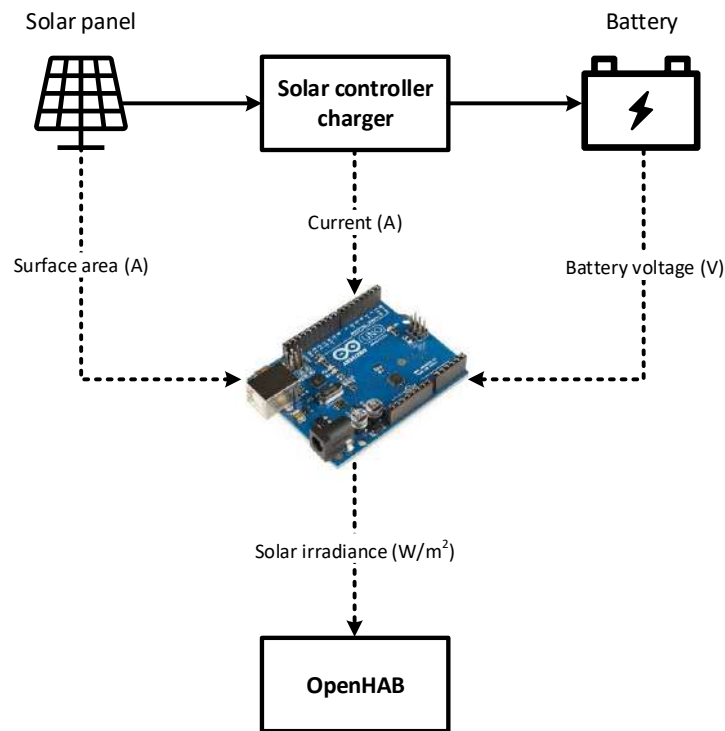


Figure 9: Architecture of custom solar irradiation measurement system

Initially, the chosen solar controller charger is connected to both the solar panel and a battery. An Arduino is used to monitor the battery's voltage, which helped to determine the solar panel's current power output using formula (1):

$$P = V \times I, \quad (1)$$

where P represents power in watts (W), V is the voltage in volts (V) and I stands for the current in amperes (A).

Subsequently, once the power output and the surface area of the panel are known, the level of solar irradiation can be calculated using formula (2):

$$I = \frac{P}{A}, \quad (2)$$

where I is the solar irradiance in watts per square meter (W/m²).

This solution will provide real-time data on local solar irradiation, necessary for enhancing the accuracy of the forecasting model without the use of specialized meteorological stations.

3. Related works

To enhance the accuracy of power generation forecasting, the next step is to thoroughly review related works in this area. This includes examining different techniques and machine learning models, applied to a range of energy sources.

In the work [26], the authors introduced an online domain adaptive learning approach, enhanced with the AdaBoost algorithm, for solar power forecasting. This model is specially designed for its ability to adapt to changing weather conditions, thereby significantly enhancing the accuracy of solar energy output predictions. Unlike traditional batch learning models, which become static after training, this innovative approach allows for continuous adaptation to new data without the need for retraining. This makes it particularly suitable for the unpredictable nature of solar irradiation.

This adaptive learning model [26] demonstrates remarkable performance improvements over traditional forecasting methods, such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), Extreme Learning Machines (ELM), Gaussian Mixture Regression (GMR) and the Persistence Model, across various datasets. For example, on one dataset, the proposed adaptive learning model achieved a Root Mean Square Error (RMSE) of 94.6843 W/m², far surpassing the RMSE values of GMR (106.7835 W/m²) and ANN (126.6187 W/m²). Furthermore, when tested on other datasets, the model's accuracy remained consistent. On another dataset, its RMSE measure of 44.21007 W/m² outperforms those of GMR (52.4752 W/m²) and SVM (60.7089 W/m²), underscoring an improvement of approximately 15% over GMR, which is the closest competitor. These findings highlight not only the model's enhanced accuracy but also its reliability in forecasting solar power output under varied meteorological conditions.

In the research [27], the Adaptive Learning Hybrid Model (ALHM) was introduced as a solution for solar intensity forecasting, crucial for the integration of renewable energy sources into smart grids. This model stands out for its ability to adaptively learn from new data over time, capturing both linear and nonlinear dynamics through its integration of Time-varying Multiple Linear Model (TMLM) and Genetic Algorithm Back Propagation Neural Network (GABPNN). Using an extensive dataset that includes various meteorological variables such as temperature, humidity, dew point, wind speed and precipitation, the model has demonstrated superior forecasting performance.

The proposed model [27] outperformed both ANN and SVM models in terms of Mean Absolute Percentage Error (MAPE) and RMSE. The proposed model achieved a MAPE of 13.68% and an RMSE of 16.95 W/m², indicating a higher forecasting accuracy and reliability. In contrast, ANN and SVM models reported higher MAPE values, demonstrating their lesser ability to forecast. One of the key advantages of the proposed model over ANN and SVM is its adaptive learning capability. It can adaptively learn from new data, improving its forecasting accuracy over time. This is particularly valuable in dynamic environments like solar intensity forecasting, where weather patterns and other influencing factors can change.

In the study [28], an innovative Adaptive Long Short-Term Memory (ALSTM) model was proposed to address the challenge of day-ahead forecasting of photovoltaic (PV) power generation. This model focuses particularly on overcoming the problem of concept drift, where the data distribution changes over time, making traditional models less effective. The main idea of the proposed is to enhance its adaptability by enabling it to continuously learn from new data as it arrives. This allows the model to maintain high forecasting accuracy even in the presence of concept drift.

The dataset used for evaluating the proposed model [28] included records from a PV plant over several months, incorporating both historical and newly-arrived data streams. Compared to other forecasting methods such as Persistence, Autoregressive Integrated Moving Average (ARIMA), k-Nearest Neighbors (k-NN) and the traditional offline LSTM model, the ALSTM demonstrated superior performance across various metrics. For example, when compared to the offline LSTM, the ALSTM showed an MSE reduction ranging from approximately 56.07% to 92.77%. This indicates a higher forecasting accuracy and its effectiveness in adapting to new data

and mitigating the impacts of concept drift, offering a significant advancement over traditional forecasting approaches.

In the work [29], an innovative methodological framework that incorporates incremental learning to improve the accuracy in the field of energy forecasting was introduced. This approach is the adoption of incremental learning techniques applied to a Multi-Layer Perceptron (MLP). This approach effectively addresses the limitations associated with traditional batch learning models by facilitating continuous learning from real-time data, thus ensuring the model's relevance and accuracy are maintained over time.

The effectiveness of the proposed method [29] was evaluated using real-world data from a microgrid located in Italy, which includes a multi-story building and a PV system. A comparative analysis between the performance of the proposed model and the traditional learning model clearly demonstrates the benefits of the new approach. The results represent a significant improvement in forecasting accuracy. The incremental learning MLP model reduces the MAE by 7.93% and the RMSE by 7.52%, compared to the traditional MLP model. This numerical evidence strongly supports the superiority of the proposed method. It highlighting its enhanced adaptability to changing data patterns and its increased predictive performance, making it an optimal choice for real-time energy forecasting applications.

In the study [30], a novel solar PV power generation forecasting model was introduced that combines different weather information to compensate for the lack of real-time power generation data. This approach uses Deep Neural Networks (DNN) for data fitting and LSTM networks for temporal forecasting. The model was rigorously tested using datasets from six solar power plants in Taiwan, covering diverse environmental conditions. The results demonstrated an exceptional forecasting accuracy of over 97% compared to traditional models like LSTM, DNN, SVM and Back-Propagation Neural Network (BPNN). Among these traditional models, the DNN-LSTM model showed superior performance, as evidenced by its lower Normalized Root Mean Square Error (nRMSE) and Normalized Mean Absolute Error (nMAE). This underscores its robustness and reliability in comparison to conventional forecasting methodologies.

In the study presented in [31], a new approach using transfer learning for predicting day-ahead PV power is introduced. This methodology relies on the fundamental idea of leveraging the deep learning models trained on large historical datasets from existing PV power plants to enhance the prediction accuracy for newly installed PV systems. The transfer learning framework improves prediction accuracy by applying patterns and insights extracted from extensive historical data to new situations where the historical data is limited.

The datasets used in the study [31] include hourly historical data from two PV power farms that are located in close proximity to each other. The first provides a more extensive dataset, while the second farm provides more recent data. A comparative analysis of the developed models: linear, dense, Convolutional Neural Network (CNN) and LSTM reveals the superior performance of the transfer learning models over their new and untrained counterparts. For example, the trained transfer LSTM model consistently outperformed their counterparts, with improvements in MAE, MSE and RMSE reaching up to 41.42%, 69.45% and 45.91% respectively. This demonstrates the potential of transfer learning in renewable energy forecasting applications.

The results of the machine learning models and techniques reviewed in related works, along with their comparisons to traditional models, are presented in Table 1.

The survey of related works [26, 27, 28, 29, 30 31] consistently underscores the advantage of real-time adaptive learning methodologies over traditional forecasting models in enhancing the accuracy of power generation predictions. These studies demonstrate that integrating adaptive learning techniques can greatly improve the forecasting process by allowing models to continuously update and adjust to new data. Specifically, the work presented in [31] offers a strong foundation for developing an improved predictive model that takes into account the architecture of a real HAS [21]. By integrating real-time learning and transfer learning techniques, this model has the potential to significantly improve the accuracy of power generation forecasts. This approach not only enhances accuracy but also improves the operational efficiency of energy systems in constantly changing environmental conditions.

Table 1**Comparative analysis of real-time learning models to traditional models**

Model / Technique used	Core features	Comparative improvement
Domain Adaptive Learning with AdaBoost [26]	High adaptability to changing conditions	Outperforms ANN, SVM, ELM, GMR models
Adaptive Learning Hybrid Model (ALHM) [27]	Adaptively learns from new data	Superior to ANN and SVM models
Adaptive Long Short-Term Memory (ALSTM) [28]	Effectively handles concept drift	Better than traditional LSTM model
Incremental Multi-Layer Perceptron (MLP) [29]	Continuous learning from real-time data	Enhances accuracy over traditional MLP model
DNN-LSTM Hybrid [30]	Utilizes varied weather information	Superior to LSTM, DNN, SVM, BPNN models
Transfer Learning LSTM [31]	Applies historical insights to new setups	Outperforms traditional linear, dense, CNN and untrained LSTM models

4. An improved two-stage predictive model for enhanced forecasting accuracy with real-time data

Following a detailed review of related works and considering the architecture of a real HAS [21], an improved two-stage predictive model has been proposed. This will enhance the accuracy of power generation predictions for solar panels and wind turbines using local weather data.

The first stage involves the development of a predictive model that aims to bridge the gap between general weather forecasts provided by online weather service and the specific local weather conditions in a given area. This stage uses a predictive model f to convert the general weather forecasts provided by an online service to local weather predictions. The general forecast serves as the input features while the local historical data serves as the targets, enabling a better understanding of how general weather forecasts correlate with local weather phenomena, that can be expressed as follows (3):

$$\hat{W}_l(t) = f(W_g(t); \theta), \quad (3)$$

where $W_g(t)$ represents the general weather data at time t , $\hat{W}_l(t)$ represents the predicted local weather conditions at time t , and θ are the parameters of the model. The model f is trained using historical data from both online service and local meteorological station. Once the model is trained, it can be used to generate forecasts for local weather conditions based on the input from an online service. This approach will provide more accurate predictions for specific areas where a real HAS is located.

The second stage involves the development of the LSTM model, which is trained exclusively on historical weather data collected from a local meteorological station. Using the accurate and relevant information provided by the local meteorological station, this model is specifically designed to predict power generation based on the predicted local weather conditions obtained from the first stage. The predictive model for power generation can be expressed as follows (4):

$$\hat{P}(t) = LSTM(\hat{W}_l(t); \phi), \quad (4)$$

where $\hat{P}(t)$ is the predicted power generation at time t , $\hat{W}_l(t)$ is the local weather prediction derived from formula (3), and ϕ represents the parameters of the LSTM model. This LSTM model

is specifically designed to predict power generation from these local weather predictions, thus enabling more accurate forecasts of power generation and taking full advantage of the understanding of local weather patterns established in the first stage.

The architecture of the proposed predictive model is illustrated in Figure 10, providing a visual representation of the two-stage model and its integration with real-time data for enhancing the accuracy of power generation forecasts.

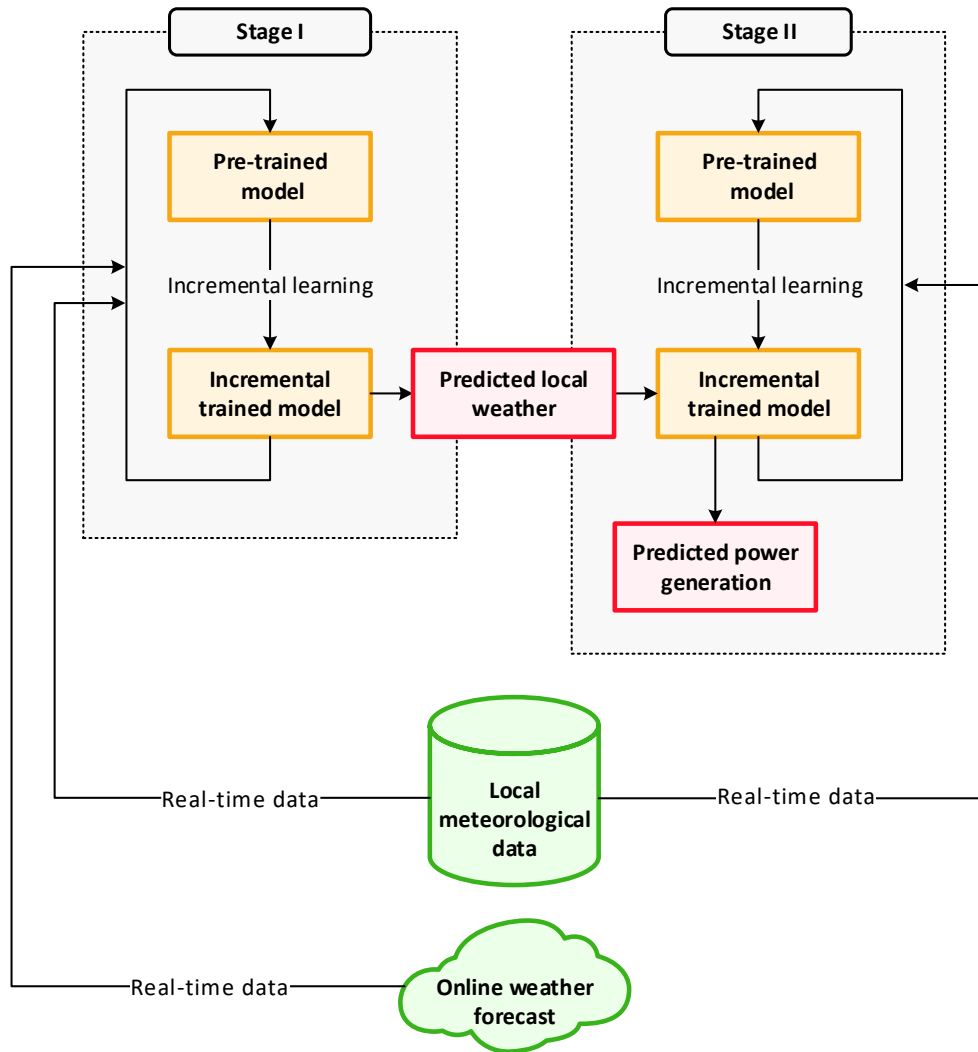


Figure 10: Architecture of the proposed two-stage predictive model incorporating real-time data for enhanced power generation forecasting

The data flow of the improved two-stage predictive model is illustrated in Figure 11, showing interactions between various components, including historical data from different sources and two predictive models. Significantly, it demonstrates the integration of historical weather and power generation data, which are crucial for training the first predictive model and the LSTM model, respectively, to accurately forecast power generation. The forecasted power generation is used to provide recommendations in the developed intelligent support subsystem [21].

Incorporating real-time data processing greatly enhances the accuracy of power generation forecasts. By updating the models in real-time with the latest data from the local meteorological station, the predictive models are guaranteed to adapt to the latest weather patterns. This real-time approach allows for continuous learning and adjustment, which will significantly improve the model's responsiveness to sudden weather changes and will enhance the accuracy of power generation predictions. The integration of real-time data will not only refine the models'

predictive capabilities but will also ensure that the system remains relevant and accurate over time, despite the inherent variability and unpredictability of weather conditions. This approach provides a robust framework for reliably forecasting power generation from renewable energy sources, leveraging the synergy between localized data collection and advanced machine learning techniques.

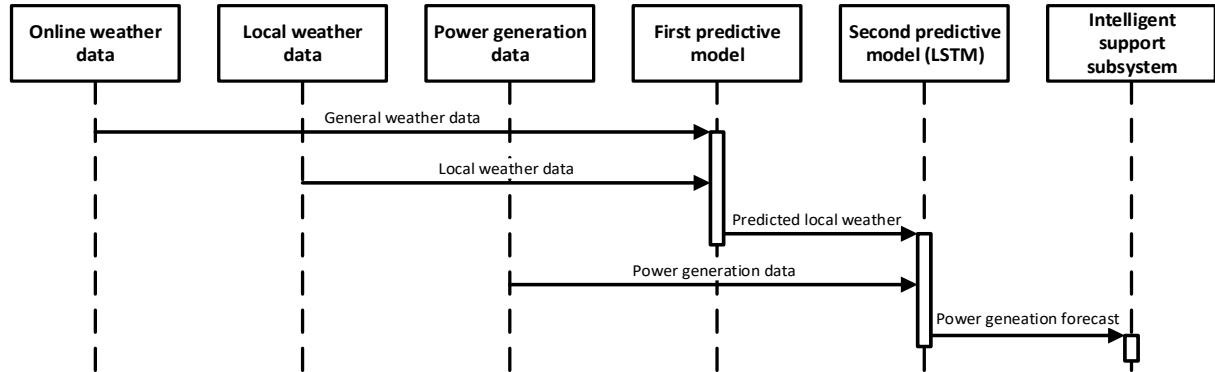


Figure 11: Sequence diagram of the improved two-stage predictive model incorporating real-time data for enhanced power generation forecasting

5. Conclusion

As a result of the conducted research, it has been demonstrated that integrating real-time local meteorological data with advanced machine learning techniques can significantly enhance the accuracy of power generation forecasts for alternative energy sources. The proposed two-stage predictive model, which combines general weather forecasts with specific local conditions, offers a more precise understanding of how weather impacts power generation. This approach not only overcomes the limitations of existing forecasting methods but also opens up the possibility for a more efficient and reliable use of alternative energy sources in HAS. By incorporating real-time data processing, the developed intelligent support subsystem [21] will be able to remain adaptable and responsive to sudden weather changes, thereby improving the sustainability and efficiency of energy use in HAS.

The scientific novelty of the work lies in the development of an improved two-stage predictive model that uses local weather data to improve forecasts of power generation from solar panels and wind turbines. The improved architecture of this model, which combines real-time data collection from local meteorological stations with advanced LSTM machine learning algorithms, represents a significant advancement in the field of alternative energy forecasting. Furthermore, the methodological approach of correlating general weather forecasts with local weather phenomena to enhance forecast accuracy introduces a new paradigm in predictive modeling for alternative energy sources.

The practical significance of this work is that it provides an effective way to improve the reliability and efficiency of using alternative energy sources, especially in HAS. With more accurate forecasts for power generation, homeowners can better manage their energy consumption, reducing their dependence on traditional power grids. This leads to cost savings and supports environmental sustainability by promoting the adoption of clean energy.

In future work, it is planned to implement the proposed two-stage model and then integrate it into a real HAS. This step will allow to evaluate and fine-tune the model within a real-world context, guaranteeing its effectiveness in optimizing energy management and forecasting capabilities. Additionally, exploring the potential of incorporating advanced artificial intelligence techniques to further improve the model's forecasting precision and operational performance is planned.

Acknowledgements

This work is partly carried out with the support of Erasmus + KA2 project WORK4CE “Cross-domain competences for healthy and safe work in the 21st century” (619034-EPP-1-2020-1-UA-EPPKA2-CBHE-JP) and DAAD project ViMUK.

References

- [1] J. Zhu, F. Lauri, A. Koukam, V. Hilaire, Scheduling optimization of smart homes based on demand response, in: Proceedings of the 11th IFIP International Conference on Artificial Intelligence Applications and Innovations, Bayonne, France, 2015, pp. 223-236. doi:10.1007/978-3-319-23868-5_16.
- [2] C. Deng, S. Zhang, W. Yang, W. Yao, J. Tan, Z. Liu, Two-stage optimization model for Smart House daily scheduling considering user perceived benefits, in: Proceedings of the International conference on Mathematics, modelling, simulation and algorithms, Chengdu, China, 2018, pp. 64-68. doi:10.2991/mmsa-18.2018.15.
- [3] Z. Qu, N. Qu, Y. Liu, X. Yin, C. Qu, W. Wang, J. Han, Multi-objective optimization model of electricity consumption behavior considering combination of household appliance correlation and comfort, Journal of electrical engineering and technology 13(5) (2018) 1821-1830. doi:10.5370/JEET.2018.13.5.1821.
- [4] I. Gupta, G. N. Anandini, M. Gupta, An hour wise device scheduling approach for demand side management in smart grid using particle swarm optimization, in: Proceedings of the 19th National power systems conference, Bhubaneswar, India, 2016, pp. 1-6. doi:10.1109/NPSC.2016.7858965.
- [5] Y. Zhou, Y. Chen, G. Xu, C. Zheng, M. Cheng, Home energy management in smart grid with renewable energy resources, in: Proceedings of the 16th International conference on Computer modelling and simulation, Cambridge, United Kingdom, 2014, pp. 351-356. doi:10.1109/UKSim.2014.46.
- [6] F. Iqbal, K. Iqbal, Optimal load scheduling using genetic algorithm to improve the load profile, ADRII Journal of engineering and technology 4(9) (2021) 1-15. doi:10.55058/adrijet.v4i9(3).621.
- [7] A. Thivy, K. Thamini, P. Narmadha, I.S. Vishaka, K.R. Vairamani, Demand side management in smart grid using heuristic algorithm, in: Proceedings of the 2nd International conference on Emerging enhancement in engineering and technology, Tiruchirappalli, 2016, pp. 361- 365. doi: 10.1109/TSG.2012.2195686.
- [8] E. Lee, H. Bahn, Electricity usage scheduling in Smart building environments using smart devices, The scientific world journal (2013) 11 p. doi:10.1155/2013/468097.
- [9] M. Taksir, S. Aktar Data-driven time series-based prediction in smart home appliance energy consumption, International journal of computer applications 178(15) (2018) 41-46. doi:10.5120/ijca2019918932.
- [10] R. E. Alden, H. Gong, C. Ababei, D.M. Ionel, LSTM forecasts for Smart home electricity usage, in: Proceedings of the 9th International conference on Renewable energy research and application, Glasgow, UK, 2020, pp. 434-438. doi:10.1109/ICRERA49962.2020.9242804.
- [11] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, Y. Hu, Short-term residential load forecasting based on resident behaviour learning, IEEE Transactions on power systems 33(1) (2017) 1087-1088. doi:10.1109/TPWRS.2017.2688178.
- [12] Y. Wang, N. Zhang, X. Chen, A short-term residential load forecasting model based on LSTM recurrent neural network considering weather features, Energies 14(10) (2021) 1-13. doi:10.3390/en14102737.
- [13] O. D. Diaz-Castillo, A. E. Puerto-Lara, J. A. Saenz-Leguizamon, V. Ducon-Sosa, Prediction of electrical energy consumption through recurrent neural networks, CEUR Workshop Proceedings, vol. 2992 (2021). Online http://ceur-ws.org/Vol-2992/icaiw_wssc_3.pdf.

- [14] K. Amarasinghe, D. L. Marino, M. Manic, Deep neural networks for energy load forecasting, in: Proceedings of the IEEE 26th International symposium on Industrial electronics, Edinburgh, UK, 2017, pp. 1483-1488. doi:10.1109/ISIE.2017.8001465.
- [15] S. Mahjoub, L. Chrifi-Alaoui, B. Marhic, L. Delahoche, Predicting Energy Consumption Using LSTM, Multi-Layer GRU and Drop-GRU Neural Networks, *Sensors* 22(1) (2022) 1-20. doi:10.3390/s22114062.
- [16] S. Emshagin, W. K. Halim, R. Kashef, Short-term Prediction of Household Electricity Consumption Using Customized LSTM and GRU Models, *arXiv preprint* 1(1) (2022) 1-11. doi:10.48550/arXiv.2212.08757.
- [17] P. B. Gopikrishna, J. A. Mathew, Power Consumption Analysis and Prediction of a Smart Home Using ARIMA Model, *SSRN* 1(1) (2021) 1-9 doi:10.2139/ssrn.3819512.
- [18] M. Beigi, H. B. Harchegani, M. Torki, M. Kaveh, M. Szymanek, E. Khalife, J. Dziwulski, Forecasting of power output of a PVPS based on meteorological data using RNN approaches, *Sustainability* 14(5) (2022) 1-12. doi:10.3390/su14053104.
- [19] J. Wojtkiewicz, M. Hosseini, R. Gottumukkala, T. L. Chambers, Hour-Ahead Solar Irradiance Forecasting Using Multivariate Gated Recurrent Units, *Energies* 12(1) (2019) 1-13. doi:10.3390/en12214055.
- [20] U. Singh, M. Rizwan, M. Alaraj, I. Alsaidan, A Machine Learning-Based Gradient Boosting Regression Approach for Wind Power Production Forecasting: A Step towards Smart Grid Environments, *Energies* 14(1) (2021) 1-21. doi:10.3390/en14165196.
- [21] Y. Horichenko, A. Parkhomenko, O. Pozdnyakov, A. Tulenkov, Y. Zalyubovskiy, O. Gladkova, A. Parkhomenko, Research and Software Implementation of Intelligent Method of Energy Consumption Control, *CEUR Workshop Proceedings*, vol. 3392 (2023). Online <http://ceur-ws.org/Vol-3392/paper9.pdf>.
- [22] Y. E. Horichenko, O. A. Pozdnyakov, A. V. Tulenkov, A. V. Parkhomenko, Research and practical implementation of intelligent methods and models for controlling power generation from alternative energy sources, *Scientific works of DonNTU [Naukovi pratsi DonNTU]*, 2(37) (2023) 22-32. doi:10.31474/1996-1588-2023-2-37-22-33.
- [23] Weather API – OpenWeatherMap, 2024. URL: <https://openweathermap.org/api>.
- [24] Understanding the differences between On-grid, Off-grid (Hybrid), and On-grid Solar Inverters with Energy Storage Systems, 2024. URL: <https://prolink2u.com/blogs/discover/understanding-the-differences-between-on-grid-off-grid-hybrid-and-on-grid-solar-inverters-with-energy-storage-systems>.
- [25] Z. Hyder, C. Lane, Compared: Grid-tied, off-grid, and hybrid solar systems, 2024. URL: <https://www.solarreviews.com/blog/grid-tied-off-grid-and-hybrid-solar-systems>.
- [26] H. Sheng, B. Ray, K. Chen, Y. Cheng, Solar Power Forecasting Based on Domain Adaptive Learning (by HANMIN SHENG), *IEEE Access* 8(1) (2020) 198580-198590. doi:10.1109/ACCESS.2020.3034100.
- [27] Y. Wang, Y. Shen, S. Mao, G. Cao, R. M. Nelms, Adaptive Learning Hybrid Model for Solar Intensity Forecasting, *IEEE Transactions on Industrial Informatics* 14(4) (2018) 1635-1645. doi:10.1109/TII.2017.2789289.
- [28] X. Luo, D. Zhang, An Adaptive Deep Learning Framework for Day-ahead Forecasting of Photovoltaic Power Generation, in: *Sustainable Energy Technologies and Assessments* 52(D) (2022) 1-26. doi:10.1016/j.seta.2022.102326.
- [29] E. Sarmas, S. Stropoulos, V. Marinakis, F. Santori, M. A. Bucarelli, H. Doukas, An Incremental Learning Framework for Photovoltaic Production and Load Forecasting in Energy Microgrids, *Electronics* 11(23) (2022) 1-17. doi:10.3390/electronics11233962.
- [30] R. J. Wai, P. X. Lai, Design of Intelligent Solar PV Power Generation Forecasting Mechanism Combined with Weather Information under Lack of Real-Time Power Generation Data, *Energies*, 15(10) (2022) 1-30. doi:10.3390/en15103838.
- [31] S. M. Miraftabzadeh, C. G. Colombo, M. Longo, F. Foiadelli, A Day-Ahead Photovoltaic Power Prediction via Transfer Learning and Deep Neural Networks, *Forecasting* 5(1) (2023) 213-228. doi:10.3390/forecast5010012.

Implementation of swarm intelligence methods for preprocessing in neuroevolution synthesis

Serhii Leoshchenko^a, Andrii Oliinyk^a, Sergey Subbotin^a and Tetiana Kolpakova^a

^a National University "Zaporizhzhia Polytechnic", Zhukovskogo street 64, Zaporizhzhia, 69063, Ukraine

Abstract

Using swarm intelligence techniques for data preprocessing over neuroevolution synthesis of artificial neural networks (ANNs) can provide a number of advantages. Therefore, swarm analysis techniques such as particle swarm optimization (PSO) or ant colony optimization (ACO) can effectively identify the most relevant traits from multidimensional data. This reduces the dimension of the input space, mitigating the Curse of dimension and improving the effectiveness of ANNs training. In addition, swarm analysis methods can filter out noisy or mismatched data points and detect outliers, increasing ANNs resistance to noisy input data and expanding generalization capabilities.

In general, the introduction of swarm intelligence techniques for data preprocessing prior to neuroevolution ANNs synthesis results in improved model performance, reduced computational complexity, and improved generalization capabilities, making it an appropriate approach in machine learning tasks.

This paper is proposed to consider the implementation of swarm intelligence methods for preprocessing to improve the performance of neuromodel synthesis.

Keywords 1

Preprocessing, swarm intelligence, neurosynthesis, neuromodels, machine learning, artificial neural networks

1. Introduction

Today, ANN-based neuromodels are widely used to automate and solve many processes and tasks of human activity (industry, medicine, operational processes, etc.). ANNs are computational models based on the structure and functioning of biological neurons in the human brain. They consist of interconnected nodes (neurons) organized in layers, with each neuron performing simple calculations and transmitting signals to connected neurons [1].

Let's look at how ANNs are used in diagnostic tasks.

1. Medical image analysis: Anns are used to analyze medical images such as X-rays, magnetic resonance imaging, computed tomography, and histopathological images. Convolutional neural networks (CNNs), a special type of deep ANNs designed for processing spatial data, are particularly effective in tasks such as tumor detection, organ segmentation, and pathology classification. CNNs can automatically study image features, providing accurate diagnostics and anomaly detection [1].
2. Disease diagnosis: Anns are used to diagnose various diseases by analyzing patient data such as symptoms, medical history, laboratory tests, and genetic information. Recurrent neural networks (RNNs) and long-term short-term memory networks (LSTMs), thanks to their structural features (the presence of feedback and gates, which allows to store the context/history of states), are able to

The Seventh International Workshop on Computer Modeling and Intelligent Systems (CMIS-2024), May 3, 2024, Zaporizhzhia, Ukraine
EMAIL: sergio.zntu@gmail.com (S. Leoshchenko); olejnikaa@gmail.com (A. Oliinyk); subbotin@zntu.edu.ua (S. Subbotin);
t.o.kolpakova@gmail.com (T. Kolpakova)
ORCID: 0000-0001-5099-5518 (S. Leoshchenko); 0000-0002-6740-6078 (A. Oliinyk); 0000-0001-5814-8268 (S. Subbotin); 0000-0001-8307-8134 (T. Kolpakova)



© 2023 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

process sequential data, used for tasks such as predicting disease progression, identifying risk factors, and diagnosing conditions such as heart disease, diabetes, and cancer [1].

3. Medical signal processing: Anns are used to process physiological signals such as an electrocardiogram (ECG), electroencephalogram (EEG), and electromyogram (EMG). They are used for tasks such as detecting arrhythmias, predicting seizures, classifying sleep stages, and analyzing muscle activity. Given the nature of the input data that is the result of clinical trials, models that best represent time series are used, in particular the RNNs and LSTM networks, which perfectly capture time dependencies and patterns in sequential data, making them suitable for analyzing medical signals [2].

4. Drug discovery and development: Anns are used in drug discovery and development processes, including virtual screening, molecular modeling, and drug toxicity prediction. They are used to analyze chemical structures, predict drug-target interactions, and develop new compounds with desired pharmacological properties. ANN-based models help speed up the drug search process and reduce the time and cost associated with developing new drugs [2].

5. Predictive and predictive modeling: ANNs are used to build predictive and predictive models in healthcare that help clinicians make informed decisions about patient management and treatment strategies. These models predict outcomes such as disease progression, treatment response, and patient survival based on clinical data, biomarkers, and other relevant factors [3].

Overall, ANNs are versatile diagnostic tools that offer the ability to analyze different types of data, extract meaningful patterns, and make accurate predictions, thereby improving medical decision-making and patient care.

Preprocessing refers to a set of methods applied to raw data before it is passed to a machine learning model, such as an ANN. These methods are aimed at converting data to a format that is more appropriate to the chosen training method, and at improving the overall performance and effectiveness of the model. Preprocessing includes steps such as data cleaning, normalization, object scaling, dimensionality reduction, and processing missing values or outliers [4].

Pretreatment is crucial for Ann synthesis and training for several reasons.

1. Normalization and scaling: Anns often work better when input objects are normalized or scaled to a similar range. Preprocessing techniques, such as minimum-maximum scaling or Z-score normalization, ensure that features make an equal contribution to the learning process, preventing certain features from dominating the learning process due to differences in scale [5].

2. Preprocessing methods allow you to process missing values or outliers in a data set. Imputation methods can be used to fill in missing values, while outlier detection and removal methods help ensure that extreme values do not have an excessive impact on model behavior [5].

3. Object design and selection: preprocessing allows for object design and selection, which removes irrelevant or redundant objects and creates new objects to better represent basic patterns in the data. This helps improve Model generalization and reduce overtraining [5].

4. Dimensionality reduction: large-size data can create problems for ANNs in terms of computational complexity and retraining. Dimensionality reduction techniques, such as Principal Component Analysis (PCA) or feature selection algorithms, help reduce the number of input measurements while preserving as much information as possible, thereby improving the efficiency and speed of the model [5].

5. Data balancing: classification problems with unbalanced class distributions can use preprocessing techniques such as resampling or understated sampling to balance the data set, ensuring that the model learns from representative samples of each class [5].

In general, preprocessing plays a crucial role in preparing data for Ann synthesis and training, ensuring that the model assimilates data efficiently, generalizes well to invisible instances, and provides reliable and accurate predictions.

Swarm intelligence is an area of research inspired by the collective behavior of decentralized, self-organizing systems in nature, such as ant colonies, flocks of birds, and flocks of fish. Swarm intelligence techniques aim to solve complex problems by coordinating the actions of multiple agents, each following simple rules, without the need for centralized control or global knowledge.

Some key characteristics of swarm intelligence include [6].

1. Self-organization: swarm-based intelligence systems exhibit emerging behavior where global patterns and solutions arise from local interactions between individual agents without explicit coordination.
2. Swarm intelligence relies on distributed decision-making, where each agent makes decisions based on local information and interaction with nearby neighbors, rather than relying on centralized control or external leadership.
3. Adaptation and reliability: swarm intelligent systems are often adaptive and resistant to environmental changes or disturbances. They can self-adjust environments and reorganize in response to dynamic conditions, providing stability and flexibility.
4. Research and Operation: Swarm Intelligence techniques combine exploration of the search space to identify new solutions and use known solutions to effectively optimize performance.

Swarm intelligence techniques can be applied to neurosynthesis, which involves computer-aided design or synthesis of artificial neural networks (ANNs) using optimization techniques. These methods can help you explore the wide range of possible neural network architectures, optimize network parameters, and improve network performance. Some uses of swarm intelligence techniques for neurosynthesis include [7]:

1. Self-organization: swarm-based intelligence systems exhibit emerging behavior where global patterns and solutions arise from local interactions between individual agents without explicit coordination.
2. Swarm intelligence relies on distributed decision-making, where each agent makes decisions based on local information and interaction with nearby neighbors, rather than relying on centralized control or external leadership.
3. Adaptation and reliability: swarm intelligent systems are often adaptive and resistant to environmental changes or disturbances. They can self-adjust environments and reorganize in response to dynamic conditions, providing stability and flexibility.
4. Research and Operation: Swarm Intelligence techniques combine exploration of the search space to identify new solutions and use known solutions to effectively optimize performance.
5. Swarm intelligence techniques can be applied to neurosynthesis, which involves computer-aided design or synthesis of artificial neural networks (ANNs) using optimization techniques.

These methods can help you explore the wide range of possible neural network architectures, optimize network parameters, and improve network performance. Some uses of swarm intelligence techniques for neurosynthesis include [8]:

6. Swarm intelligence techniques such as particle swarm optimization (PSO), ant colony optimization (ACO), or genetic algorithms (GAs) can be used to find optimal neural network architectures by exploring the space of possible configurations, including the number of layers, neuron types, connectivity patterns, and more. hyperparameters.
7. Swarm intelligence techniques can optimize neural network hyperparameters, such as learning speed, activation functions, regularization parameters, and network topology, to improve performance and generalization.
8. Swarm intelligence algorithms can help optimize the learning process of neural networks by tweaking parameters related to optimization algorithms (such as gradient descent options), convergence criteria, and data preprocessing techniques, resulting in faster convergence and improved performance.
9. Ensemble training: swarm intelligence can be used to optimize the creation of ensemble models that train and combine multiple neural networks to improve prediction accuracy and reliability.

Overall, swarm intelligence techniques offer powerful neurosynthesis tools to automate the design and optimization of neural networks for a variety of applications, including pattern recognition, classification, regression, and management tasks [8].

2. Related Works

Let's analyze why preprocessing of input data is necessary for neuroevolution synthesis of analytical models based on ANN and what such processes are reduced to [9].

Normalization and scaling. Preprocessing techniques such as normalization and scaling ensure that the input characteristics are at the same scale. This is important because neural networks can be sensitive to the magnitude of input values. Data normalization prevents certain functions from dominating the training process solely because of their scale, resulting in more stable and effective learning [9].

Filling in missing values and monitoring emissions. Preprocessing allows you to process missing values and outliers in input data. Missing values can be calculated using methods such as calculating the Mean, median calculation, or interpolation. Outliers, if left untreated, can negatively affect the learning process, distorting the studied model parameters. Preprocessing techniques such as outlier detection and removal ensure that the neural network learns from clean and reliable data, resulting in improved performance [9].

Data reduction and feature selection. Preprocessing allows you to develop and select objects, while removing irrelevant or redundant objects and creating new informative objects. This process helps reduce the size of the input space by focusing the neural network's attention on the most relevant functions and improving its ability to generalize invisible data [10].

Data balancing. In classification problems with unbalanced class distribution, preprocessing methods can balance a data set by over-sampling minority classes, under-sampling majority classes, or using more advanced methods such as the Synthetic Minority over-sampling Technique (SMOTE). Data set balancing ensures that the neural network is trained on a representative set of samples from each class, preventing it from shifting to the majority class [10].

Noise reduction and improved data quality: preprocessing techniques can help filter out noise and improve the overall quality of input data. This is especially important in real data sets that may contain inappropriate or erroneous information. Noise removal ensures that the neural network focuses on basic data patterns, resulting in more accurate and reliable predictions [10].

Overall, input preprocessing plays a crucial role in neuroevolution synthesis, ensuring that the neural network learns from high-quality, well-structured data. This contributes to more efficient learning, faster convergence, and better generalization performance, which ultimately leads to more reliable and accurate neural network models [11].

Also, when studying the very practical implementation of input preprocessing, it should be noted that such processes can help save time during neurosynthesis as follows.

Obviously, preprocessing techniques such as selecting or extracting objects can reduce the dimension of input data by eliminating non-essential or redundant objects. With fewer input parameters, the neuroevolution process requires less computational resources and time to study the reduced feature space, which leads to faster processing [11].

Normalizing or scaling the input data ensures that all functions are on the same scale, preventing certain functions from dominating the learning process based solely on their size. Normalized data contributes to more efficient learning convergence by reducing the time required for the neuroevolution synthesis process [11].

Preprocessing methods for handling missing values and outliers ensure that the input data is clean and reliable. By removing or attributing missing values and outliers, the neuroevolution process can focus on learning based on high-quality data, resulting in faster convergence and more efficient use of computational resources.

In general, pretreatment of input data before neuroevolution synthesis helps optimize the learning process, increases the efficiency of model training, and reduces the overall processing time required to achieve the desired results [12].

Let's take a look at some of the most popular methods of swarm intelligence.

PSO method:

- PSO is inspired by the social behavior of flocks of birds or schools of fish;
- each candidate solution (particle) in the search space adjusts its position based on its own experience and the most known position of its neighbors;
- PSO is relatively easy to implement and can efficiently search in large-dimensional spaces;
- the method tends to approach local optima quickly, but may have difficulty going beyond local optima in multi-modal or deceptive landscapes.

ACO method [13], [14]:

- ACO is inspired by the search behavior of ants.
- artificial ants leave pheromone traces along the edges of the graph representing the problem space, and the intensity of pheromone traces reflects the desirability of the path;
- ACO is excellent in combinatorial optimization problems and is particularly effective in solving problems with discrete and undifferentiated objective functions;
- the method requires careful parameter adjustment and may suffer from slow convergence, especially in large and complex problem spaces.

Firefly algorithm (FA) [13], [14]:

- FA is inspired by the flashing behavior of fireflies, when brighter fireflies attract others;
- the method optimizes the set of solutions by iteratively moving brighter solutions to brighter ones in the search space;
- FA is easy to implement and requires no gradient information, making it suitable for optimization tasks with complex and multi-modal landscapes.;
- however, in some cases, FA can suffer from slow convergence and premature convergence, especially in multidimensional and nonlinear optimization problems.

Each of these swarm analysis methods has its own strengths and weaknesses, which makes them suitable for different types of optimization problems and subject areas. The choice of method depends on factors such as problem complexity, search space characteristics, computational resources, and desired convergence properties [13], [14].

Let's compare the most well-known methods of swarm intelligence according to the following criteria:

- basic idea: this criterion describes the natural phenomenon or behavior that inspired the development of each method. Understanding a biological or natural process can provide insight into how the method works and its suitability for various optimization tasks [15], [16];
- objective function: refers to the type of optimization problems that each method is primarily designed to solve. Some methods are better suited for continuous optimization problems, while others are excellent for combinatorial or discrete optimization problems [15], [16];
- aggregate initialization: this criterion describes how the initial set of candidate Solutions is generated in each method. Random initialization is common, but some methods may have specific initialization strategies [15], [16];
- environment vs. exploitation research: reflects the balance between research (finding new areas of the solution space) and exploitation (refining known promising solutions). Different methods may show different trends in research or exploitation [17];
- communication mechanism: this criterion refers to how information is exchanged between individuals in a swarm. This may include mechanisms such as pheromone traces in ACO or brightness-based attraction in FA [17];
- convergence properties: describes the speed and efficiency with which a method comes to a solution. So, for example, some methods can converge quickly, but run the risk of getting stuck in local optimums, while others can converge more slowly, but with better global optimization properties [17];
- this criterion evaluates how sensitive the method is to the choice of metaparameters. Methods that are less sensitive can be easier to configure and more reliable in various problem areas;
- implementation complexity: it reflects the ease of implementation and computational complexity of each method. Methods with simpler implementations may require less computing resources;
- areas of application: this criterion defines the types of optimization problems that each method is usually applied to. Understanding typical application areas can help you choose the most appropriate method for a particular task [18]-[21].

In the table. 1 shows the results of comparison by criteria.

Table 1

Results of comparing PSO, ACO and FA

Criteria	PSO	ACO	FA
Nature Inspiration	Social behavior of bird flocks or fish schools	Foraging behavior of ants	Flashing behavior of fireflies
Objective Function	Continuous optimization problems	Combinatorial optimization problems	Continuous optimization problems
Population Initialization	Random initialization	Random initialization	Random initialization
Exploration vs. Exploitation	Balanced exploration and exploitation	Exploration focused	Exploration focused
Communication Mechanism	No direct communication	Pheromone-based communication	Attraction based on brightness
Convergence Properties	May converge quickly to local optima	Slow convergence	May suffer from slow convergence
Robustness to Parameter Settings	Moderately sensitive	Sensitive	Moderately sensitive
Implementation Complexity	Relatively simple to implement	Moderate complexity	Relatively simple to implement
Application Domains	Continuous optimization problems	Combinatorial optimization problems	Continuous optimization problems

From the results of the comparison, we can conclude that the FA – method is the most universal in order to use it for preprocessing input data in the future.

2.1. Main features of applying the FA method

The FA method is a nature-inspired optimization method developed by Xin-she Yang in 2008. It is inspired by the blinking behavior of fireflies, where fireflies use bioluminescence to attract partners or prey. FA is a metaheuristic algorithm used to solve optimization problems in various fields [1], [6], [20].

The Firefly method was proposed by Xin-she Yang in 2008.

The method was developed as a population-based optimization method inspired by the blinking behavior of fireflies to solve optimization problems more efficiently.

Since its introduction, the FA method has gained popularity and has been applied to various optimization problems in Engineering, Computer Science, and other fields.

The main idea of the Firefly algorithm is to simulate the blinking behavior of fireflies to find optimal solutions to the optimization problem. The method works on the basis of the following principles [1], [6], [20]:

attraction: fireflies are attracted to each other depending on the brightness of their flashes. Similarly, in FA, the attractiveness of a firefly (solution) is determined by its suitability, with brighter solutions representing better solutions;

moving towards brighter solutions: Fireflies tend to move towards brighter fireflies nearby. In FA, each Firefly (solution) adjusts its position in the search space, moving towards brighter solutions, and the intensity of attraction is determined by the difference in brightness and distance between the Fireflies [1], [6], [20];

randomization and research: to help explore the search space, FA includes randomization by adding a random component to Firefly movement. This ensures that the method does not get hung up on local optimums and can explore different areas of the solution space.;

global optimization: FA aims to find a global optimal solution by iteratively updating Firefly positions based on their attractiveness and distance between them. The method converges when the Firefly positions no longer change significantly or the predefined completion criterion is met.

Overall, the FA method is a simple but effective optimization technique inspired by Firefly behavior. It is able to effectively solve a wide range of optimization problems and is successfully used in various fields due to its simplicity and efficiency [1], [6], [20].

Now we can dive into the intricacies of firefly optimization in more detail. The essence of the method is clearly shown in Fig. 1.

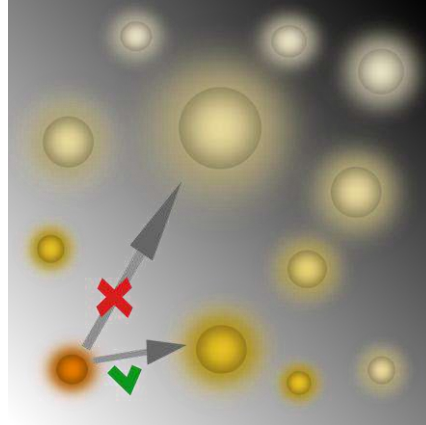


Figure 1: Fireflies in the search space. Visibility decreases with increasing distance

3. Proposed method

Levy's flight function can be incorporated into the Firefly Algorithm (FA) to enhance its exploration capability and improve its convergence speed. Levy's flight is a random walk process described by Levy distribution, which exhibits heavy-tailed behavior and allows for long-distance jumps in the search space. By incorporating Levy's flight function, the fireflies in the FA can explore the search space more effectively, leading to better global optimization performance [1], [6], [20].

Here's how Levy's flight function can be integrated into the Firefly Algorithm [22]-[25]:

1. Step 1: Levy Flight Generation:

- at each iteration of the algorithm, Levy flights are generated for each firefly to determine their movement direction and distance;
- Levy flights are generated using Levy's flight function, which produces step sizes according to the Levy distribution. The step sizes are typically drawn from a Levy distribution with a specified scale parameter.

2. Movement of Fireflies:

- the fireflies adjust their positions based on the generated Levy flights. Each firefly moves a distance determined by Levy's flight function in a random direction;
- the movement of fireflies is guided by their attractiveness, with brighter fireflies attracting others more strongly. Fireflies move towards brighter individuals while incorporating Levy flights for exploration.

3. Exploration and Exploitation:

- Levy flights facilitate exploration by allowing fireflies to make long-distance jumps in the search space, enabling the algorithm to escape local optima and explore new regions;
- at the same time, the attractiveness mechanism of the Firefly Algorithm ensures that fireflies tend to converge towards brighter solutions, promoting exploitation of promising regions of the search space.

4. Update Positions and Iteration:

- after adjusting their positions based on Levy flights and attractiveness, the positions of fireflies are updated, and the algorithm proceeds to the next iteration;
- the process continues until a termination criterion is met, such as a maximum number of iterations or convergence of solutions.

By incorporating Levy's flight function into the Firefly Algorithm, the algorithm gains enhanced exploration capabilities, enabling it to efficiently search for global optima in complex optimization problems. This modification can lead to improved convergence speed and better overall performance of the algorithm across various domains [22]-[25].

4. Experimental research

For the experimental research and comparison of proposed method with another approaches was be used the following as the training and testing data:

- RT-IoT2022 [26], [16]. The RT-IoT2022 is a data sample formed by scientists on the basis of the Internet of things infrastructure, when a wide range of Internet of Things devices and a range of network attacks are combined in real-time conditions. The sample covers normal and hostile network behaviors that model a general view of real-world scenarios;
- Visegrad Group companies data (VGCD) [27], [16]. This dataset contains information about companies from the Visegrad Group countries, which include the Czech Republic, Hungary, Poland, and Slovakia. The Visegrad Group is a cultural and political alliance of these four Central European countries;
- Influenza Outbreak Event Prediction via Twitter (IOEP) [28]. This dataset aims to predict influenza outbreaks using Twitter data

The general information about datasets presented in Table 2.

Table 2

General information about datasets

Datasets	Feature Type	Number of Features	Number of Instances
RT-IoT2022	Real	83	123117
VGCD	Real	83	85
IOEP	Real	523	75839

The meta-parameters for neuroevolution synthesis of models demonstrate at Table 4.

Table 3

The meta-parameters for neuroevolution synthesis

Metaparameter	Value
Population size	100
Elite size	5%
Activation function (fitness functions)	hyperbolic tangent
Mutation probability	25%
Crossover type	uniform
Types of mutation	deleting an interneuronal connection
	removing a neuron
	adding interneuronal connection
	adding a neuron
Clustering method	changing the activation function
Neighbors number	k -nearest neighbors
	7

The results of the work present at Table 4.

Table 4

The work results of proposed method

Datasets	ANN time synthesis (without feature selection)	ANN time synthesis (using PSO)	ANN time synthesis (using ACO)	ANN time synthesis (using FA)	Error on test sample
RT-IoT2022	8456 s	7225 s	6935 s	5689 s	0.91
VGCD	9555 s	8123 s	7456 s	6126 s	0.924
IOEP	9832 s	8526 s	7626 s	6751 s	0.942

5. Discussions of results

Analyzing the results obtained, it is worth noting that using any method for data preprocessing significantly reduces the time of subsequent synthesis. However, it is the use of FA that demonstrates the greatest optimization of synthesis time. This can be explained precisely by better selection of informative features. After all, if features can be shortened differently during preprocessing by different methods, the FA method helps to better track hidden relationships between data, and therefore significantly simplify the process of forming structural relationships between really dependent features.

6. Conclusion

At the paper examines the using swarm intelligence techniques for data preprocessing over neuroevolution synthesis of ANN can provide a number of advantages. Therefore, swarm analysis techniques such as particle swarm optimization or ant colony optimization can effectively identify the most relevant traits from multidimensional data. This reduces the dimension of the input space, mitigating the Curse of dimension and improving the effectiveness of ANNs training. In addition, swarm analysis methods can filter out noisy or mismatched data points and detect outliers, increasing ANNs resistance to noisy input data and expanding generalization capabilities.

The results of experiments prove that the new concept has quite acceptable (certifying) indicators of time use during the synthesis of neural networks.

7. Acknowledgements

The work was carried out with the support of the state budget research projects of the state budget of the National University "Zaporozhzhia Polytechnic" "Intelligent information processing methods and tools for decision-making in the military and civilian industries" (state registration number 0124U000250).

8. References

- [1] X.-S. Yang, M. Karamanoglu, Z. Cui, A. H. Gandomi, R. Xiao, *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications*, Elsevier, 2013.
- [2] T. M. Shami, A. A. El-Saleh, M. Alswaiti, Q. Al-Tashi, M. A. Summakieh, S. Mirjalili, *Particle Swarm Optimization: A Comprehensive Survey*, IEEE Access 10 (2022) 10031–10061. doi:10.1109/access.2022.3142859.
- [3] F. Marini, B. Walczak, *Particle Swarm Optimization*, Comprehensive Chemometrics, Elsevier, 2020, p. 649–666. doi:10.1016/b978-0-12-409547-2.14581-0.

- [4] P. A. Gkaidatzis, A. S. Bouhouras, D. P. Labridis, Application of PSO in Distribution Power Systems: Operation and Planning Optimization, Applying Particle Swarm Optimization, Springer International Publishing, Cham, 2021, p. 321–351. doi:10.1007/978-3-030-70281-6_17.
- [5] A. Vasuki, Ant Colony Optimization, Nature-Inspired Optimization Algorithms, Chapman and Hall/CRC, 2020, p. 99–114. doi:10.1201/9780429289071-8.
- [6] X.-S. Yang, X.-S. He, Why the Firefly Algorithm Works?, Nature-Inspired Algorithms and Applied Optimization, Springer International Publishing, Cham, 2017, p. 245–259. doi:10.1007/978-3-319-67669-2_11.
- [7] Ant Colony Optimization — Intuition, Code & Visualization, 2024. URL: <https://towardsdatascience.com/ant-colony-optimization-intuition-code-visualization-9412c369be81>
- [8] Growth Optimizer (GO), 2024. URL: <https://medium.com/@melika.akz97/growth-optimizer-go-dd6da47a4e93>
- [9] Wolf Search Algorithm (WSA): Harnessing the Social and Hunting Strategies of Wolves for Optimization, 2024. URL: <https://python.plainenglish.io/wolf-search-algorithm-wsa-harnessing-the-social-and-hunting-strategies-of-wolves-for-9de4d1fc19c6>
- [10] Machine Learning with Particle Swarm Optimization by Sebastian Raschka, 2021. URL: <https://sebastianraschka.com/>
- [11] M. F. Ahmad, N. A. M. Isa, W. H. Lim, K. M. Ang, Differential evolution: A recent review based on state-of-the-art works, Alex. Eng. J. (2021). doi:10.1016/j.aej.2021.09.013.
- [12] A. Sharma, A. Sharma, J. K. Pandey, M. Ram, Swarm Intelligence Applications in Artificial Neural Networks, Swarm Intelligence, CRC Press, Boca Raton, 2021, p. 99–122. doi:10.1201/9781003090038-5.
- [13] S. J. Anand, C. Tamilselvi, D. Sam, C. Kamatchi, N. Dey, K. Sujatha, Swarm Intelligence-based Framework for Image Segmentation of Knee MRI Images for Detection of Bone Cancer, Swarm Intelligence and Machine Learning, CRC Press, Boca Raton, 2022, p. 95–115. doi:10.1201/9781003240037-6.
- [14] M. Ehteram, A. Seifi, F. B. Banadkooki, Structure of Particle Swarm Optimization (PSO), Application of Machine Learning Models in Agricultural and Meteorological Sciences, Springer Nature Singapore, Singapore, 2023, p. 23–32. doi:10.1007/978-981-19-9733-4_2.
- [15] Ant Colony Optimization, Handbook of Machine Learning, WORLD SCIENTIFIC, 2019, p. 123–141. doi:10.1142/9789811205675_0007.
- [16] A. Sharma, A. Sharma, J. K. Pandey, M. Ram, Swarm Intelligence Applications in Artificial Neural Networks, Swarm Intelligence, CRC Press, Boca Raton, 2021, p. 99–122. doi:10.1201/9781003090038-5.
- [17] O. Nagaya, A. W. Pillay, E. Jembere, Comparative Study of Image Resolution Techniques in the Detection of Cancer Using Neural Networks, Artificial Intelligence Research, Springer Nature Switzerland, Cham, 2023, p. 187–202. doi:10.1007/978-3-031-49002-6_13.
- [18] D. Kumar, B. G. R. Gandhi, R. K. Bhattacharjya, Firefly Algorithm and Its Applications in Engineering Optimization, Nature-Inspired Methods for Metaheuristics Optimization, Springer International Publishing, Cham, 2020, p. 93–103. doi:10.1007/978-3-030-26458-1_6.
- [19] D. Varela, J. Santos, Crowding Differential Evolution for Protein Structure Prediction, From Bioinspired Systems and Biomedical Applications to Machine Learning, Springer International Publishing, Cham, 2019, p. 193–203. doi:10.1007/978-3-030-19651-6_19.
- [20] H. Xu, S. Yu, J. Chen, X. Zuo, An Improved Firefly Algorithm for Feature Selection in Classification, Wirel. Pers. Commun. 102.4 (2018) 2823–2834. doi:10.1007/s11277-018-5309-1.
- [21] W. Li, Z.-d. Lu, D.-L. Deng, Quantum Neural Network Classifiers: A Tutorial, SciPost Phys. Lect. Notes (2022). doi:10.21468/scipostphyslectnotes.61.
- [22] L. Fernández-Brillet, O. Álvarez, J. L. Fernández-Martínez, The PSO Family: Application to the Portfolio Optimization Problem, Applying Particle Swarm Optimization, Springer International Publishing, Cham, 2021, p. 111–132. doi:10.1007/978-3-030-70281-6_7.
- [23] H. Gao, B. Feng, Y. Hou, L. Zhu, Training RBF Neural Network with Hybrid Particle Swarm Optimization, Advances in Neural Networks - ISNN 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, p. 577–583. doi:10.1007/11759966_86.

- [24] Y. Geng, L. Zhang, Y. Sun, Y. Zhang, N. Yang, J. Wu, Research on Ant Colony Algorithm Optimization Neural Network Weights Blind Equalization Algorithm, *Int. J. Secur. Its Appl.* 10.2 (2016) 95–104. doi:10.14257/ijisa.2016.10.2.09.
- [25] RT-IoT2022, 2022. URL: <https://archive.ics.uci.edu/dataset/942/rt-iot2022>
- [26] E Visegrad Group companies data, 2021. URL: <https://github.com/7ossam81/EvoloPy>
- [27] Influenza Outbreak Event Prediction via Twitter, 2021. URL: <https://archive.ics.uci.edu/dataset/861/influenza+outbreak+event+prediction+via+twitter>

Algorithms and methods for comparing microstructures of materials based on their images

Valeriia Hritskova¹, Oleh Semenenko¹, Mariia Shapovalova¹ and Oleksii Vodka¹

¹ National Technical University «Kharkiv Polytechnic Institute», 2, Kyrpychova str., Kharkiv, 61002, Ukraine

Abstract

The study aims to develop robust image processing algorithms capable of automatically identifying and characterizing individual grains within material microstructures. These algorithms extract essential grain properties, including area, shape factor, and orientation angle. Additionally, the study explores which grain characteristics are most effective for microstructure comparison. The proposed algorithm segments microstructure images to isolate individual grains. Grain properties (e.g., area, perimeter, circularity) are quantified. The distributions of grain characteristics are analyzed using violin plots. Both visual comparisons and statistical measures (mean, variance, skewness) inform microstructure similarity. Proposed algorithms have been tested on validation images and errors have been estimated. Understanding microstructure properties is crucial for material design, quality control, and performance optimization. The proposed algorithms contribute to automated microstructure analysis, benefiting fields such as materials science, engineering, and manufacturing.

Keywords

Image Processing Algorithm, Feature Extraction, Grain Characteristics, Violin Plots, Statistical Comparison

1. Introduction

Algorithms and methods for comparing the microstructures of materials based on their images have become the object of considerable attention in modern research. The significance of this area lies in the possibility of obtaining information about the properties of materials that affect their functionality and application. The development of algorithms for comparing microstructures has become an important task in the context of finding optimal solutions in materials science. The initiation of this process involved establishing a complex for generating microstructures; however, a challenge emerged concerning the comparison between these generated structures and real ones. In light of this, the solution to this problem involved the development of image processing methods for comparing the generated microstructures with experimental data.

The microstructures of materials play a fundamental role in their properties and functionality, determining their mechanical, thermal, and electrical characteristics. The contribution of microstructures to the development of science and technology cannot be overstated, as their understanding and optimization have a significant impact on the design and production of materials. Research emphasizes not only the significant role of the microstructure itself but also its components. Particular attention is given to the process of microstructure formation - crystal growth. Since physical experiments are difficult to study the characteristics of microstructure evolution during grain growth, computer modeling is used as an effective alternative.

Using the cellular automata method, various models have been considered so far: 2D modeling of microstructures [4], [5], 3D modeling of microstructures [6]-[8], [19], [20] modeling using probabilities [9], [10], modeling of crystal growth by recrystallization [11]-[12]. The study of these models has allowed us to analyze various aspects of the processes of microstructure formation, their evolution, and their influence on the properties of materials. The use of different

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ valeriia.hriskova@infiz.khpi.edu.ua (V. Hritskova); oleh.semenenko@infiz.khpi.edu.ua (O. Semenenko); mariia.shapovalova@khpi.edu.ua (M. Shapovalova); oleksii.vodka@khpi.edu.ua (O. Vodka)

ORCID: 0009-0002-6825-726X (V. Hritskova); 0009-0003-7275-2412 (O. Semenenko); 0000-0002-4771-7485 (M. Shapovalova); 0000-0002-4462-9869 (O. Vodka)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

modeling approaches allows for obtaining a more complete understanding of the physical and chemical processes that occur during crystal growth and microstructure formation.

A special application MatViz3D (<https://github.com/MME-NTU-KhPI/MatViz3D>) has been developed for computer modelling of the crystallization process and generation of microstructures. The main advantages of the application are the three-dimensional generation of microstructures, which allows to obtain detailed and realistic images of the material structure, the selection of cell neighborhoods for the diversity of generation, and the ability to follow the step-by-step crystallization process. Overall, MatViz3D is a powerful tool for modeling, analyzing, and visualizing material microstructures using a variety of approaches and techniques.

Once the microstructure is generated, it is necessary to conduct research and identify the characteristics inherent in the model. The identification of model characteristics is carried out through data processing and systematization. Among the tools that can be used for this purpose, a special place is occupied by processing data in the form of images. Using techniques such as image analysis and image processing, it is possible to identify the main features of microstructures, such as grain size, shape, and distribution of grains in space. For example, the use of segmentation algorithms can automatically identify individual grains in microstructure images, which simplifies further analysis. In addition, the use of image processing techniques to determine various characteristics, such as grain size, shape, and texture, provides quantitative data that can be used for further analysis and comparison with experimental data or other models. This approach to image processing is becoming increasingly common in the study of material microstructures and plays an important role in the development of the fields of materials science and mechanics [14]-[18].

Having processed the image and obtained the grain properties of the microstructures, it becomes necessary to develop an algorithm for their comparison. In this context, a special emphasis is placed on comparing the distributions and statistical characteristics of each of the grain properties. This approach will allow for a deeper analysis and comparison of material microstructures, contributing to the further development of the fields of materials science and mechanics.

Although there are already software tools, such as CLEMEX [21] and Fiji [22], designed to process images and detect regions on them, this study developed a system for comparing grain characteristics that goes beyond the capabilities of these programs. The developed system allows for automated comparison of the distribution of microstructure characteristics and determines statistical parameters that are not available in existing software. This expands the analytical capabilities and provides a deeper understanding of the microstructures of materials.

2. Problem statement

The objective of this study is to develop image processing algorithms for automatically identifying regions in an image that correspond to individual grains in the microstructure of a material. The resulting algorithms should also provide the calculation of the characteristics of each grain, such as area, shape factor, orientation angle, etc.

Further, the developed algorithms will be tested on test images of microstructures, after which the results will be presented. Based on the analysis of these results, recommendations should be made as to which grain characteristics can be used for more effective comparison of microstructures and which may have limited variability and therefore be less important for comparison.

In addition, the study will develop an algorithm for comparing microstructures by analyzing the distributions of grain characteristics. For this purpose, violin plots will be used to visualize the distributions of grain size and other characteristics. In addition to comparing the visual characteristics, the statistical properties of the distributions will be analyzed, which will provide additional data on the similarity or difference of microstructures in different cases.

3. Methodology

The study and reproduction of microstructures plays a key role in the development of new materials with unique properties that will find applications in a wide range of industries,

including electronics, aviation, medicine, and energy. Microstructures are defined as the organization and arrangement of materials at the microscopic level, and they have a significant impact on the properties and behavior of materials. It is important to note that the formation of microstructures is a complex and multifactorial problem, which is influenced by a variety of factors, including chemical composition, temperature, pressure, cooling rate, and others. Furthermore, microstructures can exhibit extreme diversity depending on the type of material, the number of impurities, the manufacturing process, and the conditions.

The key stage is the formation of the internal structure of the material, as it allows analyzing the interaction between its elements, such as location, quantity, and nature. This helps to determine the optimal conditions for achieving the desired material parameters (e.g. strength, wear resistance, thermal conductivity, etc.). The detailed analysis and presentation of such material information can enable modeling and simulation with an accurate description of specific microstructural features. It also opens up the possibility of performing highly accurate engineering calculations using well-known methods, such as the finite element method.

The ability to characterize microstructural features using statistical methods is a significant advance in materials science. This helps to increase the accuracy of material property predictions.

Conventional techniques and methodologies employed for the quantitative analysis of three-dimensional structures using data derived from two-dimensional images or their cross-sections enable the prediction and determination of three-dimensional structural attributes. These encompass parameters such as volume, surface area, boundary length, and other descriptors derived from the analysis of images acquired from various orientations or viewpoints. Such approaches facilitate the evaluation of geometric and morphological characteristics of structures and materials.

However, there are microstructural parameters that cannot be determined from a single two-dimensional section, such as the connectivity of features, the true shape of inclusions, and the number of inclusions per unit volume. The need for a more complete characterization of microstructures has led to the development of techniques that allow for the direct acquisition of three-dimensional microstructural data of grain structures, such as serial sectioning, intergranular corrosion, and various X-ray tomography-based techniques. This includes references [2] and [3], where the authors quantify a set of microstructural parameters and their relationships to determine morphological characteristics using the serial sectioning technique to collect 3D crystallographic data. Many of the parameters have been quantified in two-dimensional space, while only a few have been determined in three-dimensional space, but the study of the relationships between these parameters has remained limited [1].

Thus, despite the significant progress in microstructure generation and the development of related applications, this topic requires further research and development. There is potential for improving generation algorithms, expanding the functionality of applications, and improving the accuracy of microstructure visualization. This study aims to analyze and compare the structures obtained experimentally with similar structures generated by artificial methods in an analytical context. The paper uses a set of metrics to compare such structures. Features such as sample and inclusion areas, their perimeters, and the ratio of these parameters are analyzed. The concept of area equivalence is introduced, and equivalent radii are found. Based on these indicators, the results are visualized using histograms and Kernel Density Estimation (KDE). By performing a statistical analysis of the results, using mathematical expectation and variance of values, a comparison is made, and a conclusion is drawn about the adequacy of the use of computer modeling of microstructure and artificial sample generation in comparison with experimental data.

3.1. Metrics for comparing experimental and generated structures

Several parameters are employed to ascertain the characteristics of grains, facilitating the evaluation of their shape and size.

- The normalized grain area is determined by the ratio of the grain area to the total image area using the formula (1):

$$A_n = \frac{A_{gr}}{A}, \quad (1)$$

where, A_{gr} - is the grain area, and A - is the total area of the image.

- The grain shape coefficient is calculated as the ratio of the grain area to the square of its perimeter according to (2):

$$C_s = 4\pi \frac{A_{gr}}{P_{gr}^2}, \quad (2)$$

where P_{gr} - is the perimeter of the grain.

- The equivalent circle radius of the grain, as the value of the radius of a circle having the same area as the area of the grain projection onto the plane, is calculated by (3):

$$ecr = \sqrt{\frac{A_n}{\pi}}, \quad (3)$$

- In addition, the orientation angle ψ which is reflected in the deviation of the major axis of the grain S_x from the horizontal axis.
- The scale factor of the grain is defined as the ratio of the large to the small grain axes according to (4):

$$S_c = \frac{S_x}{S_y}, \quad (4)$$

where S_x - is the major axis, and S_y - is the minor axis of the grain.

An example of the appearance of an arbitrary grain with the designation of the minor and major axes and the orientation angle is shown in Figure 1.

- The grain inertia tensor is represented as (5):

$$I = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{pmatrix}, \quad (5)$$

where I_{xx} and I_{yy} - are the principal moments of inertia, and I_{xy} (or I_{yx} since the inertia tensor is symmetric) - is the element representing the moment of inertia between the x and y axes.

- The grain aspect ratio, which indicates how closely a shape coincides with the rectangle described around it, is calculated using the formula (6):

$$AR = \frac{A_b}{A_{gr}}, \quad (6)$$

where A_b - is the area of the described grain rectangle. The closer the value of the coefficient is to 1 means the more rectangular the shape. In the ideal case, when the shape is a pure rectangle, this coefficient will be equal to 1.

- The grain compactness ratio is calculated using formula (7):

$$C_k = \frac{A_c}{A_{gr}}, \quad (7)$$

where A_c - is the area of the convex polygon that is described around the grain. This coefficient provides information on how close the grain is to a round or uniform shape: the larger the coefficient, the more compact the grain shape.

- The ratio of the area to the grain axes, which indicates how elliptical or circular the grain shape is, is calculated using formula (8):

$$E = \frac{A_{gr}}{S_x \cdot S_y}. \quad (8)$$

- The ratio of the inertia tensor to the grain area, which indicates the mass distribution relative to the geometric properties of the grain, is calculated using formula (9):

$$I_{area} = \frac{I}{A_{gr}}. \quad (9)$$

During the research, an issue arose concerning the compatibility of existing image processing algorithms, which were primarily tailored for analyzing two-dimensional images, with the three-dimensional microstructures generated. To address this challenge, the approach involved partitioning the three-dimensional microstructure into layers, each of a single voxel thickness, as a preliminary step before the analysis of grain properties. This partitioning effectively transforms

the problem into a series of two-dimensional image sets, facilitating subsequent processing procedures.

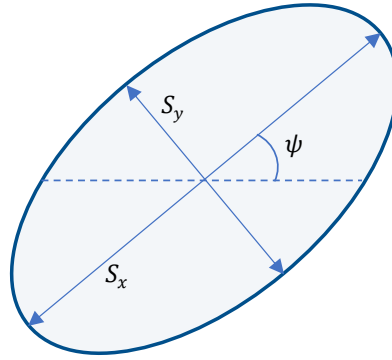


Figure 1: Arbitrary grain with labels for minor and major axes and orientation angle

The generated microstructure is depicted in Figure 2, while the experimental microstructure is shown in Figure 3. These images will be utilized in the study for characterization and comparison purposes.

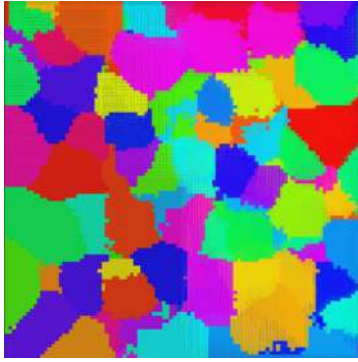


Figure 2: Generated microstructure

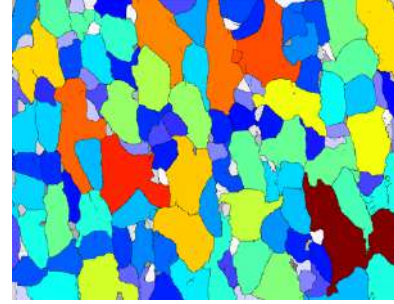


Figure 3: Experimental microstructure

3.2. Calculating and visualizing statistical characteristics

As part of the study, statistical characteristics were calculated for the results of the grain characteristics analysis. For each of the characteristics obtained by software, the mean (10), standard deviation (11), median (12), mode, range (13), and interquartile range (14) were calculated.

- The average value is calculated as the arithmetic mean of the characteristic values for each grain using the formula (10):

$$\mu = \frac{\sum_{i=1}^n x_i}{n}, \quad (10)$$

where x_i - is the value of the characteristic for each grain, n - is the number of grains, and i - belongs to the set of integers.

- The standard deviation is defined as the square root of the variance using formula (11):

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}}, \quad (11)$$

where x_i - is the value of the characteristic for each grain, i - belongs to the set of integers, μ - is the average value, and n - is the number of grains.

- The median for a data set of the form $(x_1, x_2, \dots, x_1)$ is defined as (12):

$$M = \begin{cases} \frac{x_{i+1} + x_i}{2}, & \text{if } i \text{ is odd} \\ \frac{x_i + x_{i+1}}{2}, & \text{if } i \text{ is even} \end{cases}, \quad (12)$$

where x_i - are the values of the characteristic for each grain, ordered in ascending order, and i - belongs to the set of integers.

- The mode (Mo) is the most frequent value in the data set.
- The range is the difference between the maximum and minimum values in the data set, determined by formula (13):

$$Range = x_{max} - x_{min}, \quad (13)$$

where x_{max} - is the maximum value in the dataset, and x_{min} - is the minimum value.

- The interquartile range is the difference between the third (75%) and first (25%) quartiles in the dataset, determined by the formula (14):

$$IQR = Q_3 - Q_1, \quad (14)$$

where Q_3 - is the third quartile, and Q_1 - is the first.

After these statistics were calculated, the data were visualized using violin plots. This method made it possible to compare the distribution of grain property values in the generated and real microstructures, providing an opportunity to conclude the interaction and characteristics of the grains.

To build a violin chart, the data is first processed, including filtering by the selected characteristic.

Then, the data is prepared for display, namely, it is combined into one DataFrame, where the category (generated or real grains), the name of the property, and its value are indicated. The next step is to build the graph itself using the Matplotlib and Seaborn libraries. In a scatterplot, each grain property has its distribution of values for the generated and real grains. Adding statistical information such as the mean, standard deviation, and median helps to better understand the distribution of the data.

4. Testing

4.1. Testing image processing methods

One of the objectives of the study is to test the methods of processing images obtained from the experiment and generated artificially. The testing process involves the experimental application of various in-age processing and analysis algorithms to accurately determine the characteristics of the microstructural elements of a material. It is primarily used to study the properties of material grains.

The test results indicate the effectiveness of the chosen approach and the possibility of its application in further research in this area. To achieve these goals, four images of different sizes were generated (Fig. 2-5), containing the same grain. Each image includes one grain, which is the same for all images. The first two samples consist of square grains of 50×50 pixels each, with sample sizes of 200×200 and 250×250 pixels, respectively (Fig. 2 and Fig. 3). The third and fourth samples contain round grains with a radius of 50 pixels each. The overall dimensions of the studied images are 200×200 (Fig. 4) and 300×300 pixels (Fig. 5).

For each image, the grain characteristics such as area, normalized area, shape factor, equivalent circle radius, orientation angle, scale factor, and inertia tensor were calculated according to (1)-(9). The calculated values of the characteristics were compared with the analytically calculated values to verify the correctness of the image processing. The results of the calculated characteristics are shown in Table 1.

Analyzing the results, it can be seen that the values of the characteristics do not depend on the size of the age, so they can be compared for further study and matching the generated structures to the experimental ones.

Additionally, the calculation of the relative error for the diverse grain characteristics acquired from the analysis is presented in Table 1. The relative error is determined through the comparison between the software-calculated value of the characteristic and the analytically measured value. Elevated relative error values signify substantial deviations between measurements, whereas lower values suggest relatively precise measurements.

In general, the analysis results show very good agreement between the values obtained analytically and by software for most grain characteristics.



Figure 4: A 200×200 pixel sample with 50×50 pixel square grains

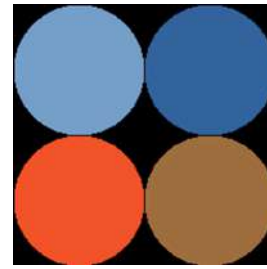


Figure 6: A 200×200 pixel sample with round grains of 50 pixels radius

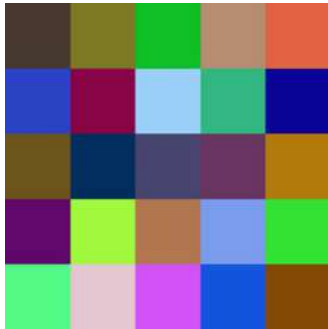


Figure 5: A 250×250 pixel sample with 50×50 pixel square grains

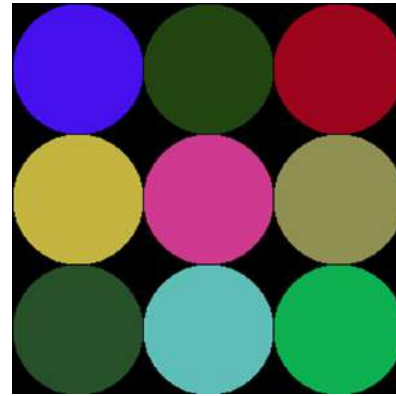


Figure 7: A 300×300 pixel sample with round grains of 50 pixels radius

For example, for the characteristics A_{gr} and A_n the discrepancy between analytical and artificial values does not exceed 1%, which indicates the high accuracy of the analysis methods. However, for the characteristics C_s , and ecr which reflect the shape and size of the grain, the discrepancies between analytical and software values are much larger, reaching almost 15% for C_s and 3.7% for ecr in some cases. These deviations, especially in the case of C_s characteristic, may be the result of inaccurate calculation of the grain perimeter. This indicates the need to improve image processing methods, in particular, to adapt algorithms to the peculiarities of grain shape and size. Further research is aimed at improving the methods for calculating these characteristics to provide more accurate results.

4.2 Analysis of the statistical characteristics of the distribution of grain properties

One of the key tasks in materials science is to analyze the microstructures of materials to study their mechanical properties and behavior. To do this, it is important to identify the relationship between the characteristics of microstructure grains and material properties. A large number of methods are used in modern science to analyze microstructures, among which one of the most powerful tools is violin plots. Violin plots are an effective tool for comparing the distribution of grain characteristics between generated and experimental microstructures. In each scatter plot, the distribution of grain characteristics for the generated microstructures is shown on the left and for the experimental microstructures on the right. In addition, the graph also provides statistical information such as the mean, standard deviation, and median, according to (10)-(12), which provides additional context for comparing the distributions.

In this paper, violin plots are used to compare the distribution of normalized grain area (Fig. 6), grain scale factor (Fig. 7), grain shape factor (Fig. 8), equivalent grain circle radius (Fig. 9), grain orientation angle (Fig. 10) and other characteristics (Fig.11-19).

This allows researchers to gain a deeper understanding of the microstructure of the material and its impact on material properties.

Thus, the use of violin plots to compare the distribution of grain characteristics in generated and experimental microstructures is a powerful tool in materials science and mechanics research.

Table 1
Comparison of characteristics

	Figure 2			Figure 3			Figure 4			Figure 5		
	Analytical	Program	Error	Analytical	Program	Error	Analytical	Program	Error	Analytical	Program	Error
<i>Size, px</i>		200 × 200			250 × 250			200 × 200			300 × 300	
<i>A_{gr}, px</i>	2500	2500	0%	2500	2500	0%	7854	7825	0.4%	7854	7825	0.4%
<i>A_n</i>	0.06	0.06	0%	0.04	0.04	0%	0.2	0.196	0.3%	0.09	0.087	0.2%
<i>P_{gr}, px</i>	200	199	0.5%	200	199	0.5%	314.2	340	8.2%	314.2	340	8.2%
<i>C_s</i>	0.785	0.793	1.1%	0.785	0.793	1.1%	1	0.851	15%	1	0.851	15%
<i>ecr</i>	0.141	0.141	0%	0.113	0.113	0%	0.25	0.249	0.2%	0.16	0.166	3.7%
<i>ψ, rad</i>	±0.79	-0.79	0%	±0.79	-0.79	0%	±0.79	-0.79	0%	±0.79	-0.79	0%
<i>S_c</i>	1	1	0%	1	1	0%	1	1	0%	1	1	0%
<i>I_{xx}</i>	208.3	208.2	0.02%	208.3	208.2	0.02%	625	622.7	3.7%	625	622.7	3.7%
<i>AR</i>	1	1	0%	1	1	0%	1.27	1.25	1.5%	1.27	1.25	1.5%
<i>C_k</i>	1	1	0%	1	1	0%	1.001	1.011	0.9%	1.001	1.011	0.9%
<i>E</i>	0.5	0.75	50%	0.5	0.75	50%	0.79	0.785	0%	0.79	0.785	0%
<i>I_{xx}/A_{gr}</i>	0.083	0.083	0%	0.083	0.083	0%	0.079	0.08	1.3%	0.079	0.08	1.3%

After constructing the violin plots, a comparative analysis of the statistical characteristics of the distribution of grain properties in real and generated microstructures has been carried out. The analysis conducted facilitated the identification of similarities and differences between both types of microstructures, enabling an assessment of the compliance of the generated microstructures with real conditions. The data obtained will be used to further improve the image processing algorithms and virtual reconstruction of grain microstructures. Information on the analysis of statistical characteristics is presented in Table 2.

Conclusions

Throughout this study, the objectives have been attained, yielding significant results conducive to the advancement of image processing methodologies and the analysis of material microstructures.

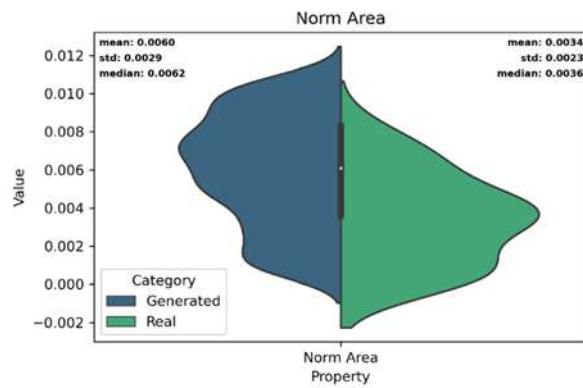


Figure 8: Comparison of normalized grain area distributions

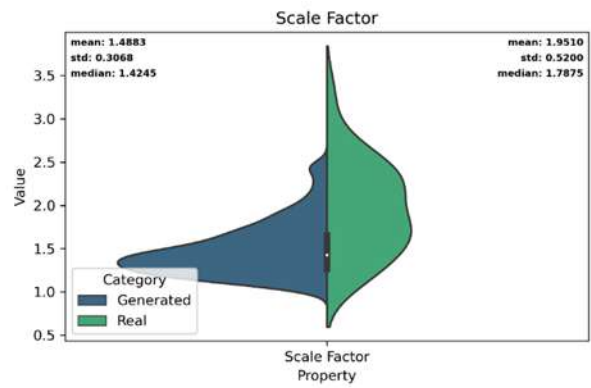


Figure 9: Comparison of grain-scale factor distributions

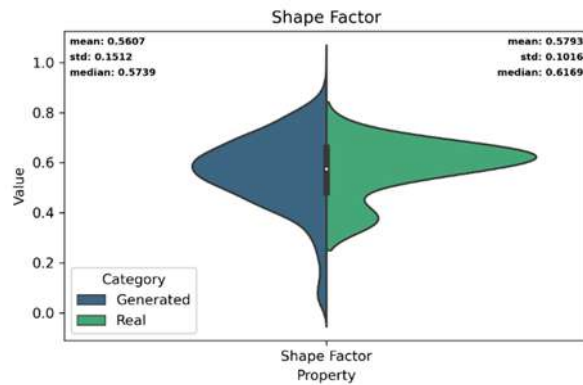


Figure 10: Comparison of grain shape factor distributions

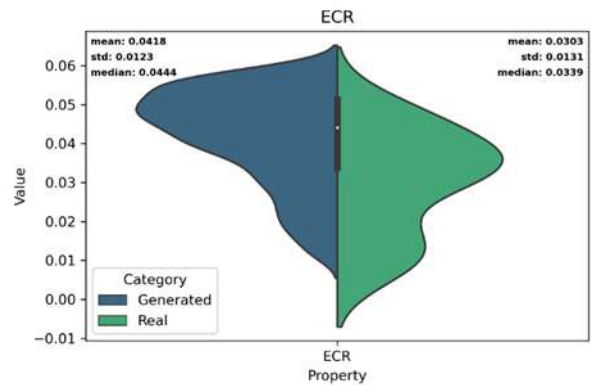


Figure 11: Comparison of the distributions of the grain equivalent circle radius

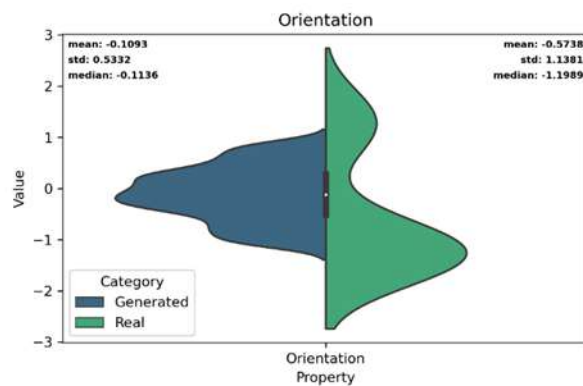


Figure 12: Comparison of grain orientation angle distributions

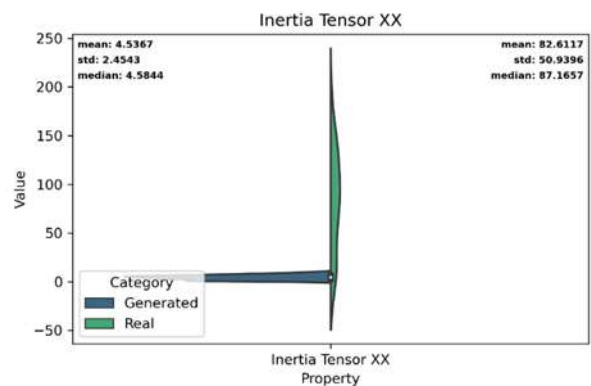


Figure 13: Comparison of grain moment of inertia distributions

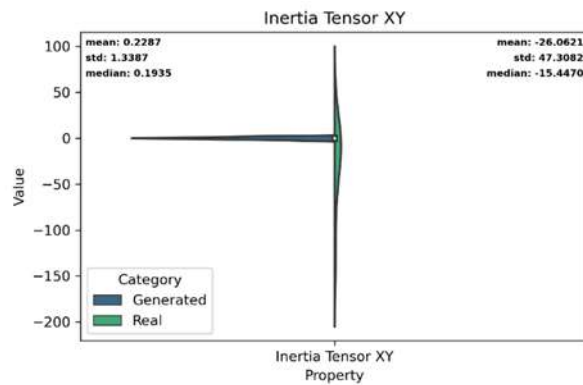


Figure 14: Comparison of grain moment of inertia distributions between grain axes

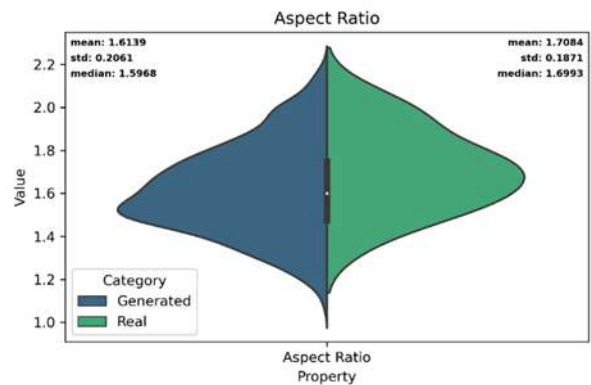


Figure 15: Comparison of grain aspect ratio distributions

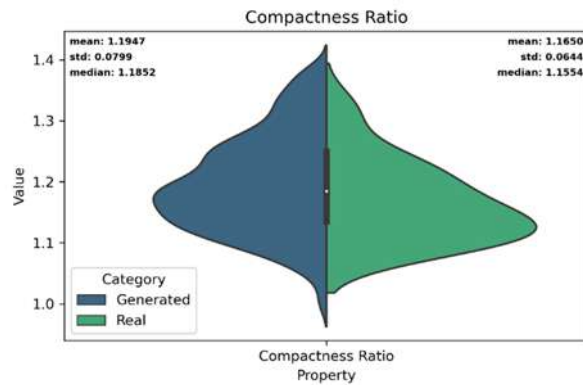


Figure 16: Comparison of grain compactness ratio distributions

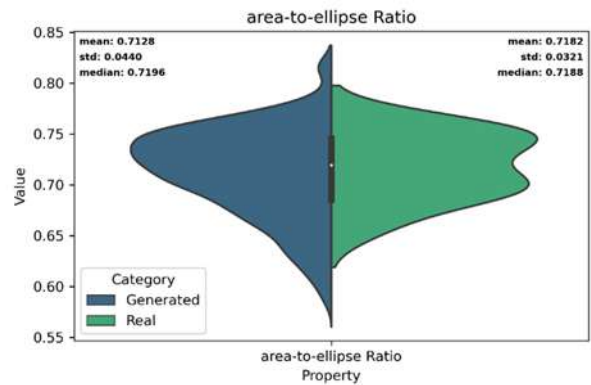


Figure 17: Comparison of the distribution of the ratio of the area to the grain axes

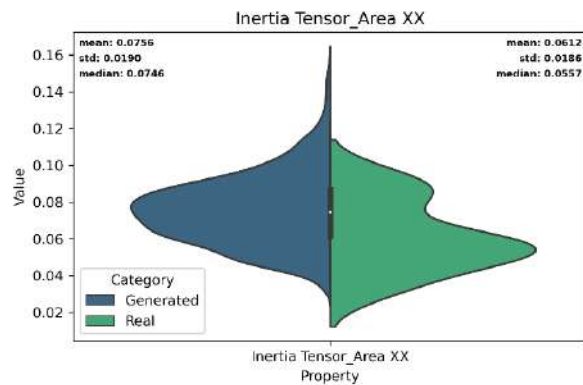


Figure 18: Comparison of distributions of the ratio of the main moment of inertia to the grain area

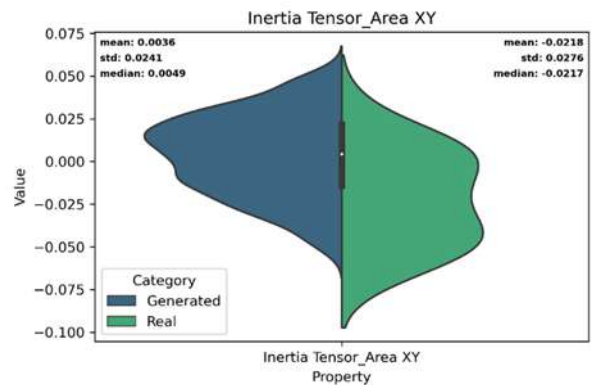


Figure 19: Comparison of distributions of the ratio of the moment of inertia between the axes to the grain area

Table 2
Statistical summary table

	Mean	Standard deviation	Median	Mode	Range	IQR
Generated	Generated	Real	Generated	Real	Generated	Real
A_n	0.003	0.006	0.003	0.004	0.005	0.004
C_s	0.579	0.561	0.151	0.617	0.179	0.091
ecr	0.030	0.042	0.012	0.034	0.018	0.022
ψ	-0.574	-0.109	0.533	-1.199	0.766	1.855
S_c	1.95	1.488	0.307	1.788	0.38	0.744
I_{xx}	82.61	4.537	2.454	87.17	3.705	90.32
I_{xy}	-26.06	0.229	1.339	-15.45	1.864	42.49
AR	1.708	1.614	0.206	1.699	0.276	0.23
C_k	1.165	1.195	0.08	1.155	0.114	0.091
E	0.718	0.713	0.044	0.719	0.061	0.051
$\frac{I_{xx}}{A_{gr}}$	0.061	0.076	0.019	0.056	0.025	0.028
$\frac{I_{xy}}{A_{gr}}$	-0.022	0.004	0.024	-0.02	0.036	0.047

First, image processing algorithms were developed to automatically detect regions corresponding to individual grains in the material microstructure. The successful testing of these algorithms on test images enabled the accurate determination of various characteristics of each grain, including area, shape factor, orientation angle, and so forth.

The resulting grain characteristics were further analyzed and it has been found that dimensionless characteristics, such as normalized area, equivalent to the radius of a grain circle, which is determined based on the normalized area, are most suitable for more efficient comparison of microstructures. The study also showed that other dimensionless quantities, such as the ratio of the inertia tensor to the grain area, the grain wrapping ratio, the ratio of the area to the principal grain axes, and the grain scale factor, are very useful for comparing microstructures, as they will not depend on the size of the image itself. Also, when comparing the characteristics calculated analytically and software, it has been found that grain properties calculated using the perimeter have a larger error, so it is recommended to avoid characteristics such as the grain shape coefficient and use characteristics defined as a ratio to the area, such as the ratio of the inertia tensor to the grain area.

Our study also included the development of an algorithm for comparing grain characteristic distributions based on the analysis of violin plots and the comparison of statistical properties of the distributions. This algorithm will be used to further tune the generation of microstructures using the MatViz3D software package.

The results obtained are an important step towards the further development of image processing methods and virtual reconstruction of grain microstructures. They can be used in further research in the field of materials science and mechanics to gain new knowledge and develop new materials with improved properties.

Acknowledgments

This work has been supported by the Ministry of Education and Science of Ukraine in the framework of the realization of the research project “Algorithms, models, and tools of artificial intelligence for two-level modeling of complex materials behavior for dual-use technologies” (State Reg. Num. 0124U000450).

References

- [1] X. Zeng, Ch. Liu, Ch. Zhao, J. Dong, F. Roters, D. Guan Three-dimensional study of grain scale tensile twinning activity in magnesium: A combination of microstructure characterization and mechanical modeling. *Acta Materialia*, 2023, 255, 119043. <https://doi.org/10.1016/j.actamat.2023.119043>
- [2] M. Sitko, M. Mojzeszko, L. Rychlowski, G. Cios, P. Bala, K. Muszka, L. Madej Numerical procedure of three-dimensional reconstruction of ferrite-pearlite microstructure data from SEM/EBSD serial sectioning. *Procedia Manufacturing*, 2020, 47, pp. 1217-1222. <https://doi.org/10.1016/j.promfg.2020.04.183>
- [3] P. Seibert, A. Raßloff, K. A. Kalina, J. Gussone, K. Bugelnig, M. Diehl, M. Kästner Two-stage 2D-to-3D reconstruction of realistic microstructures: Implementation and numerical validation by effective properties. *Computer Methods in Applied Mechanics and Engineering*, 2023, 412, 116098. <https://doi.org/10.1016/j.cma.2023.116098>
- [4] García-García, V., Mejía, I. and Reyes-Calderón, F. Two-dimensional Monte Carlo–Voronoi simulation of grain growth and nucleation in the heat affected zone of TWIP-Ti welds. *Materialia*, 2019 5, 100223. doi: 10.1016/j.mtla.2019.100223
- [5] Z. Li, J. Wang, H. Huang, Grain boundary curvature based 2D cellular automata simulation of grain coarsening. *Journal of Alloys and Compounds*, 2019, 791, pp. 411–422. doi: 10.1016/j.jallcom.2019.03.195
- [6] H. L. Ding, Y. Z. He, L.F. Liu, W. J. Ding, Cellular automata simulation of grain growth in three dimensions based on the lowest-energy principle. *Journal of Crystal Growth*, 2006, 293(2), pp. 489–497. doi: 10.1016/j.jcrysgr.2006.05.060
- [7] M. Sitko, Q. Chao, J. Wang, K. Perzynski, K. Muszka, L. Madej, A parallel version of the cellular automata static recrystallization model dedicated for high performance computing platforms – Development and verification. *Computational Materials Science*, 2020, 172, 109283. doi: 10.1016/j.commatsci.2019.109283

- [8] O. Vodka, Computer modeling of microstructures with probabilistic cellular automata method using different nucleation rate functions, CMIS, 2020, pp. 450-461 doi: 10.32782/cmris/2608-34
- [9] A. N Hytowitz, Review of using the Dyop optotype for acuity and refractions per the article. Journal of Optometry, 2023, 16(4), pp. 317-318. doi: 10.1016/j.optom.2022.12.002
- [10] M. Bakhtiari, M. S. Salehi, Reconstruction of deformed microstructure using cellular automata method. Computational Materials Science, 2018, 149, pp. 1–13. doi: 10.1016/j.commatsci.2018.02.053
- [11] Y. C. Lin, Y.-X. Liu, M.-S. Chen, M.-H. Huang, X. Ma, Z.-L. Long, Study of static recrystallization behavior in hot deformed Ni-based superalloy using cellular automaton model. Materials & Design, 2016, 99, pp. 107–114. doi: 10.1016/j.matdes.2016.03.050
- [12] N. Maazi, R. Boulechfar, A modified grain growth Monte Carlo algorithm for increased calculation speed in the presence of Zener drag effect. Materials Science and Engineering: B, 2019, 242, pp. 52–62. doi: 10.1016/j.mseb.2019.03.003
- [13] J. Melville, V. Yadav, L. Yang, A. R. Krause, M. R. Tonks, J. B. Harley, A new efficient grain growth model using a random gaussian-sampled mode filter. Materials & Design, 2023, 112604. doi: 10.1016/j.matdes.2023.112604
- [14] J. Pu, N. S. Gezer, S. Ren, A. O. Alpaydin, E. R. Avci, M. Risbano, B. Rivera-Lebron, S. Y.-W. Chan, J. K. Leader, Automated detection and segmentation of pulmonary embolisms on computed tomography pulmonary angiography (CTPA) using deep learning but without manual outlining. Medical Image Analysis, 2023, 102882. doi: 10.1016/j.media.2023.102882
- [15] X. Xu, Z. Yu, W.-Y. Chen, A. Chen, A. Motta, X. Wang, Automated analysis of grain morphology in TEM images using convolutional neural network with CHAC algorithm. Journal of Nuclear Materials, 2024, 588, 154813. doi: 10.1016/j.jnucmat.2023.154813
- [16] Ma, Q. and Wang, Z. Characterizing heterogenous microstructures of fiber-reinforced composite materials using an advanced image processing-based approach through optical microscopic images. Manufacturing Letters, 2023, 35, 1163–1172. doi: 10.1016/j.mfglet.2023.08.123
- [17] Sawai, K., Chen, T.-T., Sun, F., Ogawa, T. and Adachi, Y. Image regression analysis for linking the microstructure and property of steel. Results in Materials, 2024, 21, 100526. doi: 10.1016/j.rinma.2023.100526
- [18] Gao, P., Lv, Z., Song, Y., Song, M. and Qian, P. Evolution of morphology and microstructure of coarsened nanoporous gold studied by automatic thresholding and image recognition algorithms. Scripta Materialia, 2023, 226, 115256. doi: 10.1016/j.scriptamat.2022.115256
- [19] Ding, Y., Camanho, P. P. and Silva, A. Comparison of three algorithms generating virtual microstructures in terms of the degree of randomness. Composites Part A: Applied Science and Manufacturing, 2023, 107959. doi: 10.1016/j.compositesa.2023.107959
- [20] Cai, C., Zhang, T., Wang, X., Yin, W., Xu, Z., Wang, R. and He, X. A versatile and highly efficient algorithm to generate representative microstructures for heterogeneous materials. Composites Science and Technology, 2023, 110138. doi: 10.1016/j.compscitech.2023.110138
- [21] CLEMEX VISION, P. E. Clemex intelligent microscopy, Quebec, Canada.
- [22] Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., ... Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. Nature Methods, 9(7), 676–682. doi:10.1038/nmeth.2019

Resilience-aware MLOps for resource-constrained AI-system

Viacheslav Moskalenko¹, Alona Moskalenko¹, Anton Kudryavtsev¹ and Yuriy Moskalenko¹

¹ Sumy State University, 116, Kharkivska st., 40007 Sumy, Ukraine

Abstract

Artificial intelligence systems are increasingly used in security-critical applications with limited computing resources, which makes them vulnerable to such disturbances as adversarial attack noise, out-of-distribution data, and fault injections. To absorb disturbances and adapt the AI system during its life cycle, it is necessary to expand the Machine learning operations structure with stages related to the implementation of resilience mechanisms. In this case, increasing resilience in one form or another is associated with the introduction of redundancy in the form of additional resources to absorb disturbances and quickly recover performance. It is proposed to provide Affordable Resilience for resource-constrained AI systems by implementing a resilience optimization stage and adding add-ons with a small number of parameters that will allow for uncertainty calibration and rapid adaptation to labeled and unlabeled data. This approach is also intended to separate the work related to the development and deployment of the basic AI model that solves the applied problem and the work related to ensuring resilience in the Machine learning operations structure. The experiments were conducted on CIFAR-10 and CIFAR-100 datasets using the MobileViT network, which is a modern network architecture for visual image analysis in conditions of limited computing resources. We have experimentally confirmed the increase in the resilience of an AI system at different stages of its life cycle by implementing the stages of resilience optimization, uncertainty calibration, and Test-Time Adaptation. Post-hoc resilience optimization improved robustness to fault injection by 5% and robustness to adversarial attack by 7%. Moreover, tuning with 10% of the test data allowed for an additional 6% increase in robustness to fault injection and 7.1% increase in robustness to adversarial attack on the new data. Also, the use of post-hoc resilience optimization increased the integral indicator of resilience to task changes by 10.5%. Post-hoc uncertainty calibration makes it possible to further increase the robustness of fault injection models by an average of 4.4% and the robustness to adversarial attacks by an average of 1.3%. Test-Time Adaptation increases robustness to Fault Injection by 6.9% and robustness to Adversarial Attack by 4.72%.

Keywords

MLOps, Resilience, Artificial Intelligence, Confidence Calibration, Optimization, Test-Time Adaptation

1. Introduction

Artificial intelligence (AI) systems are increasingly deployed on safety-critical but resource-constrained devices (such as unmanned aerial vehicles, autonomous robots, autopilots). All AI systems to some extent are susceptible to data noise through adversarial attacks, novelty in data, concept drift, and the injection of faults into the memory of neural weight [1]. Moreover, AI systems with constrained resources are more vulnerable to disturbances. The ability to absorb disturbances (robustness), graceful degradation due to the impact of disturbances that could not be absorbed, and rapid adaptation to new disturbances are considered to be the key features of resilient system [2].

Traditional approaches in Machine Learning Operations (MLOps) predominantly emphasize data integrity, model efficiency, and security aspects but often fall short in tackling resilience issues as highlighted in [3]. Considering the critical decisions entrusted to AI systems, insufficient resilience may lead to severe implications, including unreliable performance and financial losses.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ v.moskalenko@cs.sumdu.edu.ua (V. Moskalenko); a.moskalenko@cs.sumdu.edu.ua (A. Moskalenko); (A. Kudryavtsev), yuriy.mosk@gmail.com (Y. Moskalenko)

ORCID iD 0000-0001-6275-9803 (V. Moskalenko); 0000-0003-3443-3990 (A. Moskalenko); 0000-0003-0967-0185 (A. Kudryavtsev); 0009-0002-3635-3337 (Y. Moskalenko)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Therefore, this study aims to enrich MLOps methodology by incorporating resilience as a fundamental component.

The object of the research is MLOps process under resource constraints and the impact of various types of disturbances on the AI system. **The subjects of the research** are the architecture of MLOps and methods of ensuring the resilience of an AI system with limited resources to various types of disturbing influences. **The goal** is to develop a new MLOps methodology that ensures the resilience of resource-constrained AI-system to such negative factors as adversarial attacks, fault injections, drift, and out-of-distribution of data.

2. Related works

The concept of resilience in AI systems is not new and has been examined across various fields, including cybersecurity, manufacturing, and even autonomous vehicles. Techniques like adversarial training [4], fault-aware training [5], and uncertainty quantification [6] have been employed to improve resilience. The paper [7] proposes the concept of Secure Machine Learning Operations paradigm, but without proposals for effectively protecting the same AI system from different types of threats. The issue of ensuring compatibility and computational efficiency of combining different methods to provide resilience and efficiency of the AI system is not considered.

A number of works consider the aspects of the MLOps methodology for AI systems with resource constraints [8, 9]. In addition to typical MLOps aspects, more attention is focused on model optimization by quantization or pruning of weights and knowledge distillation. However, the online adaptation of resource-constrained AI systems to changes is considered only for shallow machine learning models or within the framework of federated learning, which requires network communication with distributed nodes. In [2, 4], various types of destructive disturbances for AI systems are considered, to which systems with limited resources are most vulnerable. However, most existing MLOps frameworks are designed to ensure efficient operation rather than resilience to various disturbances in resource-limited environments.

The papers [10, 11] consider the use of Parameter-Efficient Tuning methods as one of the effective approaches to increase computational efficiency and speed of adaptation of AI system to changes. The papers [12, 13] consider approaches to Test-Time Adaptation, which increases the efficiency of adaptation to novelty in the absence of labeled data. The papers [14, 15] consider meta-learning algorithms for increasing the robustness and efficiency of few-shot learning. These methods are promising to be combined in the MLOps methodology, but the possible configurations of their combination are not well studied.

3. Resilience-aware MLOps architecture

Key concepts in MLOps include the delineation of duties and the collaboration among different teams. Specialized platform-level solutions that bolster the resilience of any AI model assign the task of updates and maintenance to a dedicated team of AI resilience specialists. Additionally, to enhance the delineation of responsibilities, new stages in MLOps that focus on resilience should be introduced as subsequent (post-hoc) procedures.

Figure 1 shows a diagram of the proposed resilience-aware MLOps, which additionally includes the stages of Post-hoc Resilience Optimization, Post-hoc Uncertainty Calibration, Uncertainty Monitoring, Test-Time Adaptation and Graceful Degradation. In addition to Uncertainty Monitoring, the Explainable AI mechanism can be used to assist decision-making by the human to whom control is delegated in case of uncertainty. The article [16] questions the adequacy of existing methods of explaining decisions, so the explanation mechanism will be excluded from further consideration, but for generality, the diagram shows this MLOps stage.

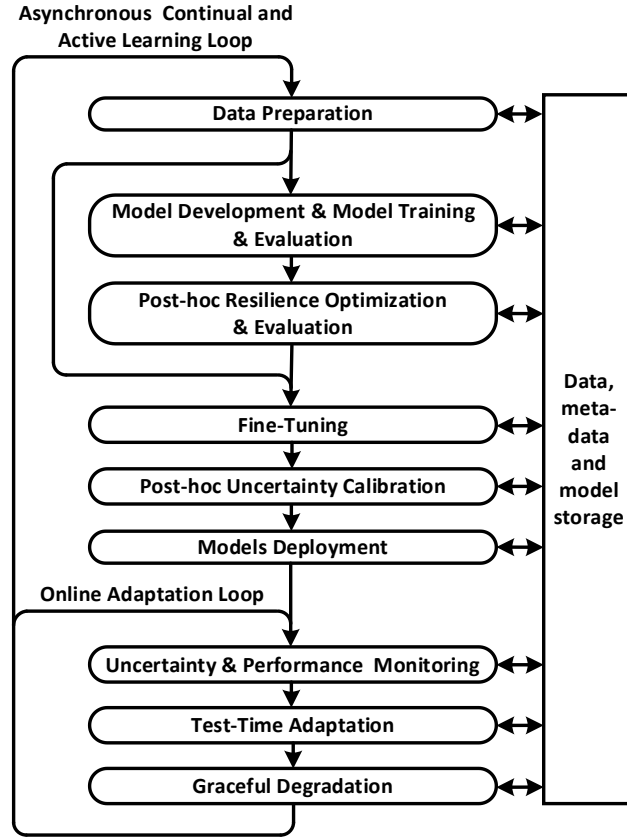


Figure 1: Basic stages of resilience-aware MLOps

In the phase focused on enhancing resilience, it is suggested to attach computationally efficient (meta-)adapters to the existing model to improve robustness and speed up fine-tuning [11]. During this enhancement, the weights of the original model are maintained unchanged. Typically, the original model comprises specific units or modules, such as a ResNet Block or MobileViT Block. To generalize, we will refer to these blocks as frozen operations and denote them as $OP(x)$. The parallel method of connecting an adapter to the frozen blocks of the model is the most convenient and versatile approach (Figure 2a). In this case, to ensure the properties of resilience, it is proposed to use three consecutive blocks of adapters at once, two of which are tuned during meta-training [10]. To balance between different modules, we introduce a channel-wise scaling.

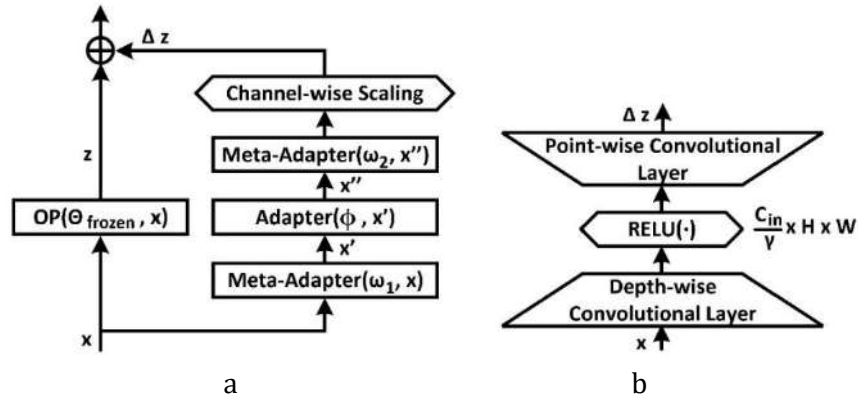


Figure 2: Design of Parallel Adapter. (a) Parallel correction scheme for the frozen block; (b) Model architecture of adapter or meta-adapter

The adapter designs shown in Figure 2b utilize convolutional layers. The convolutional adapter shown in Figure 2b can adjust channel compression by a factor of 1, 2, 4, or 8 using the γ hyperparameter.

The base model is trained on the original dataset $D_{base} = \{D_{base}^{tr}; D_{base}^{val}\}$ to perform the main task under normal conditions. Resilience optimization requires generating a set of synthetic

disturbance implementations $\{\tau_i \mid i = \overline{1, N}\}$ [2]. As disturbances τ_i can be considered adversarial attacks, fault injection, or switching to a new task. In addition, it is necessary to provide datasets $D = \{D_k^{tr}; D_k^{val} \mid k = \overline{1, K}\}$, that solve other problems for K few-shot learning tasks, where fine-tuning data D_k^{tr} is used in the fine-tuning stage and validation set D_k^{val} is used in the meta-update stage. There is given a set of parameters θ, ϕ, ω and W , where θ are parameters of a pretrained and frozen base AI model, ϕ and ω are adaptation parameters of AI model backbone, and W are task specified head parameters. Head weights W_{base} for the main task are pre-trained on the data D_{base} . To simplify the description of the problem, we denote the set of all parameters as $\mathcal{E} = \langle \theta, \phi, \omega, W \rangle$. Then the meta-learning process for direct maximization of the expected resilience indicator can be described by the formula [2, 17]

$$\mathcal{E}^* = \underset{\mathcal{E}}{argmax} \underset{\tau_i \sim p(\tau)}{E} [R_{\tau_i}(U(\mathcal{E}, D))] = \underset{\mathcal{E}}{argmax} F(\mathcal{E}), \quad (1)$$

where U is an operator that combines disturbance generation and adaptation in T steps, which maps the current state of ϕ to the new state of ϕ ;

R_{τ_i} is a function that calculates the value of the resilience indicator on test sample $D_{\tau_i}^{val}$ for τ_i disturbance implementation over model parameters ω during its adaptation by formula

$$R_{\tau_i} = \frac{1}{P_0 T} \sum_{t=1}^T P_{\tau_i}(\theta, \omega, \phi_t, W_t, D_{\tau_i}^{val}), \quad (2)$$

where P_{τ_i} is a performance metric for current state of model parameters and evaluation data.

In the implementation of the operator U , it is proposed to use the SGD stochastic gradient descent algorithm with T steps. The results of adaptation according to the SGD algorithm are proposed to be used for meta-updating the gradient in the outer loop. The metagradient is estimated using a Gaussian-smoothed version of the outer loop objective and is calculated according to the formula [2, 13]

$$\nabla_{g \sim N(0, I)} \underset{g \sim N(0, I)}{E} [F(\mathcal{E} + \sigma g)] = \frac{1}{2\sigma} E[R(\mathcal{E} + \sigma g) - R(\mathcal{E} - \sigma g)]. \quad (3)$$

A perturbation vector g is generated for the meta-optimized parameters at the beginning of each meta-optimization iteration. Thus, the pseudocode of the proposed resilient-aware meta-learning algorithm is shown in Figure 3.

As shown in Figure 3, one type of destructive influence is used within one step of meta-adaptation. Each step of the meta-adaptation process starts with a random selection of the disturbance type. Then, n implementations of the disturbance are generated, followed by a nested loop of adaptation to each disturbance. Mixing different disturbances at once might not be effective. For example, following the injection of faults into the weights of the neural network, we would obtain a model that is no longer relevant for conducting adversarial attacks.

The *Adv_perturbation()* function is used to generate adversarial samples. Adversarial training of differentiated models can be performed using FGSM attacks or other white-box attacks [15]. However, to test models, it is proposed to use attacks based on the covariance matrix adaptation evolution strategy (CMA-ES) algorithm, which are more universal for any model [18]. The amplitude of the perturbation is limited by the L_∞ -norm or L_0 -norm. If the image is normalized by dividing the pixel brightness by 255, then the specified perturbation amplitude is also divided by 255.

The introduction of faults is carried out through the *Fault_injection()* function [19]. It is recommended to select the fault type that poses the greatest difficulty for absorption, which entails generating an inversion of a randomly chosen bit (bit-flip injection) within the model's weight. For models that exhibit differentiability, it is advisable to pass the test dataset through the network and compute the gradients, which can subsequently be sorted according to their absolute values. In the top- k weights exhibiting the highest gradient magnitudes, a single bit is inverted at a random position. The proportion of weights subjected to the inversion of a random bit can be denoted as the fault rate.

Changing tasks can be considered as a way to mimic the concept drift and out of distribution data. The selection of other tasks can be done either by randomizing the domain of the base task or by randomizing tasks of the same domain, or in combination.

Augmented training data can be used to improve calibration on in-distribution data. Data from other datasets with semantically different annotations can be used to generate out-of-

distribution data. Also, for experimental research, Soft Brownian Offset with an autoencoder can be used to generate out-of-distribution data [20].

```

Require:    Distribution over disturbances  $p(\tau)$ ; Step size hyperparameters  $\alpha, \beta$ ;
            Precision parameter  $\sigma$ ; Number of adaptation steps  $T$ .

1  Pretrain  $\phi, \omega$  on original data  $D_{base}$ 
2  While not done do:
3      Select type of disturbance from set { fault injection, evasion adversarial attack, task change}
4      Sample disturbance implementations  $\tau_i \sim p(\tau), i = \overline{1, n}$ 
5      Sample perturbation vectors  $g_{\phi, \tau_i} \sim N(0, I), i = \overline{1, n}; g_{\omega, \tau_i} \sim N(0, I), i = \overline{1, n}$ 
6      For  $i=1, 2, \dots, n$  do:
7          Clone the current parameters:  $\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i}, \hat{\phi}_{\tau_i}, \hat{W}_{\tau_i} \leftarrow copy(\theta, \omega, \phi, W_{base})$ 
8           $\hat{\phi}_{\tau_i+} \leftarrow \hat{\phi}_{\tau_i} + \sigma g_{\phi, \tau_i}; \hat{\phi}_{\tau_i-} \leftarrow \hat{\phi}_{\tau_i} - \sigma g_{\phi, \tau_i}$ 
9           $\hat{\omega}_{\tau_i+} \leftarrow \hat{\omega}_{\tau_i} + \sigma g_{\omega, \tau_i}; \hat{\omega}_{\tau_i-} \leftarrow \hat{\omega}_{\tau_i} - \sigma g_{\omega, \tau_i}$ 
10         If disturbance type is a task change:
11             Sample the training and validation data  $D_{\tau_i}^{tr}, D_{\tau_i}^{val}$  from new task
12         else:
13             Sample the training and validation data  $D_{\tau_i}^{tr}, D_{\tau_i}^{val}$  from  $D_{base}$ 
14         If disturbance type is a fault injection:
15              $\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i+}, \hat{\omega}_{\tau_i-}, \hat{\phi}_{\tau_i+}, \hat{\phi}_{\tau_i-} \leftarrow \text{Fault\_injection}(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i+}, \hat{\omega}_{\tau_i-}, \hat{\phi}_{\tau_i+}, \hat{\phi}_{\tau_i-})$ 
16         If disturbance type is an evasion adversarial attack:
17              $D_{\tau_i}^{tr}, D_{\tau_i}^{val} \leftarrow \text{Adversarial\_perturbation}(D_{\tau_i}^{tr}, D_{\tau_i}^{val})$ 
18              $\{\hat{\phi}_{\tau_i+, t} | t = \overline{1, T}\} \leftarrow SGD_{\phi, W}(L_{\tau_i}(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i+}, \hat{\phi}_{\tau_i+}, \hat{W}_{\tau_i}, D_{\tau_i}^{tr}), T, \alpha)$ 
19              $\{\hat{\phi}_{\tau_i-, t} | t = \overline{1, T}\} \leftarrow SGD_{\phi, W}(L_{\tau_i}(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i-}, \hat{\phi}_{\tau_i-}, \hat{W}_{\tau_i}, D_{\tau_i}^{tr}), T, \alpha)$ 
20              $R_{+, \tau_i} \leftarrow R(\{P_t(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i+}, \hat{\phi}_{\tau_i+, t}, D_{\tau_i}^{val})\})$ 
21              $R_{-, \tau_i} \leftarrow R(\{P_t(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i-}, \hat{\phi}_{\tau_i-, t}, D_{\tau_i}^{val})\})$ 
22              $R_{\tau_i} \leftarrow \frac{1}{2}(R_{+, \tau_i} - R_{-, \tau_i})$ 
23          $\omega \leftarrow \omega + \beta \frac{1}{\sigma n} \sum_{i=1}^n R_{\tau_i} g_{\omega, \tau_i}$ 
24          $\phi \leftarrow \phi + \beta \frac{1}{\sigma n} \sum_{i=1}^n R_{\tau_i} g_{\phi, \tau_i}$ 

```

Figure 3: Pseudocode of model-agnostic meta-learning with evolution strategies for AI-system resilience optimization

The post-hoc confidence calibration algorithm necessitates the integration of certain supplementary components to the frozen model, which are tuned on the calibration data to minimize the discrepancy between the predicted confidence and the actual probability. Calibration enhancements for classification models encompass techniques such as Isotonic Regression, Histogram Binning, Bayesian neural networks, etc [21].

Despite the existence of preparatory stages in the form of resilience optimization and confidence calibration, unexpected errors in input or model weight, unexpected changes in the domain or other distributional shifts can always occur. A promising approach to mitigate such disturbances is to use the ideas and methods of Test-Time Augmentation and Test-Time Adaptation [12]. It is proposed to calculate the entropy of the marginal probability at the model output for its tuning over a certain number of iterations for low confidence predictions by analogy with scientific research [13]. But instead of tuning the entire model, it is proposed to tune only the adapters. The loss function for test-time adaptation to the input exemplar x which may belong to a certain category from the set Y , will be

$$l(\theta, x) \approx H(\bar{p}_\theta(\cdot | x)) = \sum_{y \in Y} \bar{p}_\theta(y|x) \log \bar{p}_\theta(y|x), \quad (4)$$

where $\bar{p}_\theta(y|x)$ is an estimate of the marginal probability at the model output calculated by the formula

$$\bar{p}_\theta(y|x) = \frac{1}{B} \sum_{i=1}^B p_\theta(y|\tilde{x}_i), \quad (5)$$

where $\tilde{x}_i$ is the i -th augmented version of the input exemplar.

If, after tuning, the entropy at the model output does not exceed the threshold value, it is necessary to switch to the graceful degradation mode. The easiest way to use graceful degradation is to hand over control to a human to fine-tune the system on labeled samples.

Active learning is part of the feedback loop in our MLOps diagram. Unlike traditional MLOps, it is not the base model that is fine-tuned, but the adapters and meta-adapters. Re-training of the base model and resilience optimization can be performed when a sufficient amount of new labeled data is accumulated.

Test-time adaptation can be performed autonomously in Online Loop, and its results can be stored in the local storage. Active Learning requests can be processed asynchronously or synchronously depending on the available service resources.

4. Experiments

Experimental research is performed using the CIFAR-10 and CIFAR-100 datasets that contain annotated images [22]. The CIFAR-10 dataset contains 10 classes and will be considered as the main task dataset. The CIFAR-100 dataset contains 100 classes and therefore can be used as a source of data describing additional tasks for few-shot learning in the meta-learning process. The few-shot learning task has 10 classes that are randomly selected from the set of available classes ($n_{way} = 10$). It is suggested to use $T=40$ iterations to adapt to disturbances, each iteration processing a mini-batch of 4 images ($mini\_batch\_size=4$). It is proposed to use 16 images for each class ($k_{shot}=16$) to balance the data during adaptation. The learning rate of the inner loop and the outer loop of the resilient-aware meta-learning algorithm is $\alpha=0.001$ and $\beta=0.0001$, respectively. The maximum number of iterations of the outer loop of the meta-learning algorithm is 300. However, the meta-learning can be stopped earlier if the average integral resilience criterion does not increase for 10 consecutive iterations. Experiments used one of the modern architectures of visual transformers Mobile ViT, which is popular in machine vision tasks [23]. Adapters and meta-adapters are connected in parallel to each Mobile ViT Block. The following parameters are used in the Test-Time Adaptation algorithm: learning rate is 0.001; optimizer is SGD.

The effect of each additional resilience-aware MLOps stage on the accuracy and speed of accuracy recovery is investigated by analyzing several MLOps configurations:

- Config 0 is a traditional MLOps with a fine-tuning stage based on active feedback loop data;
- Config 1 is an improvement of Config 0 by adding a fault tolerance optimization stage;
- Config 2 is an improvement of Config 1 by adding a stage of forecast uncertainty calibration;
- Config 3 – upgraded Config 2 with Test-Time Adaptation stage.

CIFAR-10 datasets contain training, validation, and test subsamples. To simplify the experiment and analyze the results, we will divide it into 4 parts. Each part of the test data is needed to simulate a part of the AI model's life cycle. Let's consider 4 consecutive parts of the life cycle:

- Test 0 – training the AI model on the training dataset and testing the model on the first part of the test dataset, followed by selecting 10% of the test data points with the highest uncertainty;
- Test 1 – fine-tuning the AI model on the selected test data points from the previous test and testing the model on the second part of the test dataset under the disturbance, followed by selecting 10% of the test data points with the highest uncertainty;
- Test 2 – fine-tuning the AI model on selected data points from the previous test and testing the model on the third part of the test dataset under the disturbance, followed by selecting the 10% of test data points with the highest uncertainty;
- Test 3 – fine-tuning the AI model on selected test data points from the previous test and testing the model on the fourth part of the test dataset under increased disturbance intensity.

The graceful degradation mechanism is proposed to be implemented in the simplest form by rejecting a decision if the entropy threshold is exceeded. Therefore, the control is transferred to

a human or a larger and more powerful AI model. In this case, we consider the accuracy of the model, which is calculated in two ways:

- ACC1 is the accuracy of the AI system taking into account all test examples;
- ACC2 is the accuracy that does not take into account the examples for which the decision was rejected due to high uncertainty.

Conventional MLOps reject decisions based on predictive confidence, while resource-aware MLOps reject decisions based on uncertainty.

For training adapters with meta-adapters, fault injection is carried out by selecting weights with the largest absolute gradient values. The proportion of modified weights is $\text{fault\_rate} = 0.1$. For testing the resulting model, fault injection will be performed by random bit-flips in randomly selected weights, the proportion of which (fault_rate) are equals to 0.1 or 0.15.

The training of the tuners and meta-tuners involves generating adversarial samples using the FGSM algorithm with $\text{perturbation\_level}$ according to L_∞ up to 3. However, to test the resulting model against adversarial attacks, the adversarial samples are generated using the CMA-ES algorithm with $\text{perturbation\_level}$ according to L_∞ -norm are equals to 3 or 5. The number of solution generation in the CMA-ES algorithm is set to 10 to reduce the computational cost of conducting experiments.

Instead of directly modeling different types of concept drift or novelty in the data, it is proposed to model the ability to quickly adapt to task changes, as this can be interpreted as the most difficult case of real concept drift. The preparation for the experiment involved adding adapters and meta-adapters to the network, which had been trained on the CIFAR-10 dataset. During meta-training, these adapters performed adaptations to either attacks or a 10-class classification task, which was randomly generated from a selection of the CIFAR-100 set. Subsequently, to verify the capability for rapid adaptation to a new task change, the new task was considered either as a classification task with the full set of CIFAR-100 classes. The resilience curve is constructed over 40 mini-batch fine-tunings, from which the resilience criterion (2) is calculated.

Taking into account the elements of randomization, it is proposed to use their average values when assessing the accuracy of the model. To this end, 100 implementations of a certain type of disturbance are generated and applied to the same model or data.

Uncertainty calibration will be performed on a dataset containing augmented test samples and out-of-distribution samples generated by Soft Brownian Offset Sampling. 300 images per class are generated for in-distribution test set to calibrate the uncertainty. The total number of out-of-distribution images is the same as the in-distribution calibration set. In this case, the Soft Brownian Offset Sampling algorithm is used with the following parameter values: minimum distance to in-distribution data is equal to 25; offset distance is equal to 20; softness is equal to 0. Bayesian Binning into Quantiles with 10 bins was chosen as the calibration algorithm.

5. Results

Table 1 shows the average values of accuracy (ACC1) and accuracy excluding rejected solutions (ACC2) at different life cycle successive stages of the MobileViT model with add-ons under fault injection for each MLOps configuration. The average accuracies in Table 1 are calculated based on 100 repetitions of the experiments to reduce the effect of randomization.

Experimental data confirm the increase in fault tolerance for configuration 1 (with resilience optimization) compared to configuration 0 and configuration 2 (with uncertainty calibration) compared to configuration 1. The dynamics of accuracy growth during adaptation (Test 1-Test 2) is higher for Config 1 and Config 2, and Config 3 is characterized by the highest accuracy values. In the last two configurations, even an increase in the fraction of damaged tensors does not lead to a significant drop in accuracy compared to the previous configurations. Also, when comparing ACC2 with ACC1, it is noticeable that ACC2 is always larger than ACC1. Config 3 ensures recovery and improved accuracy even as fault injection intensity increases. Note that the averaged values of ACC1 and ACC2 for the MobileViT-based model on Test 0-Test 3 test data with the corresponding fault injection rate without fine-tuning on 10% of human-labeled examples are 0.811 and 0.878, respectively. It proves the importance of using an active feedback loop for

adaptation. For the average accuracy values in Table 1, the margin of error does not exceed 1% at a 95% confidence level.

Table 1

Accuracy of the MobileViT-based model under the influence of fault injection during the life cycle depending on the MLOps configuration

MLOps configuration	Test 0		Test 1		Test 2		Test 3	
	ACC1	ACC2	ACC1	ACC2	ACC1	ACC2	ACC1	ACC2
Config 0	0.925	0.929	0.802	0.855	0.837	0.887	0.829	0.881
Config 1	0.929	0.933	0.852	0.870	0.891	0.922	0.889	0.920
Config 2	0.938	0.941	0.860	0.890	0.917	0.922	0.903	0.918
Config 3	0.938	0.947	0.920	0.925	0.930	0.943	0.929	0.941

Table 2 shows the average values of accuracy (ACC1) and accuracy excluding rejected solutions (ACC2) at different life cycle successive stages of the MobileViT model with add-ons under adversarial evasion attacks for each MLOps configuration. The average accuracies in Table 1 are calculated based on 100 repetitions of the experiments to reduce the effect of randomization.

Table 2

Accuracy values of the MobileViT-based model under adversarial attack during the life cycle depending on the MLOps configuration

MLOps configuration	Test 0		Test 1		Test 2		Test 3	
	ACC1	ACC2	ACC1	ACC2	ACC1	ACC2	ACC1	ACC2
Config 0	0.925	0.929	0.720	0.775	0.787	0.817	0.819	0.821
Config 1	0.929	0.933	0.782	0.820	0.891	0.922	0.889	0.920
Config 2	0.938	0.941	0.805	0.830	0.917	0.922	0.903	0.918
Config 3	0.938	0.947	0.850	0.915	0.890	0.923	0.923	0.929

Experimental data confirm the increase in robustness for configuration 1 (with resilience optimization) compared to configuration 0 and configuration 2 (with uncertainty calibration) compared to configuration 1. The dynamics of accuracy growth during adaptation (Test 1-Test 2) is higher for Config 1 and Config 2, and Config 3 is characterized by the highest accuracy values. Traditional MLOps (config 0) also showed the ability to quick adaptation during fine-tuning (results of Test 1 and Test 2), but it was not successful in performance recovery. Config 1 - Config 3 show a noticeable recovery in accuracy. Increasing the magnitude of the perturbation (test 3) leads to a decrease in accuracy in all configurations, while config 1 - config 3 demonstrate greater resilience compared to configuration 0. Config 3 also provides recovery and improved accuracy even as adversarial attack intensity increases. Note that the averaged values of ACC1 and ACC2 on perturbed test data from Test 0-Test 3 stages without fine-tuning on 10% of human-labeled examples are 0.791 and 0.802, respectively. It also proves the importance of using an active feedback loop for adaptation. For the average accuracy values in Table 2, the margin of error does not exceed 1.2% at a 95% confidence level.

To evaluate the robustness and speed of adaptation of a pre-configured AI system to concept drift, it is proposed to calculate the integral resilience criterion (2) in fine-tuning mode (few-shot learning) on 10 class subset of CIFAR-100 set (Table 3).

Table 3

The value of the integral resilience criterion (2) to the change of the medical image analysis task depending on the MLOps configuration

MLOps configuration	$\bar{R}$
Config 0	0.751
Config 1	0.830

Analysis of Table 3 shows that adding a resilience optimization stage to MLOps increases resilience to concept drift, i.e., robustness and speed of adaptation. For the average accuracy values in Table 3, the margin of error does not exceed 1.1% at a 95% confidence level.

According to Table 1, the resilience optimization increased the model's robustness to fault injections (inversion of one randomly selected bit in each of 10% of randomly selected weight tensors) by 5% on average. After tuning by 10% in the first part of the test data, a 6% increase in robustness to fault injections is demonstrated on the next part of the test data, even with an increase in the intensity of fault injections (15% of randomly selected weight tensors are damaged) compared to the configuration without resilience optimization. Similarly, according to Table 2, resilience optimization increases the model's robustness to adversarial attacks (maximum amplitude of 3 according to L_∞ -norm) by 7%. After tuning on 10% of the first part of the test data, a 7.1% increase in robustness is demonstrated even after increasing the disturbance intensity (maximum amplitude of 5 according to L_∞ -norm) compared to a configuration without using resiliency optimization. The results of Table 3 show that the use of resilience optimization increases the integral resilience indicator during adaptation to task changes by 10.5% compared to the configuration without resilience optimization.

The analysis of Table 1 and Table 2 shows that the application of Post-hoc Uncertainty Calibration makes it possible to further improve the model's accuracy under the influence of fault injection by an average of 4.4%, and to improve the model's accuracy under the influence of adversarial attack by an average of 1.3%. The analysis of Table 1 shows that The Test-Time Adaptation improved the model's accuracy under the influence of fault injections by an average of 6.9%. Even with an increase in the intensity of fault injections, the obtained classification accuracy using Test-Time Adaptation is approximately equal to the accuracy of the model under normal conditions without additional add-ons. Similarly, the analysis of Table 2 shows that The Test-Time Adaptation increased the model's accuracy under adversarial attack by an average of 4.72%. Moreover, even with an increase in the intensity of the attack, the model's accuracy is close to its accuracy without the influence of disturbances and without the use of add-ons.

6. Discussion

The proposed framework for resilience-aware MLOps facilitates the implementation of diverse specific solutions for its distinct stages and mechanisms. The central concept revolves around segregating the responsibilities of developers focused on crafting the foundational AI model, optimized for nominal operating conditions, and experts tasked with ensuring the intelligent system's resilience against disturbances and changes. Developers of the core AI model are typically burdened with accounting for data specifics, data collection methodologies, and the application itself to tackle the applied data analysis challenge. Addressing issues pertaining to AI resilience, encompassing security, trustworthiness, robustness, and rapid adaptation to changes, relies on specialized expertise detached from a particular application domain [4]. The primary obstacle in separating these tasks stems from the lack of universality in resilience-ensuring methods and an incomplete comprehension of the compatibility among methods that cater to different aspects of resilience and protection against diverse types of disturbances [2]. Determining the compatibility of these methods and combining them judiciously could augment flexibility and resilience in accordance with requirements and constraints.

The proposed implementation of Post-hoc Resilience Optimization represents merely one of the viable solutions that demonstrates the fundamental feasibility of segregating the development stage of a basic AI model tailored for normal conditions from the supplementary components aimed at ensuring resilience against disturbances and changes. The significance of employing the proposed Post-hoc Uncertainty Calibration stage has been experimentally substantiated. This stage enables, firstly, the detection of disturbances and, secondly, the accurate assessment and tolerance of uncertainty. The Test-Time Adaptation stage allows for real-time enhancement of robustness to various minor changes in input data or weights.

Unlike many existing MLOps methodologies, this approach adapts to changes at the level of adapters. In this case, adapters can be tuned both on labeled data during the fine-tuning stage and on unlabeled data during the Test-Time Adaptations stage. This ensures the continuity of the adaptation process regardless of the frequency of feedback. The size and architecture of the

adapters can vary depending on the architecture of the base model and available resources. The key is that the small size of the adapters allows for adaptation on constrained resources. The preparatory stage of resiliency optimization is able to configure meta-adapters in such a way that adaptation using adapters is accelerated.

7. Conclusion

7.1. Summary

The structure of resilience-aware MLOps for resource-constrained AI-systems has been proposed. The main novelty lies in the separate work on creating a basic model for normal operating conditions and work on ensuring its resilience. This is significant for the many industries, as the developer of the basic model should devote more time to comprehending applied field at hand, rather than specializing in a specific area of resilient systems. Therefore, Post-hoc Resilience Optimization, Post-hoc Predictive Uncertainty Calibration, Uncertainty Monitoring, Test-Time Adaptation and Graceful Degradation are used as additional stages of MLOps.

Resilience optimization aims to maximize robustness to disturbances and the ability to adapt quickly. Fault injection attack, adversarial evasion attack, and concept drift are considered as disturbances to the AI system. Additional add-ons in the form of adapters and meta-adapters are used to optimize the resilience of the AI system. Meta-adapters are updated based on meta-gradient calculated on results of adaptation to synthetic disturbances. Add-on for post-hoc calibration of predictive uncertainty can be tuned on in-distribution and out-of-distribution data. Calibrated confidence values at the output of the AI system make it possible to discard a part of unabsorbed disturbances to mitigate their impact. It has also been experimentally confirmed that the Test-Time Adaptation stage allows to increase the robustness to various small changes in input data or weights.

The experiments were performed on the CIFAR-10 and CIFAR-100 datasets. The use of post-hoc resilience optimization increased robustness to fault injection by 5% and robustness to adversarial attack by 7%. Moreover, tuning on 10% of the test data increased robustness to fault injection by 6% and robustness to adversarial attack by 7.1%. In addition, the use of post-hoc resilience optimization increased the integral indicator of resilience to task changes by 10.5%. Post-hoc uncertainty calibration increases the robustness to fault injection models by an average of 4.4% and the robustness to adversarial attacks by an average of 1.3%. The use of Test-Time Adaptation additionally increases robustness to Fault Injection by 6.9% and robustness to Adversarial Attack by 4.72%. Even an increase in the intensity of attacks does not lead to a noticeable decrease in accuracy.

Thus, experimentally confirmed increase of robustness and speed of adaptation for image recognition system during several intervals of the system's life cycle due to the use of resilience optimization, uncertainty calibration and Test-Time Adaptation stages.

7.2. Limitations

This research is illustrated through a case study of an image classification system and does not detail the application of resilience-aware MLOps to self-supervised or reinforcement learning systems. However, the overarching structure of resilience-aware MLOps is applicable to all kinds of intelligent systems. Additional limitation may be associated with attempts to generalize the information found, which could influence the completeness of the literature review.

The Graceful Degradation stage is excluded from the detailed analysis of their impact on resilience. The article focuses on the analysis of the MLOp structure with regard to resilience, as well as the peculiarities of implementing the stages of resilience optimization, calibration of predictive uncertainty, and test-time adaptation.

The specifics of implementing each phase on specific IoT, Edge, and other resource-constrained platforms were not considered. The focus was not on the type of target platform, but on identifying new stages of MLOps aimed at ensuring resilience.

7.3. Future research work

Future research should focus on the development new flexible adapter and meta-adapter architectures as addons for AI system resilience. Special attention should also be paid to the question of providing resilience for self-supervised and reinforcement learning systems. Another important area of research should be the investigation of methods to ensure resilience to new types of attacks on AI systems.

Acknowledgements

The research was concluded in the Intellectual Systems Laboratory of Computer Science Department at Sumy State University with the financial support of the Ministry of Education and Science of Ukraine in the framework of state budget scientific and research work of DR No. 0124U000548 "Information Technology for Ensuring the Resilience of the Intelligent Onboard System of Small-Sized Aircraft".

References

- [1] F. Khalid, M. A. Hanif, M. Shafique, Exploiting Vulnerabilities in Deep Neural Networks: Adversarial and Fault-Injection Attacks, in: Proceedings of the Fifth International Conference on Cyber-Technologies and Cyber-Systems, 2020, pp. 24–29. <http://hdl.handle.net/20.500.12708/55602>.
- [2] V. Moskalenko, V. Kharchenko, Resilience-aware MLOps for AI-based medical diagnostic system, *Frontiers in Public Health*, 12 (2024). doi: 10.3389/fpubh.2024.1342937.
- [3] M. Testi, M. Ballabio, E. Frontoni, G. Iannello, S. Moccia, P. Soda, G. Vessio, MLOps: A Taxonomy and a Methodology, *IEEE Access*, 10 (2022) pp. 63606–63618. doi:10.1109/access.2022.3181730.
- [4] F.O. Olowononi, D.B. Rawat, C. Liu, Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS, *IEEE Communications Surveys & Tutorials*, 23 (1) (2021) pp. 524–552. doi:10.1109/comst.2020.3036778..
- [5] J. V. Duddu, N. Rajesh Pillai, D.V. Rao, V.E. Balas, Fault tolerance of neural networks in adversarial settings, *Journal of Intelligent & Fuzzy Systems*, 38 (5) (2020) pp. 5897–5907. doi:10.3233/jifs-179677.
- [6] S. Yang, T. Fevens, Uncertainty Quantification and Estimation in Medical Image Classification, in: *Lecture Notes in Computer Science*, 2021, pp. 671–683. doi:10.1007/978-3-030-86365-4_5.
- [7] X. Zhang, J. Jaskolka, Conceptualizing the Secure Machine Learning Operations (SecMLOps) Paradigm, in: 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS), IEEE, 2022. doi:10.1109/qrs57517.2022.00023
- [8] V. J. Reddi, A. Eilum, S. Hymel, D. Tischler, D. Situnayake, C. Ward, L. Moreau, J. Plunkett, M. Kelcey, M. Baaijens, et al. Edge impulse: An MLOps platform for tiny machine learning, in: *Proceedings of Machine Learning and Systems*, 2023. doi:10.48550/arXiv.2212.03332.
- [9] S. Leroux, P. Simoens, M. Lootus, K. Thakore, & A. Sharma. TinyMLOps: Operational Challenges for Widespread Edge AI Adoption, in: 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2022. doi:10.1109/ipdpsw55747.2022.00160
- [10] V.V. Moskalenko, Model-agnostic meta-learning for resilience optimization of artificial intelligence system, *Radio Electronics, Computer Science, Control*, (2) (2023), 79. <https://doi.org/10.15588/1607-3274-2023-2-9>.
- [11] W. Hou, Y. Wang, S. Gao, T. Shinozaki, Meta-Adapter: Efficient Cross-Lingual Adaptation With Meta-Learning, in: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021. doi:10.1109/icassp39728.2021.9414959.
- [12] K. Bhardwaj, J. Diffenderfer, B. Kailkhura, M. Gokhale, Benchmarking Test-Time Unsupervised Deep Neural Network Adaptation on Edge Devices, in: 2022 IEEE International

- Symposium on Performance Analysis of Systems and Software (ISPASS), IEEE, 2022. doi:10.1109/ispas55109.2022.00033.
- [13] M. Zhang, S. Levine, C. Finn, MEMO: Test Time Robustness via Adaptation and Augmentation, arXiv, Version 3, 2021. <https://doi.org/10.48550/ARXIV.2110.09506>.
 - [14] X. Yang, J. Xu. Few-shot Classification with First-order Task Agnostic Meta-learning, in: 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA). doi:10.1109/cvidliccea56201.2022.9824307.
 - [15] Wang R., Xu K., Liu S., Chen Pin-Yu et al. On Fast Adversarial Robustness Adaptation in Model-Agnostic Meta-Learning, in: 9th International Conference on Learning Representations, ICLR 2021. doi:10.48550/arXiv.2102.10454.
 - [16] M.W. Shen, Trust in AI: Interpretability is not necessary or sufficient, while black-box interaction is necessary and sufficient, arXiv, 2022. DOI:10.48550/ARXIV.2202.05302.
 - [17] X. Song, Y. Yang, K. Choromanski, K. Caluwaerts, W. Gao, C. Finn, J. Tan, Rapidly Adaptable Legged Robots via Evolutionary Meta-Learning, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020. doi:10.1109/iros45743.2020.9341571.
 - [18] S. Kotyan, D.V. Vargas, Adversarial robustness assessment: Why in evaluation both L_0 and L_∞ attacks are necessary, PLOS ONE, 17 (4) (2022) e0265723. doi:10.1371/journal.pone.0265723.
 - [19] G. Li, K. Pattabiraman, N. DeBardeleben, TensorFI: A Configurable Fault Injector for TensorFlow Applications, in: 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2018. doi:10.1109/issrew.2018.00024.
 - [20] P. Peng, J. Lu, T. Xie, S. Tao, H. Wang, H. Zhang, Open-Set Fault Diagnosis via Supervised Contrastive Learning With Negative Out-of-Distribution Data Augmentation, IEEE Transactions on Industrial Informatics, 19 (3) (2023) pp. 2463–2473. doi:10.1109/tii.2022.3149935.
 - [21] D. Karimi, A. Gholipour, Improving Calibration and Out-of-Distribution Detection in Deep Models for Medical Image Segmentation, IEEE Trans. Artif. Intell. (2022) 1. doi:10.1109/tai.2022.3159510.
 - [22] R. Doon, T. Kumar Rawat, S. Gautam, Cifar-10 Classification using Deep Convolutional Neural Network, in: 2018 IEEE Punecon, IEEE, 2018. <https://doi.org/10.1109/punecon.2018.8745428>.
 - [23] Liu, H. Chen, W. Zhou, Improved MobileViT: A More Efficient Light-weight Convolution and Vision Transformer Hybrid Model, in: Journal of Physics: Conference Series, Vol. 2562, Issue 1, IOP Publishing, 2023, p. 012012. <https://doi.org/10.1088/1742-6596/2562/1/012012>.

The apriori method in the collection of significant values of pharmaceutical data

Nataliya Maslova¹, Olena Liubymenko¹, Olha Polovynka² and Yaroslav Dorogyy¹

¹ Donetsk National Technical University, str. Potebni, 56, Lutsk, 43018, Ukraine

² National Technical University "Kharkov Polytechnic Institute", ul. Kyrpychyova, 2, Kharkov, 61002, Ukraine 1

Abstract

The paper demonstrates the relevance and necessity of seeking efficient algorithms for large-scale data applications. A review of existing approaches and methods for analyzing unstructured data, often found in pharmaceutical data and medical research results, has been conducted, providing a description of the characteristics of such data. It is indicated that Data Mining methods, including the Apriori algorithm, are applicable for discovering hidden patterns in large volumes of data. Modern implementation tools of the Apriori algorithm for parallel and distributed processing on platforms such as MPI, OpenMP, Hadoop, OpenCL, Spark, and Apache Flink are discussed. The specifics of implementing the Apriori algorithm on each platform are described. Theoretical recommendations are provided, and the results of applying the platforms in terms of the complexity (required workload) and time to obtain results are forecasted. A computational experiment is conducted, which overall confirms the theoretical conclusions presented in the concluding sections of the paper and outlines avenues for further research, envisaging deeper experimentation with further application of parallel and distributed versions of association search algorithms in various fields, including medical and pharmacological domains.

Keywords

Apriori parallel algorithm, Big Data, pharmaceutical logistics

1. Introduction

The proliferation of sources and volumes of information, its diversity, and distributed storage have led to the necessity of reviewing not only data collection and storage technologies, but also methods of analysis. Data Mining allows for the generation of new ideas and understandings of the nature and interrelationships of objects or phenomena. Typical data analysis tasks include classification, clustering, association rule mining, anomaly detection, and others.

In this context, the Apriori method serves as an important tool for analyzing big data. The Apriori method has long been recognized as a valuable tool for intelligent data analysis, especially for discovering association rules in large datasets. However, the constantly increasing volume and complexity of data, as well as the emergence of new data processing platforms, pose significant challenges for traditional analysis methods. Apriori is used to discover association rules among data elements, enabling the identification of interesting relationships in large datasets that may not be visible using conventional analysis methods. This approach facilitates drawing conclusions and making decisions based on the analysis of large volumes of data with significant efficiency and reliability.

One of the main problems associated with the constant growth of data is the complexity of processing and analyzing large datasets due to the limited performance of traditional methods. Various methodological approaches exist to address this problem, including parallel computation, distributed data processing, and the use of specialized algorithms. These approaches enable the efficient distribution of tasks and optimization of data processing for large volumes.

The paper discusses approaches and methods for the analysis of unstructured data. It demonstrates how Data Mining methods and the Apriori algorithm can be used to uncover hidden relationships in large unstructured datasets, particularly in pharmaceutical logistics. A comparative analysis of the characteristics of Apriori algorithm variants used for association rule mining is conducted, and a

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ nataliia.maslova@donntu.edu.ua (N. Maslova); olena.liubymenko@donntu.edu.ua (O. Liubymenko); olga.polovynka1@gmail.com (O. Polovynka); yaroslav.dorohyi@donntu.edu.ua (Y. Dorogyy)

ORCID: 0000-0002-9078-0973 (N. Maslova); 0000-0002-5935-6891 (O. Liubymenko); 0000-0002-0575-4587 (O. Polovynka); 0000-0003-3848-9852 (Y. Dorogyy)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

modification of the Apriori algorithm for processing unstructured distributed data using parallel methods is proposed. Results of application on various platforms, both existing and proposed variants of association rule mining algorithms, are presented.

In this context, the scientific novelty of the research lies in organizing the analysis of information platforms, with an emphasis on parallel and distributed processing of unstructured data, where the implementation of deep analysis algorithms, including the Apriori algorithm, will be most efficient in terms of speed and technological implementation during the processing of medical and pharmaceutical data.

1.1. Problems of Analyzing Large Volumes of Data, the Purpose of the Study

The concept of «Big Data» encompasses datasets characterized by large volumes, high accumulation rates, and diversity, known as volume, velocity, and variety.

Analytical publications on big data processing [1-3] reveal key trends, contributions, and research developments in this field. Processing large volumes of data requires powerful computational resources, efficient algorithms to ensure analysis speed and effectiveness, and appropriate information storage and processing systems. There is a significant number of algorithms and methods for intelligent (deep) analysis, but each task proves to be unique, so research on developing methods for solving big data processing tasks continues to be relevant.

Special importance lies in algorithms for processing unstructured distributed data, which accumulate in various fields such as finance, logistics, telecommunications, medicine, etc. [4]. Data analysis in these areas needs to be conducted in real-time, placing additional demands on processing speed and system responsiveness. Additionally, increasing data volume may raise risks of privacy and security breaches, hence organizing an adequate level of data protection is crucial.

Neural networks, decision trees, fuzzy models, regression, and clustering analysis methods are widely used for big data analysis. However, these methods are typically used for processing structured data. Processing unstructured data, such as logistic data from pharmaceutical networks or medical research data related to drug selection and interaction analysis, is not possible with similar methods [5].

The choice of processing methods and algorithms determines the research results and effectiveness. The most applicable methods for processing both structured and unstructured large-scale data are Data Mining methods.

Research Object:

Processes of intelligent analysis of unstructured large-scale data.

Research Subject:

Algorithms and methods of intelligent data analysis in solving tasks of discovering hidden patterns.

The research objective is to select platforms for the effective implementation of deep analysis algorithms (Apriori algorithm), considering the possibility of parallel and distributed data processing for large-scale data. Pharmaceutical data applicable in medical research and pharmaceutical logistics data are chosen as input data for conducting experiments.

2. Apriori Method as a Tool for Big Data Analysis

To process large arrays of unstructured data and address the mentioned tasks, it is advisable to use methods for mining associative rules. Based on identified dependencies, rule bases understandable to experts in applied fields can be synthesized.

The Apriori method in Data Mining is an algorithm used for extracting associative rules from a database.

The main goal of the Apriori method is to identify relationships between different elements in sets of structured and unstructured data.

The main Apriori algorithm has several stages.

Candidate generation: Initial itemsets are formed with one item. On the second iteration, two-item sets are analyzed. On the third iteration, three-item sets, and so on, until the algorithm provides informative results (rules are formed).

Trimming non-informative values in the Apriori method is done by determining significance indicators, which may vary depending on the research area and the required reliability of the results. These indicators are support and confidence.

Support measures how often a set of items appears in the dataset.

Candidate filtering: Itemsets that do not meet certain support criteria are removed.

Association rule creation: The remaining itemsets are used to create association rules describing relationships between elements.

A detailed description of the basic algorithm can be found, for example, in [6].

Among the main drawbacks of this algorithm, the authors note the need for repeated scanning of the original dataset and the complexity of determining threshold values for each indicator. Therefore, one of the main directions for improving the efficiency of the Apriori algorithm is the possibility of parallelizing the processing of large datasets, which can significantly reduce scanning time.

The Apriori method is widely used in areas such as recommendation systems, purchase analysis (including pharmaceuticals), advertising strategies, education [7], and other fields where the discovery of relationships between different elements in datasets is important. In medical modeling, the Apriori method is used for automated analysis of medical data in the process of providing medical care, in researching the interaction of drugs taken by the patient [8-10]. Modern data analysis methods are used to examine the patient's basic information, medical history, physical condition, and other important textual characteristics. Additionally, correlations are sought between the factors that caused the illness, its severity degree with the results of the patient's examination. This is done to achieve automatic medical clinical care and establish an accurate diagnosis. In this work, the method is applied to pharmaceutical logistics tasks, critical factors in the implementation of management methods after the end of the life cycle in the pharmaceutical care process described in [11].

We will not dwell on the description and analysis of basic and modified algorithms for mining associative rules. They are sufficiently detailed in the literature and works of the authors [11]. We will provide a brief overview of the application area emphasized in the article preview and proceed to the direct analysis of modern and most promising technologies and implementations in terms of speed and reliability.

3. Methodological Approaches to Addressing Big Data Processing Issues in Pharmaceuticals

Pharmaceutical logistics involves managing the movement of medical supplies and related financial, personnel, and information flows to effectively distribute and reduce overall costs in the supply, production, and sale of pharmaceuticals to achieve the required quality and meet consumer demands.

In the context of pharmaceutical logistics, big data may include information on production, supply, storage, and distribution of medical supplies, clinical trials, electronic medical records of patients, information on pharmaceuticals and their interactions, as well as data on patients' previous prescriptions and treatments.

The use of pharmacy network data is envisaged as input to obtain analytical dependencies necessary to solve pharmaceutical logistics tasks. They are characterized by large volumes of information, a distributed structure, and the availability of specialized basic data collection and processing packages.

Materials [12-14] describe the features of pharmaceutical logistics, list twenty main principles characterizing it, and confirm the complexity and multidimensionality of pharmaceutical data processing tasks. This data can be classified as Big Data - falling into the category of processing large volumes of structured and unstructured data.

In the article [15], the relevance and necessity of searching for effective Data Mining algorithms, including parallel ones, applied to large volumes of pharmaceutical data, are proven. Examples of existing processing packages are provided, their main capabilities and shortcomings are analyzed.

It is noted that the main problems in processing pharmaceutical data are:

Distribution: collecting information from different sources and the need to store it in a specialized repository;

Significant volume (productivity): analyzing information volumes measured in terabytes can take an unacceptable amount of time for analysis and require significant computational power.

Modern technologies supporting IDA algorithms and methods include:

- Data analysis software programs (such as STATISTICA, SAS, SPSS, STATGRAPHICS, etc.);
- Artificial neural networks (BrainMaker, NeuroShell, OWL);

- Case-based reasoning systems (tools), for example, KATE tools, Pattern Recognition Workbench;
- Decision Trees methods (KnowledgeSeeker, See5/C5.0, Clementine, SIPINA, IDIS, etc.);
- Evolutionary programming software tools (NeuroShell);
- Genetic algorithms (GeneHunter);
- Limited enumeration methods (used, for example, in the WizWhy system from WizSoft);
- Data visualization systems in multidimensional space (DataMiner 3D).

Despite this variety, [16] asserts that the main problems of processing most types of data can be solved through parallel (Parallel Data Mining) and/or distributed (Distributed Data Mining) Data Mining algorithms.

4. Tools for Implementing Parallel Versions of the Apriori Algorithm

The main modern instrumental tools for implementing the Apriori algorithm for parallel and distributed processing are platforms listed below.

MPI (Message Passing Interface, 1990): This is a standard for working with parallel programming in large computing clusters. Additionally, the mentioned platform MPI4py is a Python wrapper for the MPI (Message Passing Interface) library and was developed in 2008. This library allows developers to utilize MPI capabilities for parallel programming in the Python language.

OpenMP (Open Multi-Processing, 1997): An open standard for shared-memory parallel programming that provides directives for parallelizing code at the process level.

Hadoop (2006): A framework for processing and analyzing large volumes of data in distributed computing environments.

OpenCL (Open Computing Language, 2008): An open standard for parallel programming on heterogeneous systems such as GPUs, CPUs, and others.

Spark (2010): A framework for parallel data processing working in memory, enabling a wide range of data analysis tasks.

Apache Flink (2014): A distributed system for processing streaming data and batch operations, providing high-speed data processing.

The platforms are arranged by the years of active market entry and allow programmers to effectively utilize parallel computing resources for fast and efficient resolution of various tasks.

Let's consider the peculiarities of implementing the Apriori algorithm on some of the listed platforms.

4.1. MPI platform

The implementation of the Apriori algorithm in a specified environment is most thoroughly described in the work [17]. It discusses modern methods for mining associative rules and discovering knowledge in large volumes of data. Based on a conducted comparative analysis and taking into account the specificity of the subject area, the use of the Apriori algorithm is justified, for which criteria and methods of parallelization during the implementation of the associative rule search procedure are considered.

The results obtained from the research were tested in the form of a parallel implementation of the QuickSort algorithm.

As the hardware base, a parallel cluster of the university with the characteristics listed in Table 1 was used.

Table 1
Technical characteristics of the cluster

Fat servers	Thin servers	Network devices
- Dell Power Edge R710;	- Intel® Xeon®;	- Dell Power Connect 5448;
- 16 cores;	CPU X5560@2.8 GHz	- InfiniBand Voltaire
- 24 GB RAM;	- 16 cores;	Grid Director
- HDD: 2 x 140 GB	- 16 GB RAM;	
- 6 x 1000 GB	- HDD – no;	

- 2 IB ports;
 - 4 LAN ports
 - 1 IB ports;
 - 2 LAN ports;
 - 1 IPMI port.
-

The general scheme of the cluster is shown in Figure 1.

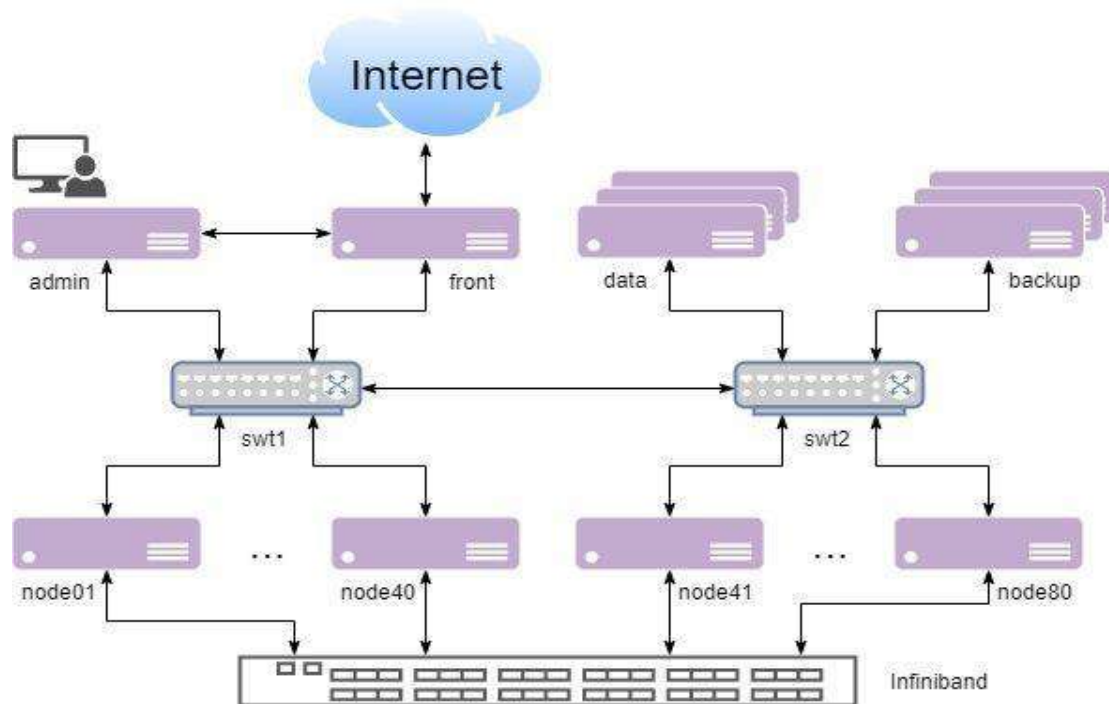


Figure 1: Cluster diagram

The MPI implementation allows for distributing data among cluster nodes to perform separate processing tasks on each. MPI does not provide specific tools for distributing processes. It's the developer's responsibility to handle tasks such as data distribution, processing, execution, and final result collection. Considering these capabilities and aiming to expedite result acquisition, the algorithm was divided into functions executed sequentially:

$$F = F1 + F2 + F3 + F4 + F5 + F6, \quad (1)$$

where the first function (F1) - is reading part of the data from the file;

the second function (F2) - distributes data between processes;

function F3 - parallel scanning and calculation of "itemsets" - a set of elements that are often found in this part of the sample;

function F4 - for data exchange between nodes (initiating data transmission operations between nodes);

function F5 - for result aggregation: Each node receives partial results from other nodes and combines them to create a common list of frequently occurring itemsets.

F6 - function implementation of the main part of the Apriori algorithm (main function), which controls the processes of data reading, distribution among processes, exchange, and initiation of each process and generation of new candidates.

Steps 2-6 are repeated until reaching the maximum (specified) value of itemsets or another stopping condition. After processing is completed, each node aggregates the results and analyzes them to discover associative rules. Calculations were performed with a fixed support count (20%).

In conclusion, it can be summarized that the implementation of the Apriori algorithm using MPI is possible, although algorithmically it proved to be multi-stage. It is also worth noting that despite the computations being performed quite fast, the data reading and distribution block among processes required significant time.

4.2. OpenMP Platform

OpenMP (Open Multi-Processing) is designed for parallel programming on multi-core architectures. It allows programs to be parallelized by adding compiler directives to regular sequential code. Implementation features of the Apriori algorithm for OpenMP include aspects such as memory-level parallelism, distributed work, mechanisms for synchronization of access to shared data between threads, memory allocation, performance optimizations, testing, and validation.

Memory-level parallelism means that OpenMP can simultaneously process data on multiple processor cores. Distributing work among different threads speeds up processing time because different parts of the data can be processed simultaneously.

OpenMP provides mechanisms for synchronizing access to shared data between threads, which is important for preventing conflicts and race conditions during parallel computation.

The implementation of Apriori for OpenMP requires performance optimization by using optimal parameters and selecting the best algorithms for the specific task.

After implementing the algorithm, testing and validation were conducted on real data to ensure the correctness and efficiency of the algorithm.

OpenMP is a standard for parallel programming on multi-core and multi-processor systems with shared memory. It provides a set of compiler directives and a library of functions that allow programs to be parallelized at the directive level. For comparison with MPI results, C and C++ programming languages were applied during the programming stage.

Table 2 shows the algorithm execution time for parallelizing Apriori on a quad-core processor using OpenMP with a fixed support count (20%) and the same algorithm implemented with MPI technology.

Table 2
Algorithm Execution Time for OpenMP and MPI

Array size	MPI 1	OpenMP 1	MPI 2	OpenMP 2	MPI 4	OpenMP 4
1000000	0.16	0.17	0.13	0.13	1.53	0.32
5000000	1.22	0.51	1.11	0.46	1.63	0.43
10000000	3.564	0.95	3.17	0.86	1.51	0.64
15000000	7.008	1.17	4.40	0.98	1.72	0.41
20000000	11.536	1.31	6.71	1.23	1.63	0.72
40000000	40.571	25.23	27.25	23.15	11.88	5.21
80000000	151.998	27.32	81.49	27.64	21.6	5.34

As demonstrated by the experiment, implementing Apriori for OpenMP can significantly improve data processing speed by leveraging parallel programming and optimizing algorithm performance across multiple processor cores.

4.3. Hadoop Platform.

The platform description is provided in [18], [19], [20]. For instance, [19] examines the most well-known modifications of Apriori algorithms for association rule mining. The research presents the performance results of linear algorithms Apriori and Apriori-TID on datasets of various sizes using computers with standard memory capacities. The feasibility of utilizing the AIS algorithm and its parallel implementation on the Hadoop MapReduce framework is justified. A comparison between the linear implementation of the Apriori-TID algorithm and the parallel implementation of the AIS algorithm is provided. Conclusions are drawn regarding the effectiveness of employing parallel algorithms to address the formulated task.

In general, developing an Apriori program for Hadoop on 12 processors requires the utilization of the Hadoop MapReduce framework and distributed computing. This can be achieved by leveraging the following capabilities of Hadoop and its compatible tools (Table 3).

Table 3
Stages of implementation of the Apriori algorithm on Hadoop

No	Stage	Processing tool	Stage description
1	Distribution of data	Hadoop Distributed File System (HDFS)	the data should be distributed among the Hadoop cluster nodes
2	Fragment processing	Mapper	In Mapper, you need to implement logic for selecting subsets of items from each fragment
3	aggregation of results	Reducer	Reducer combines and processes the results obtained from different pieces of data.
4	Apriori	Apriori Process	finding frequently occurring items and generating candidates for the next level. Each iteration is performed on a separate subset of the data
5	Processing	Parallel Processing	parallel processing of different data fragments at the same time on different nodes of the cluster.
6	Load control	Resource Management	Monitoring the correct use of cluster resources to avoid excessive load on individual nodes.

Thus, it can be concluded that implementing the Apriori algorithm for Hadoop is a rather complex process that requires a deep understanding of the Hadoop platform itself, the operation of associated tools (MapReduce, Apache Spark), and their application in software implementation.

Another aspect to consider is that Hadoop necessitates programming in Java for MapReduce task implementation. For other programming languages, such as C++, one needs to utilize the Hadoop Streaming framework, which does not allow for a direct comparison of results obtained for the Hadoop platform with those from experiments on MPI and OpenMP.

Additionally, input files must be prepared in a format understandable by Hadoop, and to obtain results, an output directory must be created.

4.4. OpenCL Platform

The implementation of the Apriori algorithm for OpenCL utilizes the open standard OpenCL for parallel programming on heterogeneous computing devices such as central processing units (CPUs), graphics processing units (GPUs), and others.

The uniqueness of the Apriori implementation for OpenCL lies in harnessing the parallelism capabilities provided by OpenCL to process large volumes of data. The algorithm is divided into parts that can be computed in parallel and processed on different computing devices simultaneously.

With OpenCL, various computing device resources can be efficiently utilized, allowing for computation distribution between CPUs and GPUs based on their capabilities and tasks. This enables significant improvements in data processing speed and algorithm performance for Apriori.

The general process of implementing Apriori for OpenCL includes the following five steps:

Development of an OpenCL kernel for computing subsets of itemsets and analyzing their frequency.

Implementation of mechanisms for data partitioning and distribution among different computing devices.

Configuration and optimization of computing system parameters for maximum performance.

Testing and validation of the implemented algorithm on real datasets.

4.5. Apache Spark platform

Apache Spark is a distributed data processing system that provides capabilities for parallel processing of large volumes of data. One of its features is the use of the Scala language for algorithm implementation.

For data processing, candidate itemsets generation, Apriori algorithm iterations, and result analysis using Spark, functions should be applied, some of which are listed in Table 4. The availability of functions is a distinctive feature of Spark.

Table 4**Basic Apache Spark functions to implement the Apriory method**

	Functions	Action
Data processing functions		
1	map(func)	Enables you to perform data transformation, processing, and cleaning operations in a distributed Spark environment.
2	flatMap(func)	as a result of the operation of the flatMap method, each element can generate zero or more elements in the original RDD.
Functions for building itemsets candidates		
1	Cartesi (anotherRDD)	Returns an RDD that contains all possible pairs (combinations) of elements from the current RDD and another RDD.
2	mapPartitions(func)	Applies the given function func to each section of the RDD and returns a new RDD with the results.
Apriori functions (iterations)		
1	map	Data processing and filtering in RDD
2	flatMap	
3	filter and others	
Analysis of results		
1	reduceByKey(func)	Combines the values for each key in the RDD using the given function func.
2	collect	Gathers all elements from the RDD into a node and returns an array.

4.6. Apache Flink Platform

The features of implementing the Apriori algorithm for Apache Flink include the following aspects:

- Stream processing: Apache Flink supports stream data processing, allowing data to be processed in real-time. This can be particularly useful for analyzing large data streams, such as in large networks or IoT systems.
- Distributed processing: Apache Flink provides distributed data processing capabilities, enabling the utilization of server clusters for fast processing of large volumes of data.
- Integration with other tools: Apache Flink easily integrates with other Apache tools such as Apache Hadoop, Apache Kafka, Apache Hive, and others. This facilitates seamless data exchange between different systems and leverages additional functionalities.
- Programming simplicity: Apache Flink offers a convenient programming interface for implementing complex analytical tasks on large datasets. It supports programming languages such as Java, Scala, and Python, making program development easier for a wide range of developers.
- Performance optimization: Apache Flink incorporates built-in mechanisms for performance optimization, including query optimization and execution plan scheduling. This maximizes cluster resource utilization and reduces data processing time.
- Scalability: Apache Flink scales easily both vertically and horizontally, allowing for the processing of both small and large volumes of data. It supports horizontal scaling, enabling the addition of new nodes to the cluster to enhance its performance.

Despite its many advantages, Apache Flink also has its drawbacks. These include configuration complexity, the prevalence of Apache Spark, and significant resource requirements, which can pose challenges for scalability and performance when processing large datasets. Additionally, the current stage of incomplete and inadequate documentation and debugging issues does not recommend Apache Flink for roles other than research-oriented ones.

5. Recommendations and Results

The execution speed of the Apriori algorithm for the same data volume and with a fixed support count can significantly depend on various factors, such as the characteristics of the program itself, data peculiarities, and the hardware on which it runs. Therefore, there is no definitive answer to the question of which algorithm is "absolutely" fast.

However, some general guidelines can be provided:

OpenCL Technologies: These can provide a speed boost in execution through parallel computations on a large number of GPU cores.

MPI and MPI4py: These can work very efficiently on clusters with a large number of nodes. However, MPI requires complex development and management of communications between cluster nodes.

OpenMP: It provides a simple way to parallelize code for multi-processor systems with multicore processors. It can be effective for execution on a single node with multiple processors.

Hadoop and Spark: These are designed to work with large data volumes on server clusters. They provide distributed computing capabilities but usually demonstrate lower speed compared to optimized solutions for parallel computing on GPUs or multicore systems.

Apache Flink: It is a distributed data processing system for streaming data and batch data. It can be an effective option for real-time data processing or streaming data.

In addition, performance can be improved by optimizing the code, using specialized libraries and frameworks, and adjusting system parameters for maximum productivity. It is also important to consider the specific task and project requirements when choosing a parallel programming platform.

Thus, an analysis of the feasibility of using different platforms for association rule mining algorithms has been conducted, and recommendations have been provided. The results of the descriptive comparison can assist researchers in selecting a data processing algorithm for large volumes of data using the Apriori algorithm or in solving other data mining tasks.

In addition to general recommendations, the authors conducted a series of experiments.

To conduct the experiments, several datasets of different sizes were formed. All datasets were stored in CSV format files. This format is lightweight, contains no metadata, and is easily readable, which is crucial when reading large volumes of data.

The experiments were conducted under identical conditions using a multicore computer with variations on 2, 4, and 8 cores.

For comparison, the parallel processing and parallelization algorithm described in the authors' works were applied, as well as implementations of data analysis algorithms from popular open-source data analysis libraries, namely RapidMiner and WEKA, and with the Apache Spark MLlib MapReduce-based parallel computing system.

Weka is a free library and software for machine learning and data analysis developed at the University of Waikato in New Zealand. Weka provides a wide range of machine learning algorithms for classification, clustering, regression, association rules, visualization, and other data analysis tasks. The Apriori algorithm in Weka is available through the user interface and through the programming interface. Weka is primarily designed for use on a single server or workstation and does not have built-in support for distributed or parallel computing using GPUs, MPI, or Hadoop. However, in combination with other tools and technologies, the Weka library can be used as a node in a general computation algorithm.

RapidMiner is a powerful data analysis platform for machine learning, model development, and visualization that allows users to effectively analyze and use data for decision making. RapidMiner can effectively work with data of various volumes, processing both small data volumes (e.g., a few megabytes or gigabytes) and large data volumes (several gigabytes or terabytes). The Apriori algorithm in RapidMiner is available as one of the modules for data analysis. Although RapidMiner is not directly associated with MPI (Message Passing Interface) or Hadoop, it can be integrated with them for processing large data volumes and distributed computing.

Apache Spark MLlib (Machine Learning Library) is a machine learning library that is part of Apache Spark, an open-source framework for parallel and distributed data processing. MLlib provides a wide range of machine learning algorithms and data analysis algorithms that can be used to solve various tasks, such as classification, regression, clustering, recommendations, image processing, and more. These algorithms are optimized for execution on distributed clusters using Apache Spark, allowing

large volumes of data to be processed quickly and efficiently. The Apriori algorithm is not included in the standard Apache Spark but is implemented using the Python programming language.

Apache Spark is designed for efficient processing of large volumes of data in distributed environments. It is integrated with Hadoop and can use it as the primary data store and platform for distributed computing, although its library can also be used separately from Hadoop. At the core of this integration is a common distributed file system mechanism (HDFS - Hadoop Distributed File System) that allows Apache Spark to work with data stored in Hadoop.

The results of the experiments for various data analysis algorithms are presented in Table 5.

Table 5
Execution Time of Association Rule Mining Algorithms

Sample a Cores	Apache Spark MLLib				DXelopes				Rapid Miner	Weka
	1	2	4	8	1	2	4	8	1	1
1000000	0,156	0,149	0,119	0,096	0,108	0,092	0,066	0,051	0,1144	0,0663
2000000	0,295	0,280	0,192	0,111	0,201	0,135	0,084	0,074	0,2145	0,1248
3000000	0,454	0,376	0,241	0,133	0,278	0,168	0,110	0,092	0,3289	0,1911
4000000	0,565	0,413	0,301	0,174	0,370	0,211	0,127	0,113	0,4225	0,247
5000000	0,618	0,485	0,350	0,237	0,415	0,301	0,190	0,131	0,5161	0,3016
6000000	0,823	0,686	0,397	0,299	0,548	0,319	0,176	0,143	0,5967	0,3484
7000000	0,935	0,761	0,475	0,360	0,632	0,344	0,194	0,155	0,6773	0,3952
8000000	1,119	0,913	0,585	0,417	0,706	0,403	0,219	0,170	0,8112	0,4732
9000000	1,240	0,999	0,692	0,464	0,800	0,446	0,235	0,198	0,8983	0,5239
10000000	1,445	1,089	0,778	0,522	0,849	0,595	0,370	0,239	1,0192	0,5954

Graphical Interpretation of Results (calculation time for the Apriori algorithm on different platforms and data volumes) is provided in Figure 2.

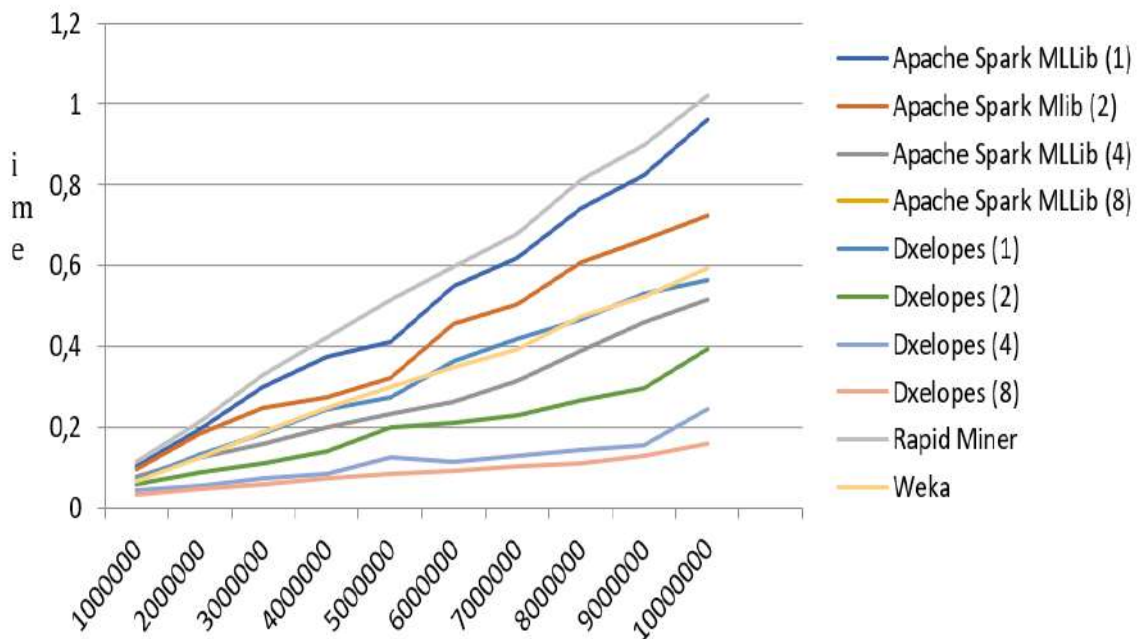


Figure2: Graphical Interpretation of Results

Thus, for algorithms and platforms without GPUs, the experiment revealed that the most efficient algorithm was the parallel implementation using the Weka platform, but it does not provide parallel processing capabilities. Additionally, its execution was preceded by significant preparatory work in data preparation for calculations. Therefore, the Weka library at this stage may be recommended for linear implementations. Slightly lower performance results were provided by the Apache Spark MLLib library, but the further prospect is the full utilization of Apache Spark's capabilities and its integration with Hadoop.

It should be noted that during the experiments, a self-developed algorithm (QuickSort algorithms) and the DXelopes algorithm (Kharkov Polytechnic Institute) were applied. Therefore, the further perspective of the research may be the integration of machine learning methods with the Apriori algorithm. For example, by applying the Apriori method, frequent item sets will be identified in the dataset and used to detect associations between different elements. Machine learning methods will allow modeling complex dependencies and predicting results based on input data and classification, regression models, as well as elements of cluster analysis for various prediction or classification tasks.

6. Conclusions

As a result of the study, findings have been obtained that both generalize and advance the previous research conducted by the authors. The focus of the work has been on selecting platforms for the efficient implementation of deep analysis algorithms (specifically the Apriori algorithm), considering the possibility of parallel and distributed data processing of large volumes.

Pharmaceutical data applicable to medical research and pharmaceutical logistics data were chosen as input data for the experiments. The characteristics of such data, notably their distributed nature and substantial volume, were highlighted. The following actions were undertaken in the study:

The concept of "big data" and the challenges associated with their processing were examined.

The necessity of applying modern algorithms to analyze unstructured data, such as the results of medical research and pharmaceutical data, including pharmaceutical logistics data, was confirmed.

The Apriori method was discussed as a tool for analyzing data of significant volumes, and the study applied the author's QuickSort program.

The instrumental tools for implementing the Apriori algorithm for parallel and distributed processing (MPI, OpenMP, Hadoop, OpenCL, Spark, and Apache Flink platforms) were analyzed.

The peculiarities of applying these platforms in constructing algorithms for association rule mining were described, recommendations were provided, and the results of applying each of the mentioned platforms were forecasted in terms of complexity (required workload) and time to obtain results.

Computational experiments were conducted for cases of using linear and four multi-processor platforms in parallel implementations, confirming the analytical conclusions presented in the final sections of the work.

Paths for further research were outlined, envisioning a deeper experiment in the future application of parallel and distributed versions of association rule mining algorithms in various fields, including the medical-pharmacological sphere, with further integration of the Apriori algorithm with machine learning methods.

It's worth noting that the scientific novelty of the research lies in the conducted and integrated analysis of the characteristics of various information platforms designed for parallel and distributed processing of unstructured data. It has been demonstrated which of these platforms the implementation of deep analysis algorithms, including the Apriori algorithm, will be most efficient in terms of speed and technological feasibility in the processing

The Apriori method has long been recognized as a valuable tool in data mining, particularly for uncovering associative rules in large datasets. However, the ever-growing volume and complexity of data pose significant challenges to traditional methods of analysis. In this context, the scientific novelty of this study lies in the exploration of effective methods and algorithms for implementing the Apriori algorithm specifically tailored to handle large volumes of pharmaceutical data, with a focus on parallel and distributed processing.

The study delves into the methodological approaches necessary for processing unstructured data in pharmaceutical logistics, highlighting the importance of finding efficient Data Mining algorithms, particularly those capable of parallel and distributed processing. This aligns with the broader challenges faced in big data processing across various sectors, emphasizing the need for specialized approaches to handle the unique characteristics of pharmaceutical data.

The research systematically evaluates various platforms for implementing the Apriori algorithm, considering factors such as parallelism, memory-level processing, performance optimization, and scalability. This comprehensive analysis extends beyond theoretical considerations to practical implementations, providing insights into the feasibility and effectiveness of each platform in real-world scenarios.

Furthermore, the study conducts experiments to empirically validate the performance of different implementations of the Apriori algorithm on various platforms and datasets. By comparing results across different platforms and data volumes, the research offers valuable insights into the relative strengths and limitations of each approach, guiding researchers and practitioners in selecting the most suitable methods for their specific requirements.

Overall, the study contributes to the advancement of data mining techniques in pharmaceutical logistics by offering novel insights into the implementation of the Apriori algorithm for processing large volumes of data. By addressing the challenges posed by big data in this domain and proposing practical solutions, the research enhances our understanding of how Data Mining methods can be effectively applied to extract valuable insights and improve decision-making processes in pharmaceutical logistics.

Acknowledgements

Thanks to Professor O. Dmitrieva for guiding the first stages of the work and scientific contribution to research.

References

- [1] M. M. Medeiros, N. C. Hoppen and A. G. Maçad, Data science for business: benefits, challenges and opportunities. The Bottom Line: Managing Library Finances, ahead-of-print. (2020). DOI: 10.1108/BL-12-2019-0132
- [2] M.Juliardi, I. A Malik, Bibliometric Analysis of Data Science: Trends, Contributions, and Research Developments. West Science Interdisciplinary Studies, 1(07) (2023) 365-375. DOI:10.58812/wsis.v1i07.81.
- [3] L. P.Favero, P. Belfiore, Data Science for Business and Decision Making. Elsevier Science. (2019). DOI <https://doi.org/10.1016/C2016-0-01101-4>
- [4] I. Dorohyi. Critical IT Infrastructure Resource Distribution Algorithm. Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, 2 (2021): 632–639.
- [5] J.Luengo, D.García-Gil, S.Ramírez-Gallego, S.García, F. Herrera, Big Data Preprocessing: Enabling Smart Data. Springer International Publishing. (2021). ISBN 978-3-030-39107-2.
- [6] Akobir Shahida, Apriori is a scalable algorithm for searching for associative rules, 2020. <https://loginom.ru/blog/apriori>
- [7] N.Maslova, O. Polovynka Associative Methods as a Tool to Improve the Quality of Knowledge Control. Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems {(COLINS} 2020). Volume {I:} Main Conference. Lviv, 2020. pp. 778-787. <https://dblp.uni-trier.de/rec/bibtex/conf/colins/MaslovaP20>
- [8] K.Gulzar, M. A.Memon, S. M.Mohsin,, S.Aslam, A Akber,, M. A.Nadeem, "An Efficient Healthcare Data Mining Approach Using Apriori Algorithm: A Case Study of Eye Disorders in Young Adults," Information 2023, 14(4), (2023). DOI: 10.3390/info14040203
- [9] X. Li, B.Tan, J. ZhengXu, J.X. Xiao, Y. Liu. The Intervention of Data Mining in the Allocation Efficiency of Multiple Intelligent Devices in Intelligent Pharmacy. Comput Intell Neurosci, (2022). DOI: 10.1155/2022/5371575
- [10] M. L. Faquetti, A. M. D. L. Torre, Th.Burkard, G. Obozinski, Andrea M. Burden, Identification of polypharmacy patterns in new-users of metformin using the Apriori algorithm: A novel framework for investigating concomitant drug utilization through association rule mining, Pharmacoepidemiology and Drug Safety, 2022, Vol. 32, Issue 3, pp. 366-381.
- [11] O.L Polovynka, D.V. Nikulin, O.A. Dmitrieva, Study of the speed of operation of a priori algorithms on large volumes of data. Science and production. Series: Information technologies, Mariupol, 2020, 22, pp. 246-253. <http://sap.pstu.edu/article/view/211383>,
- [12] E. A. R. de Campos, I. C. de Paula, C. Sc. ten Caten, K. P. Tsagarakis, J. L.D.Ribeiro, Logistics performance: critical factors in the implementation of end of life management practices in the pharmaceutical care process, Environmental Science and Pollution Research, 2023, Vol. 30, pp. 29206–29228. <https://doi.org/10.1007/s11356-022-24035-z>

- [13] B. Sarwar, Supply chain management in the pharmaceutical industry (2020). <https://enhelion.com/blogs/2020/12/30/supply-chain-management-in-pharmaceutical-industry/>
- [14] O. O. Molodid, V. V. Shemena, General principles of organizing logistics business processes in a pharmaceutical company, Electronic journal "Effective economy", №6, 2019. doi: 10.32702/2307-2105-2019.6.62,
- [15] H.Jahani, , R.Jain, D.Ivanov. Data science and big data analytics: a systematic review of methodologies used in the supply chain and logistics research. Ann Oper Res (2023). <https://doi.org/10.1007/s10479-023-05390-7>
- [16] A. Koschmider, M. Aleknonytė-Resch, Fr. Fonger, Christian Imenkamp, Ar. Lepsien, K. Apaydin, M. Harms, D. Janssen, D. Langhammer, T.Ziolkowski, Y.Zisgen, Process Mining for Unstructured Data: Challenges and Research Directions Cornell University, (30 Nov 2023), 2023. <https://doi.org/10.48550/arXiv.2401.13677>
- [17] O.A. Dmitrieva, O.L. Polovynka, Algorithmic support for parallel methods of searching for associative rules, Collection of scientific works of the Donetsk National Technical University. Series: "Computer technology and automation", Pokrovsk, 2018, Is. 1 (31). pp. 62-69, https://science.donntu.edu.ua/ota-arhiv/31/025_dmitrieva.pdf
- [18] The Apache Software Foundation. Apache Hadoop, (28 July, 2020), 2020. URL: <https://blogs.apache.org/hadoop/entry/announce-apache-hadoop-3-3>
- [19] O.Polovynka, O. Dmitrieva, Research of the efficiency of the apriori group algorithms on different database sizes, SWorldJournal, Issue 6 / Part 1, p. 71-78, <https://www.sworldjournal.com/index.php/swj/article/view/swj06-01-012>
- [20] Yange, T. S., Gambo, I. P., Ikono, R., & Soriyan, H. A. A Multi-Nodal Implementation of Apriori Algorithm for Big Data Analytics using MapReduce Framework. International Journal of Applied Information Systems (IJ AIS), 2020, 12(31). DOI: 10.5120/ijais2020451868

Methodological Aspects of Studying the Accuracy of Computer Calculations in Applied Problems

Oleh Vietrov¹, Olha Trofymenko², Vira Trofymenko¹ and Maxim Hryhorov¹

¹ Vasyl' Stus Donetsk National University, 21, 600-richchia str., Vinnytsia, 21021, Ukraine

² University of Connecticut, 341 Mansfield Road, Storrs, CT 06269, USA

Abstract

The study is dedicated to considering some examples of computer calculations that are completely correct from the point of view of the standards of computer work with numbers and arithmetic operations, the result is incorrect to a human performer. It's demonstrated that mathematically correct calculations are often impossible to reproduce correctly within the IEEE-754 standard due to the fundamental limitations of the memory allocated to represent a specific number. The authors focused on the study of the specified problem using the example of the numpy module, as one of the main software tools for modern scientific calculations. The work has methodological value in the professional training of specialists in Computer Science.

Keywords

Computer calculations, IEEE-754 standard, numpy

1. Introduction

The problem of reliability of computer calculations is one of the fundamental problems of Computer Science [1], as it lies at the intersection of applied mathematics on the one hand, and physical and technical limitations of computer technology on the other. The avoiding the fundamentals of binary arithmetic and the principles of the standard for representing numbers in computer memory can significantly affect the correctness of both theoretical research (if computer technology was used for modeling) and simply the quality of the product when developing application programs.

The problems that can arise due to incorrect representation in computer memory are well known. Sometimes they are subtle problems that do not have a significant impact in a qualitative sense on a particular result. Therefore, in application activities, floating-point computing problems and features of IEEE-754 standard are often overlooked by software developers as well as academic researchers. However, there are notorious cases when careless handling of computer representation of numbers led to serious tragedies. A well-known example of the tragic consequences of such problems is "The Patriot Missile Failure" [2-3]. The main problem was that the time value in tenths of a second was multiplied by a factor of 0.1 to get a representation of time in seconds. This calculation was done using a 24-bit fixed-point register. The Patriot battery lasted about 100 hours and the accumulated error was about 0.34 seconds. One can also recall the precedent of "The Explosion of the Ariane 5" [4-5]. As a result of computer calculations, a 64-bit floating-point number was converted to a 16-bit signed integer. The number was greater than 32,768, and so the conversion failed. An error in the system for calculating the horizontal velocity of the rocket relative to the platform resulted in damage estimated at 500 million dollars. Some examples of real losses caused by rounding errors or problems in representing numbers in computer memory are given in [6].

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

o.vietrov@donnu.edu.ua (O. Vietrov); olha.trofymenko@uconn.edu (O. Trofymenko);

v.trofymenko@donnu.edu.ua (V. Trofymenko); hryhorov.m@donnu.edu.ua (M. Hryhorov)

0000-0002-5125-9632 (O. Vietrov); 0000-0002-3925-5815 (O. Trofymenko); 0009-0005-8094-227X

(V. Trofymenko); 0000-0001-6817-2624 (M. Hryhorov)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The limitation on the amount of memory allocated by computer technology for number processing imposes fundamental limitations on the possibilities and logic of the organization of computer calculations. It's also requires the involvement of specific mathematical research methods.

The standard and most popular tool for computer calculations, especially in machine learning tasks, is the Python programming language, in particular with its powerful extension NumPy module. It allows you to efficiently work with large multidimensional arrays and matrices, along with a large library of mathematical functions. The NumPy module provides much wider possibilities for working with mathematical functions and objects compared to the math module. The NumPy is an effective tool for providing complex mathematical calculations not only in the tasks of machine learning and data analysis, but is also an indispensable component in the implementation of a large volume of computer calculations in a wide class of modern engineering and scientific tasks [7-10].

The investigation considers the main aspects related to the problems of computer calculations within the IEEE-754 [11] standard, the main technical standard format for representing floating-point numbers (floating-point numbers) (discussing possible alternatives to the IEEE-754 standard, such as Unum [12] and Posit [13-14] go beyond the scope of the presented work).

The history of the development of computing and the basics of the theory of computing with floating-point numbers are described in [15-16]. Important related issues of floating-point computing are presented in [17-18]. Recently, the research in the direction of computer-assisted proof for various classes of mathematical problems [19] is more developed. In this work, the authors touch on possible incorrect cases of the possible use of proof calculations in case of failure to take into account certain features of the representation of numbers in the computer memory.

The presented work is a continuation of the authors' research [20-21]. Previously, the authors investigated rather complex cases of violation of the correctness of computer calculations, the example of Ramp [22] and the sequence of Müller [15]. The works of [20-21] investigated the mathematical aspects of computer calculations, in particular the convergence and stability of computer calculations. The authors focused more on the methodological principles of teaching the topic of accuracy of calculations for the Computer Science specialty students in the presented work. Based on the teaching experience, the authors believe that a detailed study of the problem of the incorrectness of computer calculations on trivial examples (rather than specially designed ones) will allow students to take a more serious and attentive approach to the problems of computer calculations.

2. Methods

Let's consider the numeric data types of the NumPy library, one of the most popular Python libraries for scientific computing. Figure 1 shows the hierarchical diagram of the `numpy.generic` data class, the base class for numpy scalar types. We not consider components of `numpy.generic` like `numpy.bool_`, `numpy.object_`, `numpy.datetime64` or `numpy.flexible`.

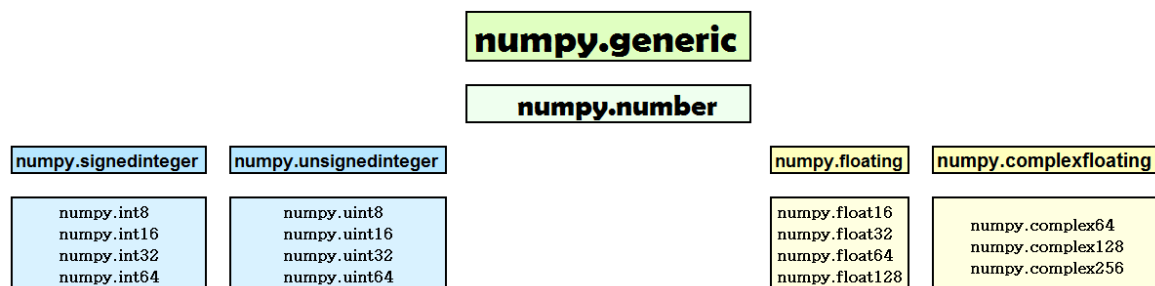


Figure 1: Numpy number types

Floating point numbers according to the IEEE 754 standard are presented in the form

$$number = (-1)^s \times 1.M \times 2^E$$

where S is number sign (signum), M is mantissa, E is number order (exponent). Figure 2 shows the distribution of bits of a single-precision number in the binary32 format (the most significant bit is the sign, the order of the number is preserved using 8 bits, 23 bits are allocated to the mantissa – a total of 32 bits).

Figure 2: Single precision number (IEEE-754)

Table 1

Floating-point types

The mantissa is normalized (the integer part is equal to one), as we can see from the formal representation of the number n . To calculate the order, a constant shift exponent is used, which for single-precision numbers is 127 (or 01111111 in binary code).

$$1/3 = 0.(3)_{10} = 0.(01)_2.$$

For a single-precision number, we can write

i.e $S = 0$ (number is positive), $M = 010101010101010101011$ (since the first discarded bit was a unit, according to the rules of the standard, we add 1 to the mantissa), $E = -10$ (the order is -2 in the decimal number system).

$$\begin{aligned} (0.0101010101010101010101011)_2 &= 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} \dots = \\ &= \sum_{k=1}^{12} \frac{1}{2^{2k}} + \frac{1}{2^{25}} = (0.3333333432674407958984375)_{10} \end{aligned}$$

The obtained approximate value of the number is greater than the real value $1/3$. Everything is still correct in the mathematical sense

Let's consider an important concept of the theory of computer calculations – machine epsilon [23]. This is numerical value below which it is impossible to specify the relative accuracy for any algorithm that produces real numbers.

Listing #1

```

3      iteration_count_eps = 1
4      mach_eps = float_format(1)
5      while float_format(1) + float_format(mach_eps)\
6          /float_format(2) != float_format(1):
7          iteration_count_eps += 1
8          mach_eps /= float_format(2)
9      return mach_eps, iteration_count_eps
10
11     def machineZero(float_format = float):
12         iteration_count_zero = 0
13         mach_zero = float_format(1)
14         while float_format(mach_zero)\
15             /float_format(2) != float_format(0):
16             iteration_count_zero += 1
17             mach_zero /= float_format(2)
18         return mach_zero, iteration_count_zero

```

Result:

Machine Epsilon

```

<class 'numpy.float16'>: (0.000977, 11)
<class 'numpy.float32'>: (1.1920929e-07, 24)
<class 'numpy.float64'>: (2.220446049250313e-16, 53)
<class 'numpy.longdouble'>: (1.084202172485504434e-19, 64))

```

Machine Zero

```

<class 'numpy.float16'>: (6e-08, 24)
<class 'numpy.float32'>: (1e-45, 149)
<class 'numpy.float64'>: (5e-324, 1074)
<class 'numpy.longdouble'>: (4e-4951, 16445)

```

We see that a specific machine epsilon, like a machine zero, is not fundamental. The order of given numbers is important. The value of the `iteration_count_eps` variable is the negative power order $2^{-\text{iteration\_count\_eps}}$. As we can see, for the machine epsilon for the corresponding type of floating point numbers, the iteration_count value is one more than the number of bits allocated for the mantissa value for the corresponding type. This makes sense because it implies the precision limits for the corresponding numeric type.

As for the results for machine zero, the given values are the smallest numbers below which the computer already perceives as zero.

That is, in fact, the machine zero is a number $2^{-(\text{iteration\_count\_zero}+1)}$. It is obvious that it is possible to obtain the order of the machine zero without computer calculations using the formula

$$\text{mantissa} + 2^{\text{exponent}-1},$$

where the exponent and the mantissa determine the number of bits for the corresponding data type.

3. Results and Discussion

3.1. Restrictions on calculations in integers

Let's consider one interesting example of working with integers when overflowing a bit grid.

This problem itself is well known, but when training specialists, it is important to demonstrate that the results obtained are incorrect to a human programmer, but are nevertheless absolutely correct from the point of view of executing the program within the framework of existing standards. A simple example is the calculation of the value of the factorial of some natural number n . With implementing directly in Python (the factorial function from the `math` module), there are no problems with accuracy. Python implements the `BigInt` [24]

mechanism by default, that is, the value of an integer is not limited by the number of bits and can expand to the limit of available memory. At the same time, using the capabilities of the numpy module allows you to explore in more detail the nuances of factorial calculation for different types of integers.

Table 2 lists [25] some of the signed and unsigned integer types available in numpy.

The second column in Table 1 is the alias of the corresponding data types on the Linux x86_64 platform.

Table 3 shows the results of calculating the factorials of the number n in Python depending on the types of integers. The correct result is obtained using Python's standard `int()` (the `BigInt` concept mentioned above).

As we can see from Table 3, when calculating the factorial using the `int32` type, starting with $n = 13$, the calculation result is incorrect. Using the `int32` and `int64` types, starting with a count of $21!$, Python throws an `OverflowError` exception.

Note that the `numpy.intc` and `numpy.int_` types are analogues of the `int` and `long long` types in C, while the generation of the `OverflowError` exception is a feature of the numpy module itself, while in C we would get the corresponding values

```
int factorial(21) = - 1195114496,
long long (21) = - 4249290049419214848.
```

We obtained a paradoxical result, that the product of natural numbers is equal to a negative number.

In integers in the `int` and `long long` formats, everything is calculated correctly.

Table 2
Integer types (Signed integer types)

Numpy type	Alias	Range of Values	
byte	int8	$-2^7 \dots 2^7-1$	-128 .. 127
short	int16	$-2^{15} \dots 2^{15}-1$	-32768 .. 32767
intc	int32	$-2^{31} \dots 2^{31}-1$	-2147483648 .. 2147483647
int_	int64	$-2^{63} \dots 2^{63}-1$	-9223372036854775808 .. 9223372036854775807

Integer types (Unsigned integer types)

ubyte	uint8	$0 \dots 2^8-1$	0 .. 255
ushort	uint16	$0 \dots 2^{16}-1$	0 .. 65535
uintc	uint32	$0 \dots 2^{32}-1$	0 .. 4294967295
uint	uint64	$0 \dots 2^{64}-1$	0 .. 18446744073709551615

The real value is $21!$ in decimal and binary number systems is

$21! = (51090942171709440000)_{10}$,

$21! =$

$(10110001010000011101111101001101101011100011000100000000000000000000)_2$

Table 3
Factorial calculation depending on integer type

n	Correct value n!	int32(n!)	int64(n!)
1	1	1	1
2	2	2	2
3	6	6	6
4	24	24	24
5	120	120	120
6	720	720	720
7	5040	5040	5040
8	40320	40320	40320

3.2. Homer Simpson's numbers

Let's consider another example of manipulating integer formats. The images in Figure 3 are stills from the episode "The Wizard of Evergreen Terrace" [26] and the episode "Treehouse of Horror VI. Homer3 (Homer Cubed)" [27] of the popular animated series The Simpsons.

Mathematical statements relating to Fermat's Great Theorem will be true when checked with a calculator [28]. Let's consider the first example when working with numeric data types from the numpy module.

Of course, the actual numerical values do not match

$$\begin{aligned} 3987^{12} + 4365^{12} &= 63976656349698612616236230953154487896987106 \\ 4472^{12} &= 63976656348486725806862358322168575784124416 \end{aligned}$$

Note that an attempt to convert any number to int32, int64 (uint32, uint64) types generates an OverflowError error.

Direct conversion to float32 also generates an OverflowError, so let's try to consider a mathematically identical expression

$$398.7^{12} + 436.5^{12} \neq 447.2^{12}$$

Now the result of the operation in Python ($398.7^{12} + 436.5^{12} == 447.2^{12}$) will be True. Indeed, it is fashionable to write within the IEEE-754 standard

Arithmetic expression:	$398.7^{12} + 436.5^{12}$
Correct value:	63976656349698612616236230953154.487896987106
Float32 format:	63976654187609440353486161051648.00
Binary format:	0 11101000 10010011101111111101110
Arithmetic expression:	447.2^{12}
Correct value:	63976656348486725806862358322168.575784124416
Float32 format:	63976654187609440353486161051648.00
Binary format:	0 11101000 10010011101111111101110

That is, the reason lies in discarding the lower bits when converting a number to the float32 format.

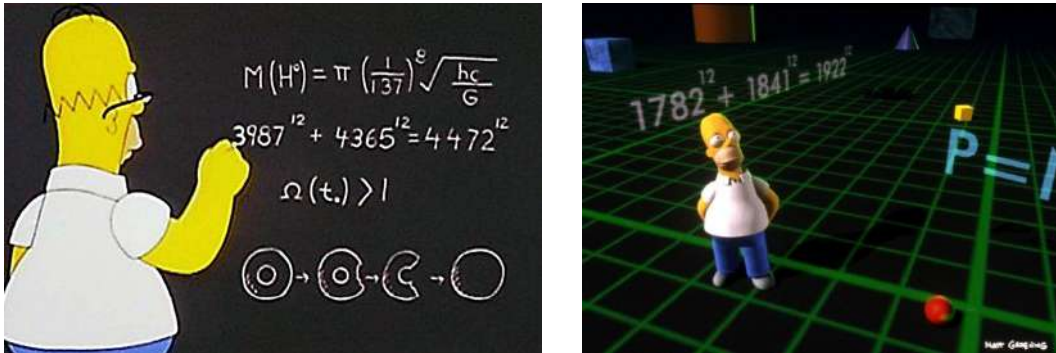


Figure 3: Homer Simpson's example

Let's consider the same problem, but staying within the framework of integer formats.

For example, for $n = 5$ for the int32 format, more than 1400 triples of natural numbers (A, B, C) can be found, so that when calculating $\text{int32}(A^{**5}) + \text{int32}(B^{**5}) == \text{int32}(C^{**5})$ would return a boolean value of True.

The maximum value of the parameter C: $C = 6208$, and as can be seen from the Table 4, several pairs (A, B) can correspond to one value of C. The explanation for this is simple: different large integers (more than 4 bytes) due to the discarding of the high-order bits in the int32 format can have the same value. For triples from the Table 4 (32, B, 6208) we have

$$\begin{aligned} \text{int32}(6208^{**5}) &= 1073741824, \\ \text{int32}(992^{**5}) &= \text{int32}(5088^{**5}) = 1040187392, \text{int32}(32^{**5}) = 33554432. \end{aligned}$$

Similarly for the triples (64, B, 6208) we have

$$\text{int32}(B^{**5}) = 0, \text{int32}(64^{**5}) = 1073741824.$$

Table 4
C² = B² + A² (float32)

C	B						A
	992, 5088						32
	128,	256,	384,	512,	640,	768,	
	896,	1024,	1152,	1280,	1408,	1536,	
	1664,	1792,	1920,	2048,	2176,	2304,	
6208	2432,	2560,	2688,	2816,	2944,	3072,	
	3200,	3328,	3456,	3584,	3712,	3840,	64
	3968,	4096,	4224,	4352,	4480,	4608,	
	4736,	4864,	5120,	4992,	5248,	5376,	
	5504,	5632,	5760,	5888,	6016,	6144	

Most of the examples for $n = 5$ look artificial and any user will immediately understand that the obtained result is incorrect from a mathematical point of view. But you can choose an example when an untrained user can mistakenly perceive the result as correct, for example, implementing C++ programs for the standard type `int` (4 bytes). From our point of view, this is the example of $A = 1504$, $B = 2960$, $C = 4496$, $n = 5$. At the same time, it is interesting that the values of the expressions A^5 and B^5 in the `int32` format are generally negative numbers:

```
int32(1504**5) = -570425344,
int32(2960**5) = -1584398336,
int32(4496**5) = 2140143616.
```

In the additional code in `int32` format, we can write

```
(-570425344)10 = (11011110000000000000000000000000)2
(-1584398336)10 = (10100001100100000000000000000000)2
```

In the binary code, the sum of these numbers will have a length of 33 bits and is equal to

```
(-570425344)10 + (-1584398336)10 = (10111111100100000000000000000000)2
```

Discarding the most significant bit (33 bits), we get the result

```
(01111111100100000000000000000000)2 = (2140143616)10.
```

3.3. Another simple examples

Let's move on to consider some typical examples of incorrect behavior of computational algorithms about working with floating point numbers.

Consider the following problem: it is necessary to find such a minimum value of n that

$$\frac{1}{m} + \frac{1}{n} = \frac{1}{2020}, \quad \begin{matrix} m \in N \\ n \in N' \end{matrix}, \quad n > m$$

This problem is an example of Diophantine equation. In a general sense, the Diophantine equation is an equation in which only integer solutions are allowed.

The program for solving this equation is presented in listing #2.

Listing #2

```
1 import numpy as np
2 def mathExpr (num_format):
3     value = num_format(1/2020)
4     n = 1
5     while True:
6         n1 = num_format(1/n)
7         for m in range(1, n):
8             if n1 + num_format(1/m) == value:
9                 return n, m
10        n += 1
11 float_formats = (np.float16, np.float32, np.float64, np.float128)
```

```

12 for form in float_formats:
13     print(f'{form}:", mathExpr (form))

```

```

Result: <class 'numpy.float16'>:      n = 4039, m = 4035
        <class 'numpy.float32'>:      n = 4041, m = 4039
        <class 'numpy.float64'>:      n = 4545, m = 3636
        <class 'numpy.longdouble'>:   n = 6060, m = 3030

```

As a result, we got four different answers depending on the format of the floating point numbers. At the same time, the pairs (4545, 3636) and (6060, 3030) are the correct solution to the equation.

It is easy to show that $n = 4545$, if we look for the minimum value of n . That is, when using the float16 and float32 formats, we will get incorrect answers, and in the case of the float128 format, we will get a correct answer from a mathematical point of view, but we will skip over the minimum possible value of n ($n = 4545$), which satisfies the problem. We can engage the specialized apparatus of computer computation to improve the accuracy of computation. Let's try to increase the accuracy of calculations in Python using a special decimal module [30] by adding a decimal.Decimal element to the float_formats tuple. However, when trying to execute the program, we will get into an infinite loop, and the task of finding an exact match of values can be considered as failed.

We can correct this situation, for example, using the isclose method from the math module. To do this, in listing #2, line 8 must be replaced with the command *if math.isclose(n1+num_format(1/m), value, rel_tol=1e-9)*, and in this case, the float128 and decimal.Decimal format will already have $n = 4545$, $m = 3636$. Let's change line 3 in listing #1 to the command *value = num_format(1/10)*, and after starting the program, we will get the result:

```

<class 'numpy.float16'>: n = 30, m = 15
<class 'numpy.float32'>: n = 35, m = 14
<class 'numpy.float64'>: n = 30, m = 15
<class 'numpy.longdouble'>: Infinite loop
<class decimal.Decimal'>: Infinite loop

```

Obviously, the answer to the problem is the pair $n = 30$, $m = 15$ (the pair $n = 35$, $m = 14$ is also a correct result, but the minimum value of the parameter n is still equal to 30). With the double-precision format, the program falls into an infinite loop, that is, the result cannot be calculated.

If we calculate the specified example using the isclose function, then for the case of *rel_tol=1e-7* for all formats of floating point numbers the result will be $n = 30$, $m = 15$, and for the value of the *rel_tol* parameter of greater precision (for example, already *rel_tol=1e-8*) – for float16, float64, float128, decimal.Decimal formats the result will be $n = 30$, $m = 15$, and for float32 format the result will still be $n = 35$, $m = 14$.

We can make an interesting intermediate conclusion: in some examples of computer calculations with floating point numbers, a given lower precision of calculations can give a more correct result than the procedure of increasing the precision. This effect is precisely what we observe in the described example: the correct answer is provided by the calculation with the float16 data type (as opposed to float32), the correct result is achieved with the value of the parameter *rel_tol=1e-7*, and not with the value $0 < \text{rel\_tol} \leq 1e-8$.

Finally, we will consider the comparison of calculating the values of the square root of integers using two operations from the math module: *sqrt* and *pow*. Despite the simplicity of the problem statement, calculating the square root of a number, the whole root of a number, and especially the inverse square root of a natural number [29] is a relevant task both to correctness of calculations and the time efficiency of algorithms. For example, consider the range of numbers from 1 to 10000, the data type of floating point numbers is standard float (numpy.float64). We will iterate through the values until we find the number value such that *math.sqrt(number) != math.pow(number)*. Table 5 shows the following numbers (for example, less than 10000) depending on the version of Python and the compiler. The table shows only typical examples of the results of computer calculations of the proposed problem. For some

versions (for example, for Python 3.6.5 [GCC 7.3.1], there are no such values under the specified restrictions).

Table 5
Comparison of sqrt and pow functions

Python version	Compiler	Numbers		
3.7.4	GCC 9.2.0			
3.8.10	GCC 9.4.0			
3.9.9	GCC 11.1.0	2921,	3541,	5579,
3.10.6	GCC 11.3.0	7827,	8414,	8415
3.10.12	GCC 11.4.0			
3.11.5	GCC 13.2.1			
3.8.2	GCC 9.3.0	550,	971,	2921,
3.8.5	GCC 9.3.0	3541,	3543,	3884,
3.10.2	Clang 15.0.0	5579,	7827,	8414,
3.11.3	Clang 18.0.0	8415,	8451,	8800

Without going into the technical nuances of the implementation of these algorithms, I will explain the reason for the mismatch of values for the corresponding numbers from Table 5.

```

math.sqrt(2921)    = 54.04627646748664204778833664022386074066162109375
binary(float64)   01000000 01001011 00000101 11101100
                  01100011 00100101 00110110 11111011

math.pow(2921, 0.5) = 54.04627646748663494236097903922200202941894531250
binary(float64)   01000000 01001011 00000101 11101100
                  01100011 00100101 00110110 11111010

```

We see from analysis of the binary representation of calculation results in float64 format that the difference is in the least significant bit of the mantissa. The reason is that the first discarded bit in the representation of the number is 1, and with the calculating `math.sqrt(2921)` this fact is taken into account. And 1 is added to the least significant bit of the mantissa, but not for calculating `math.pow(2921, 0.5)`. For example, both values are calculated with the increment of the least significant bit taken into account in Python 3.6.5 [GCC 7.3.1].

Conclusions

The work presents the results of the authors' research in the field of methodological foundations of teaching academic disciplines for students of the Computer Science specialty. The authors focused their attention on the topic of accuracy of computer calculations of various problems. This problem is relevant not only when performing engineering and scientific calculations, but also, as shown in the article, when solving the simple training problems. The authors focused on the study of the specified problem using the example of the numpy module, as one of the main software tools for modern scientific calculations.

The authors of the presented study focused on considering fairly simple examples of computer calculations that are completely correct from the point of view of the standards of computer work with numbers and arithmetic operations, the results are incorrect to a human performer. It has been demonstrated that mathematically correct calculations are often impossible to reproduce correctly within the IEEE-754 standard due to the fundamental limitations of the memory allocated to represent a specific number.

Using specific examples, it has been shown how the use of special Python computation features, such as the decimal module, can nevertheless lead to incorrect results. The use of external modules such as mpmath requires further research.

The work also gives paradoxical examples, when less accurate calculations allow you to get a more correct result. Such a statement contradicts the classical practices of calculation methods, and requires a more thorough study of the specifics of computer calculations.

It's important, for the training future specialists in the field of Computer Science during the teaching of subjects related to the accuracy of computer calculations and the fundamental limitations of computer calculations, to present all facts and principles from a single point of view so that students do not there was a false opinion about the arbitrariness of the results of computer calculations and the randomness of the corresponding errors of computer calculations. For a Computer Science professional, it is important not only to understand the basic principles of working with the floating-point number format from the standpoint of the generally accepted IEEE standard, but also to know, even at a basic level, possible modern alternatives, such as, for example, the concept of Posit.

Acknowledgements

All computer calculations and implementation of computer-mathematical models were supported by the computational capacities of the Laboratory of Machine Learning and Intellectual Data Analysis of the Vasyl' Stus Donetsk National University. The laboratory was founded in 2020 by Artem Baiev to conduct academic research and educational work.

References

- [1] U. Kulisch, R. Hammer , D. Ratz , and M. Hocks, Numerical Toolbox for Verified Computing I. Basic Numerical Problems. Theory, Algorithms, and Pascal-XSC Programs, Springer Berlin, Heidelberg, 1993. doi:10.1007/978-3-642-78423-1
- [2] R. Skeel, "Roundoff Error and the Patriot Missile", SIAM News, 25(4), 1992, 11.
- [3] T. Sağlam, "Deadly Round-Off Error: Failure of the Patriot System in Dhahran 1991", Awarded Proseminar and Seminar Papers at the Chair of Software Design and Quality, 1, 2016.
- [4] ARIANE 5. Flight 501 Failure, 1996. URL:<https://www-users.cse.umn.edu/~arnold/disasters/ariane5rep.html>
- [5] G. Le Lann, "An analysis of the Ariane 5 flight 501 failure-a system engineering perspective", Proceedings International Conference and Workshop on Engineering of Computer-Based Systems, 1997, 339-346. doi: 10.1109/ECBS.1997.581900
- [6] Disasters due to rounding error, 2022. URL:<https://web.ma.utexas.edu/users/arbogast/misc/disasters.html>
- [7] S.C. Chapra, S. Chapra, and D.E. Clough, Applied Numerical Methods with Python for Engineers and Scientists, McGraw Hill, 2022.
- [8] P. Dechaumphai, and N. Wansophark, Numerical Methods in Science and Engineering Theories with MATLAB, Mathematica, Fortran, C and Python Programs, Alpha Science International, 2022.
- [9] C. Fuhrer, J.E. Solem, and O. Verdier, Scientific Computing with Python: High-performance scientific computing with NumPy, SciPy, and pandas, Packt Publishing, 2021.
- [10] R. Johansson, Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib, Apress, 2019. doi:10.1007/978-1-4842-4246-9
- [11] "IEEE Standard for Floating-Point Arithmetic" in IEEE Std 754-2019 (Revision of IEEE 754-2008), 2019. doi:10.1109/IEEESTD.2019.8766229
- [12] J. Gustafson, The End of Error. Unum Computing, Chapman and Hall/CRC, 2015.
- [13] M. Feldman, New Approach Could Sink Floating Point Computation, 2019. URL: <https://www.nextplatform.com/2019/07/08/new-approach-could-sink-floating-point-computation/>
- [14] J. Gustafson and etc., Standard for Posit™ Arithmetic, Posit Working Group, 2022. URL:https://posithub.org/docs/posit_standard-2.pdf
- [15] J.-M. Muller, N. Brisebarre, F. de Dinechin, C.-P. Jeannerod etc., "Handbook of Floating-Point Arithmetic", Birkhäuser Basel, 2018. doi:10.1007/978-3-319-76526-6
- [16] S. Boldo, C.-P. Jeannerod, G. Melquiond and J.-M. Muller, "Floating-point arithmetic", Acta Numerica, 32 (2023), 203–290. doi:10.1017/S0962492922000101

- [17] D.M. Russinoff, Formal Verification of Floating-Point Hardware Design, Springer Cham, 2019. doi:10.1007/978-3-319-95513-1
- [18] S. Boldo and G. Melquiond, Computer Arithmetic and Normal Proofs, ISTE Press - Elsevier, 2017. doi.org:10.1016/C2015-0-01301-6
- [19] M.T. Nakao, M. Plum, and Y. Watanabe, "Numerical Verification Methods and Computer-Assisted Proofs for Partial Differential Equations", Springer Singapore, 2019. doi:10.1007/978-981-13-7669-6
- [20] O. Vietrov and R. Bilous, "Special methods of increasing the accuracy of computer calculations", 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek) (2022), 1–5. doi:10.1109/KhPIWeek57572.2022.9916383
- [21] O. Vietrov and R. Bilous, "Study of the Convergence of Muller's Sequence Computer Calculations", 2021 IEEE 3rd Ukraine Conference on Electrical and Computer Engineering (UKRCON) (2021), 547–551. doi:10.1109/UKRCON53503.2021.9575546
- [22] S.M. Rump, "Algorithms for Verified Inclusions: Theory and Practice", Reliability in Computing: the Role of Interval Methods in Scientific Computing (1988), 109–126. doi:10.15480/882.316
- [23] L.G. de la Fraga, "Differential Evolution under FixedPoint Arithmetic and FP16 Numbers", Math. Comput. Appl., 26 (1), 13 (2021). doi:10.3390/mca26010013
- [24] Integer Objects, 2024. URL:<https://docs.python.org/3/c-api/long.html>
- [25] Data types, 2023. URL:<https://numpy.org/doc/stable/user/basics.types.html>
- [26] R. Gray, Did Homer Simpson discover the HIGGS BOSON?, 2015. URL: <https://www.dailymail.co.uk/sciencetech/article-2975606/Did-Homer-Simpson-discover-HIGGS-BOSON-Maths-1998-episode-predicts-particle-s-mass-14-years-CERN.html>
- [27] D. Snierston, The Simpsons: A somewhat complete history of 'Homer?' from 'Treehouse of Horror VI', 2018. URL:<https://ew.com/tv/2018/10/19/the-simpsons-homer-cubed-treehouse-of-horror-vi/>
- [28] S. Singh, The Simpsons and Their Mathematical Secrets, Bloomsbury USA, 2013.
- [29] C.J. Walczyk, L. Moroz, and J. Cieslinski, "A Modification of the Fast Inverse Square Root Algorithm", Computation, 7(3):41 (2019). doi:10.3390/computation7030041
- [30] Decimal fixed point and floating point arithmetic, 2024. URL:<https://docs.python.org/3/library/decimal.html>