Sergey Subbotin
Editor

Computer Modeling and Intelligent Systems

Proceedings of The Sixth International Workshop
CMIS-2022

Zaporizhzhia, Ukraine
May 3, 2023

Subbotin S., (Ed.): The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023). Zaporizhzhia, Ukraine, May 3, 2023, CEUR-WS.org, online

This volume represents the proceedings of the Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), held in Zaporizhzhia, Ukraine, in May, 2023. It comprises 22 contributed papers that were carefully peer-reviewed and selected from 129 submissions.

Preface

It is my pleasure to present you the proceedings of The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), held in Zaporizhzhia, Ukraine on 3 of May 2023.

The CMIS workshop was held in 2023 for the sixth 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-2023 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-2023 workshop countries of PC members and authors of submitted papers are: Algeria, Belgium, Canada, China, Egypt, Finland, Germany, Israel, Latvia, Malasia, Netherlands, Poland, Portugal, Slovak Republic, Spain, Ukraine, and USA. The number of countries of CMIS-2023 authors and PC members is 17.

The language of CMIS-2023 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/cmis.workshop/.

The CMIS-2023 Organizing Committee have received 129 paper submissions, out of which 22 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. The paper acceptance ratio in 2023 is 17%. These papers were published in this volume of CMIS-2023 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, 2023

Sergey Subbotin, Dr. Sc., Prof.,
CMIS-2023 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. **Andreas Pester**, British University in Egypt, Egypt

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. **Carsten Wolff**, Dortmund University of Applied Sciences and Arts, Germany

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

Dr. **Peter Arras**, Katholieke Universiteit Leuven, Belgium

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

Dr. **Dmitry Gorodnichy**, Canada Border Services Agency, Canada

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
MSc **Serhii Leoshchenko**, National University "Zaporizhzhia Polytechnic", Ukraine
Eng **Maksym Andreiev**, National University "Zaporizhzhia Polytechnic", Ukraine

Partners

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)

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

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

Reviewers

Name	Email	Number of reviews
Alina Yanko	al9_yanko@ukr.net	4
Amel Boustil	a.boustil@univ-boumerdes.dz	3
Andrii Chugay	chugay.andrey80@gmail.com	4
Andrii Kopp	kopp93@gmail.com	4
Andriy Goncharenko	andygoncharenco@yahoo.com	4
Artem Sokolov	radiosquid@gmail.com	4
Artyom Yegorov	for___students@ukr.net	3
Dmitriy Klyushin	dokmed5@gmail.com	4
Dmytro Bodnenko	d.bodnenko@kubg.edu.ua	4
Dmytro Fedasyuk	dmytro.v.fedasyuk@lpnu.ua	4
Elena Bakunina	elenabakunina72@gmail.com	4
Fargana Abdullayeva	a_farqana@mail.ru	4
Georgiy Yaskov	yaskov2004@gmail.com	3
Halyna Kozbur	kozbur.galina@gmail.com	2
Igor Golinko	conis@ukr.net	8
Ihor Pilkevych	igor.pilkevich@meta.ua	4
Ihor Prots'Ko	ihor.o.protsko@lpnu.ua	4
Iryna Strutynska	strutynskairy@gmail.com	2
Karen Egiazarian	karen.eguiazarian@tuni.fi	1
Karina Melnyk	karina.v.melnyk@gmail.com	4
Kateryna Brovko	k.brovko@kubg.edu.ua	2
Lesia Dmytrotsa	dmytrotsa.lesya@gmail.com	3
Lyudmila Akhmetshina	akhmlu1@gmail.com	2
Lyudmyla Kirichenko	lyudmyla.kirichenko@nure.ua	4
Milan Guzan	milan.guzan@tuke.sk	4
Mykhailo Kopp	michaelkopp0165@gmail.com	4
Mykhailo Yadzhak	yadzhak_ms@ukr.net	4

Name	Email	Number of reviews
Nataliia Khatsko	n.khatzko@gmail.com	3
Nataliia Kuznietsova	natalia-kpi@ukr.net	1
Nataliya Boyko	nataliya.i.boyko@lpnu.ua	4
Nataliya Ryabova	nataliya.ryabova@nure.ua	3
Nickolay Rudnichenko	nickolay.rud@gmail.com	3
Oksana Choporova	o.choporova@gmail.com	1
Oksana Sychova	oksana.sychova@nure.ua	4
Oleg Dekusha	olds@ukr.net	4
Oleg Rudenko	oleh.rudenko@nure.ua	3
Oleh Pihnastyi	pihnastyi@gmail.com	4
Oleksandr Andreiev	oleks.andreyev@gmail.com	4
Oleksandr Bezsonov	oleksandr.bezsonov@nure.ua	4
Oleksandr Filipenko	oleksandr.filipenko@nure.ua	4
Oleksandr Mitsa	alex.mitsa@gmail.com	3
Oleksandra Bulgakova	sashabulgakova2@gmail.com	4
Oleksii Bychkov	oleksiibychkov@knu.ua	3
Olexandr Polishchuk	od_polishchuk@ukr.net	4
Olha Kozhyna	olga.kozhyna.s@gmail.com	1
Rostyslav Tarasenko	r_tar@nubip.edu.ua	4
Sergii Khlamov	sergii.khlamov@gmail.com	4
Serhii Choporov	s.choporoff@znu.edu.ua	3
Serhii Vladov	ser26101968@gmail.com	4
Serhiy Shtovba	shtovba@gmail.com	2
Svitlana Amelina	svetlanaamelina@ukr.net	4
Svitlana Kovtun	sveta_kovtun@ukr.net	4
Tatiana Romanova	tarom27@yahoo.com	4
Tatyana Bulanaya	tatyana.bulanaya@gmail.com	4
Tetiana Korobeinikova	tetiana.i.korobeinikova@lpnu.ua	4
Tetiana Vakaliuk	tetianavakaliuk@gmail.com	4

Name	Email	Number of reviews
Tetyana Marusenkova	tetiana.a.marusenkova@lpnu.ua	3
Vadym Shkarupylo	shkarupylo.vadym@nubip.edu.ua	2
Valentyna Dusheba	vdusheba@gmail.com	4
Vasyl Dubuk	vasyl.i.dubuk@lpnu.ua	4
Vasyl Teslyuk	vasyl.m.teslyuk@lpnu.ua	4
Viacheslav Zosimov	zosimovvv@gmail.com	4
Victor Krasnobayev	v.a.krasnobaev@gmail.com	4
Vladimir Vichuzanin	vint532@yandex.ua	1
Volodymyr Hnatushenko	vvgnat@ukr.net	4
Volodymyr Riznyk	volodymyr.v.riznyk@lpnu.ua	2
Volodymyr Rusyn	rusyn_v@ukr.net	3
Volodymyr Vychuzhanin	vint532@yandex.ua	2
Vsevolod Bohaienko	sevab@ukr.net	4
Vyacheslav Korolyov	korolev.academ@gmail.com	4
Yevheniia Znakovska	zea@nau.edu.ua	4
Youcef Tabet	y.tabet@univ-boumerdes.dz	4
Yuliya Averyanova	ayua@nau.edu.ua	4
Yuriy Stoyan	stoyan@ipmach.kharkov.ua	4
Zinaida Burova	zinaburova@nubip.edu.ua	4

Basic Theoretical Provisions of Entropy Approach for Intelligent Air Transportation Management

Andriy V. Goncharenko

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

Abstract

The paper is dedicated to the elaboration of the principal theoretical provisions for the optimal governing of the intelligent air transportation management system. Subjective entropy maximum principle developed for active systems control is an initial postulate. The optimization theory of the subjective individuals' preferences is the working hypothesis. Based upon the Jaynes' principle of the entropy maximum the preferences functions are being found in the explicit view. Conditional optimization of the subjective preferences functions entropy, as the uncertainty measure modeling the available operational multi-alternativeness, allows organizing some rational air transportation system's managerial efforts. Illustrative example simulation is performed. Necessary diagrams are plotted.

Keywords

Entropy, preferences, uncertainty, air transport, optimization, variation, management, concept, objective functional.

1. Introduction

Air transportation management operates in conditions with a rather uncertainty degree. On one hand there must be all necessary aircraft maintenance and repair procedures performed [1] including the most complex airplane structural parts such as the aircraft powerplants [2]. The due course airplane maintenance carried out in the due scope and due time scheduled ensures the aeronautical engineering operational reliability and decreases the risks of the probable aviation incidents and accidents happening [3, 4].

It is undoubtedly that the air transportation management chasing the goals of their business profitable running considers expected utilities of such kind of activity [5]. The complexity of this operational situation makes an impact upon the individual choice behavior [6].

The famous Jaynes' principle of the entropy maximum [7 – 9] allows taking into account the uncertainty degree in not only purely physical systems, and the entropy based research penetrated all spheres of science [10], but the entropy paradigm is also applicable to economic activity studies [11] in the context of the subjective analysis of active systems [12].

The scientific gap of the art-on-the-day investigations contains the lack of the uncertainty degree conditional optimization problem settings that relate with the statistical estimations in the aviation radio equipment reliability parameters monitoring [13], novelty smart materials creation [14], neural network modelling targeted at the transportation challenging tasks performance prediction [15]. The entropy ideas could be applicable to the neuro-fuzzy network synthesis [16] optimization as well.

Thus, the ideology of an entropy based theoretical research stream [7 – 10, 12, 17 – 20] should be prolonged further with the help of the previously elaborated provisions of [12], which has already been successfully implemented in [12, 17 – 20].

CMIS-2023: The Sixth International Workshop on Computer Modeling and Intelligent Systems, Zaporizhzhia, Ukraine, May 3, 2023
EMAIL: andygoncharenco@yahoo.com (A.V. Goncharenko)
ORCID: 0000-0002-6846-9660 (A.V. Goncharenko)

2. Theoretical contemplations

In order to simulate functioning of some air transportation management intelligent system it is proposed to use the entropy approach [7 – 9] developed for the active systems modeling [12].

2.1. Active systems concept

Considering an intelligent air transportation management system as an active one it is proposed to compile the objective functional in the view of [7 – 9, 12, 17 – 20]:

$$\Phi_{\pi} = \alpha H_{\pi} + \beta\varepsilon + \gamma \mathsf{N} \quad , \tag{1}$$

where α, β, and γ are corresponding structure parameters that can be considered at different problems settings as the uncertain Lagrange multipliers, some coefficients or weight coefficients. Here they are interpreted as the internal intelligent air transportation management system's parameters dealing with the intelligent artificial subject's "estimation" of the effectiveness of the alternatives available at the operational situation. H_{π} is entropy of the alternatives preferences π; ε is a function of the effectiveness that together with the alternatives preferences entropy H_{π} determines conditions of the attainable alternative preferences π distribution optimality; N is normalizing condition.

It is important to realize that the regularities of the artificial subject's intelligence existence and functioning have to resemble the natural intellectual properties of the individual responsible for making the managerial decisions in the considered sphere of the activity.

Therefore, the structure parameters of α, β, and γ 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 [12].

One of the active system's peculiarities is the system's controllability (being managed conditions) through the active element and by the active element of that active system. In particular, in the aviation and air transportation field of industry, it is the uncertainty of the operational situations that can lead to some dangerous states and conditions. That is why the active element (an individual or a subject making the responsible controlling or managerial decisions) is forced to the take actions in the conditions of some multi-alternativeness and conflicts. It happens because of the necessity to act at the presence of the lack of the resources (such as, for example, the deficiency of time, limitations of the altitudes, distances etc.).

There is no doubt that safety in aviation is the top priority. Thus, it is obvious that there is great urgency, importance, and actuality in studies touching the directions related with the influence of the notorious human factor upon safety. Such problematic is important also because the stated problems solutions require expressing individuals' subjective preferences functions with the help of some functions taken in the explicit view.

The systematic fundamental research of the multi-alternativeness in the framework of the approach connected with the concept of the problem-resource situation development has been apparently initiated in the monograph [12]. There, it has been considered mainly two adjacent problems of the general formulation; for the presented theoretical study those problems are coupled with the objective intelligent air transportation management system functional (1), [12]:

2.1.1. Conditional extremum of entropy

This formulation following [12] deals with the necessity to obtain the optimal distribution of the subjective preferences functions of the active element of the active system (individual's subjective preferences functions) $\pi(\sigma_i)$; at the set of the achievable for the individual's goals alternatives of σ_i, $i = 1, \ldots, N$, where N is the number of the attainable for the individual's goals alternatives; that are delivering a conditional extremum value to the entropy H_{π} of the subjective preferences functions of $\pi(\sigma_i)$, [12]:

$$\varepsilon(\pi,\ U,\ \ldots\ ) = \varepsilon_0\,; \qquad P_\pi(\pi) = A_0\,; \qquad \pi_{extr}(\cdot) = \arg \underset{\pi(\cdot)\in\Pi}{extr}\, \Phi_\pi\,, \tag{2}$$

where ε is a corresponding function of the subjective effectiveness; U is a corresponding function of the subjective utility; ε_0 is "isoperimetric constraints" for the function of the subjective effectiveness of ε; $P_\pi(\pi)$ is a certain subjective "perimetric function" for the functions of the subjective preferences of $\pi(\sigma_i)$; A_0 is "isoperimetric constraints" for the subjective "perimetric function" of $P_\pi(\pi)$; Π is the class of the subjective preferences functions of $\pi(\sigma_i)$, at which the extremal distribution of the $\pi_{extr}(\cdot)$ functions is chosen.

2.1.2. Conditional extremum of effectiveness

The second formulation is also after [12]; and it is about the counter problem setting: to find the distribution of the subjective preferences functions of $\pi(\cdot)$, delivering the conditional extremal value to the function of the subjective effectiveness of $\varepsilon(\pi,\ U,\ \ldots\ )$, subject to the "isoperimetric" constraints, [12]:

$$H_\pi = H_0\,; \qquad P_\pi(\pi) = A_0\,; \qquad \pi_{extr}(\cdot) = \arg \underset{\pi(\cdot)\in\Pi}{extr}\, \Phi_\pi\,, \tag{3}$$

where H_0 is a corresponding "isoperimetric constraints" for the subjective entropy of H_π.

Thus, in the view of the above formulations of the expressions of (2) and (3), it is postulated in the framework of the subjective analysis [12] (the theory of subjective preferences, or the entropy paradigm of the subjective individuals' preferences functions with respect to the available alternatives effectiveness functions, or the theory of active systems) that, for the active (in the presented case intelligent) control or management of the intelligent air transportation management system, the optimized objective functional could be constructed in the rather general form of (1), [12].

2.1.3. Generalization

As the kind of the variational problem generalization in regards to the objective intelligent air transportation management system functional (1), [12], the attempt has been made in the view of the integral form of the objective functional. The objective functional of the type of (1) in the sense of the problem setting of (2) could be represented for some research cases in the view of

$$\Phi_\pi = \int_{t_0}^{t_1} \left(-\sum_{i=1}^{N} \pi_i(t) \ln \pi_i(t) + \beta \sum_{i=1}^{N} \pi_i(t) F_i + \gamma \left[\sum_{i=1}^{N} \pi_i(t) - 1 \right] \right) dt\,, \tag{4}$$

where t is time;

$$-\sum_{i=1}^{N} \pi_i(t) \ln \pi_i(t), \tag{5}$$

is entropy H_π of the subjective preferences functions of $\pi_i(t)$; where β, γ are the structural parameters, analogous to those ones introduced above in the objective intelligent air transportation management system functional of (1); but here, they are already reduced by α; and the previously introduced designations for such parameters therefore are just deliberately kept the same for the conceptual perceptions easiness; F_i is the function of the subjective effectiveness of the i-th available alternative taken for the set of the alternatives consideration on the purpose of the subjective preferences functions entropy conditional optimization; here, these are the elements which correspond with the utility functions, but with a specific meaning related to the particular case problem setting; the final under-integral member linked with:

$$\sum_{i=1}^{N} \pi_i(t) = 1\,, \tag{6}$$

of the identity to zero value of the normalizing condition, is one more of the conventional constrains described above.

Now, the objective intelligent air transportation management system functional constructed in the integral form view of (4), comprising such under-integral members as those of the expressions represented in (5) and (6), encompasses the properties of both subjective analysis postulated form (1) and dynamical optimization characteristics through the time developing processes as it is stated with the form described by (4).

Also, some of the particular cases, for the problem settings in the dynamical framework of the variational problem, could be considering the intensive and extensive parameters optimizations; which is a constantly urgent problematic for the rational intelligent air transportation management system functioning too.

For a simplest variational problem setting, it could be taken into account such intensive and extensive parameters as, for instance, $x(t)$ and $\dot{x}(t)$.

Where

$$\dot{x}(t)=\frac{dx}{dt}, \tag{7}$$

is an intelligent air transportation management system effectiveness controlling function (the first derivative with respect to time) of the controlled parameter of $x(t)$.

In case of (4) – (7), the functions of $x(t)$ and $\dot{x}(t)$, as the functions of the subjective effectiveness of the two attainable alternatives, should have the corresponding subjective preferences functions of the active elements (subjects or individuals responsible for making the intelligent air transportation management system governing decisions): $\pi_1(t)$, $\pi_2(t)$.

As to the modifications of the objective intelligent air transportation management system functional with the elements of both intensive and extensive parameters, it could be proposed a model based upon, let us say, a combinations of the functions of $x(t)$ and $\dot{x}(t)$, for example, one of such construction might be as follows:

$$\Phi_\pi=\int_{t_0}^{t_1}\left(-\sum_{i=1}^{N=4}\pi_i(t)\ln\pi_i(t)+\beta\left[\pi_1(t)x(t)+\alpha_2\pi_2(t)\dot{x}(t)+\alpha_3\pi_3(t)x(t)\dot{x}(t)+\right.\right. \tag{8}$$

$$\left.\left.+\alpha_4\pi_4(t)\frac{\dot{x}(t)}{x(t)}\right]+\gamma\left[\sum_{i=1}^{N=4}\pi_i(t)-1\right]\right)dt,$$

where α_2, α_3, and α_4 are the coefficients that take into account the differences in the measurement units and dimensions for the corresponding elementary functions, pertaining to the achievable alternatives effectiveness functions, described with the expressions of the introduced for the consideration combinations:

$$x(t), \qquad \dot{x}(t), \qquad x(t)\dot{x}(t) \qquad \text{and} \qquad \frac{\dot{x}(t)}{x(t)}. \tag{9}$$

For the objective intelligent air transportation management system functional (8) with the elements of (9), there might be 11 particular cases of the (9) expressions combinations assessing some preferable properties.

2.1.4. Subjective optimality postulate

Formulation of the subjective optimality postulate is based upon the vision of the active systems governing and management [12]. In the considered problematic interpretation this is the intelligent air transportation system reasonable management.

If the controlled information income flow is processed effectively and adequately enough, then the participants of the intelligent air transportation system managerial process (the air transportation system operation, economical activity processes etc.) have a possibility, taking into their own consideration the resources (for example, technical, financial and so on) required for the given process, their own business running, to choose one or another attainable alternative at the available problem-resource situation that has been formed.

Along with this, following the theoretical contemplations and the theories of the described above ideas, traced with the expressions of (1) – (9), the active element of either artificial or natural intelligent governing system (the subject, individual, person responsible for the decision making, key-element of the active system) forms her/his own alternative preferences functions distributions for the achievable for her/his own purposes alternatives with the application of the objective functional taken in the fairly general view (1), [12].

Remarkable here is that constructing the objective intelligent air transportation management system functional in the view of, [12]:

$$\Phi_{\pi}^{-}=-\sum_{i=1}^{N}\pi^{-}(\sigma_i)\ln\pi^{-}(\sigma_i)-\beta\sum_{i=1}^{N}\pi^{-}(\sigma_i)L(\sigma_i)+\gamma\left[\sum_{i=1}^{N}\pi^{-}(\sigma_i)-1\right], \tag{10}$$

where $\pi^{-}(\sigma_i)$ is a corresponding negative subjective individual's preference function of the responsible decision making person; $L(\sigma_i)$ is the corresponding function of the personally estimated losses related with the available alternatives; the individual distinguishes a certain one-sided attitude to the managerial process.

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

$$\Phi_{\pi}^{+}=-\sum_{i=1}^{N}\pi^{+}(\sigma_i)\ln\pi^{+}(\sigma_i)+\beta\sum_{i=1}^{N}\pi^{+}(\sigma_i)U(\sigma_i)+\gamma\left[\sum_{i=1}^{N}\pi^{+}(\sigma_i)-1\right], \tag{11}$$

where $\pi^{+}(\sigma_i)$ is a corresponding positive subjective individual's preference function of the responsible decision making person; $U(\sigma_i)$ is 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 (10) and (11) that is both the negative (losses, harmfulness) and positive (utility, usefulness) estimations are possible.

And the both-sided estimated managerial process of (10) and (11) is extremized with the use of the necessary conditions for the subjective preferences functions entropy conditional optimization in the view of

$$\frac{\partial\Phi_{\pi}^{-}}{\partial\pi^{-}(\sigma_i)}=0, \qquad \frac{\partial\Phi_{\pi}^{+}}{\partial\pi^{+}(\sigma_i)}=0, \qquad \left(\forall i\in\overline{1,\,N}\right). \tag{12}$$

For both senses of (10) and (11) the conditions of (12) 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]:

$$\pi^{-}(\sigma_i)=\frac{e^{-\beta_L L(\sigma_i)}}{\sum_{j=1}^{N}e^{-\beta_L L(\sigma_i)}}, \qquad \pi^{+}(\sigma_i)=\frac{e^{\beta_U U(\sigma_i)}}{\sum_{j=1}^{N}e^{\beta_U U(\sigma_i)}}, \tag{13}$$

where β_L and γ_L are corresponding coefficients with the subscript for the objective intelligent air transportation management system functional of (10) having related with the negative sense alternatives σ_i effectiveness estimations; and β_U and γ_U are corresponding coefficients if the subjective attention is drawn to the positive features, in generally speaking terms to the same set of the alternatives.

2.2. Simulation

An example of the calculations results with the data of

$$\sigma=0\ldots100, \qquad \beta_L=0.009, \qquad L_1(\sigma)=10\sigma, \qquad L_2(\sigma)=15\sigma, \qquad L_3(\sigma)=20\sigma, \tag{14}$$

is illustrated in the Figures 1 – 4.

The designations in the Figure 1 are obvious. The diagrams of the losses functions of the Figure 1 are plotted for the calculations with the first equation of (13) and by the last three equations of the data set supposed with (14).

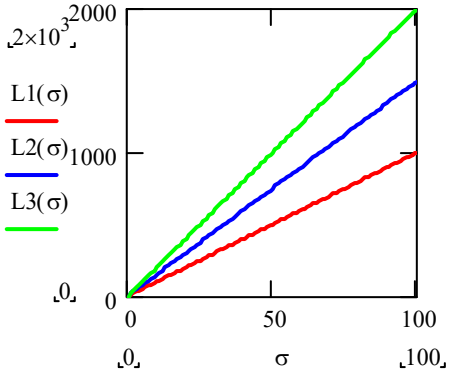


Figure 1: Example figure for losses functions

The subjective preferences functions diagrams (see the Figure 2) are plotted by the corresponding losses calculations with the first equation of (13); the designations in the Figure 2 are also easily identified and readable. The subjective preferences functions normalizing condition expressed with the equation (6) is visible in the Figure 2 too.

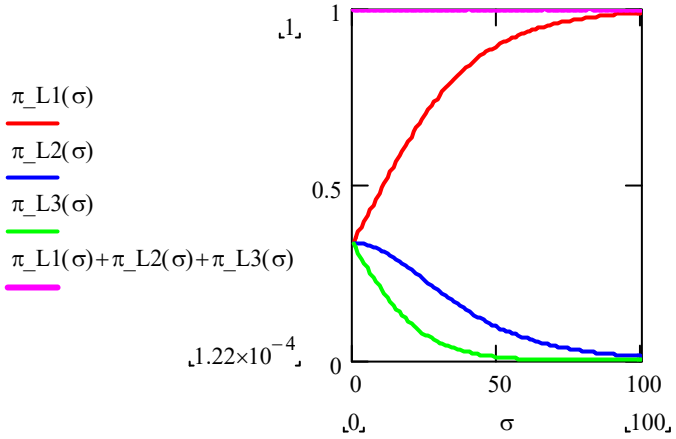


Figure 2: Example figure for preferences functions

Similarly to the diagrams shown in the Figure 2, one can notice (see the Figure 3) the equivalent character of the phase portraits (phase diagrams) of the subjective individuals' preferences functions with respect to the other phase coordinate of the corresponding losses (harmfulness) functions. The designations used in the Figure 3 are the same as those ones of the Figures 1 and 2. Hence, there is no necessity to dwell to much on the phase curves designations; as well as the normalizing condition (6) is illustrated in the Figure 3.

At last in the Figure 4, it is represented the subjective entropy computed by formula (5). The maximal value of the subjective individuals' preferences uncertainty degree (subjective entropy), which equals

$$H_{\pi} = \ln(3), \tag{15}$$

for the modeled three-alternative case, is shown in the Figure 4 as well.

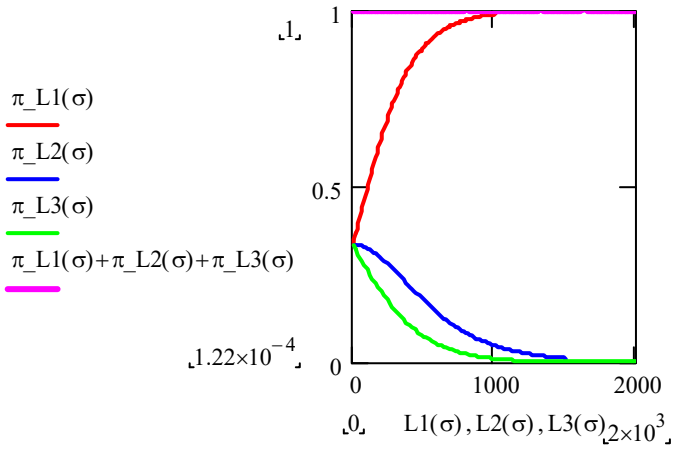


Figure 3: Example figure for preferences over losses functions

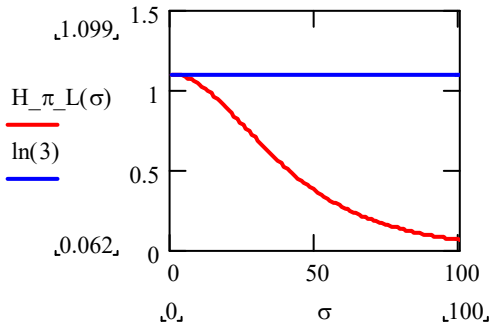


Figure 4: Example figure for preferences functions entropy

The simplicity of the example modeled with the formula (5), first equation of (13), and the data of (14) demonstrates the standard character of the dependencies in the abstract three-alternative case when the maximal preferences are allotted to the lowest losses.

In theoretical aspects the computational results shown in the Figures 1 – 4 are emphasizing the differences to the theoretical contemplations discussed in [12]; therefore, it might be considered a theoretical development in the understanding of the concept.

A bit different example of the calculations results with the differing data of

$$L_1(\sigma) = 10\sigma + 600\,, \qquad L_2(\sigma) = 15\sigma + 200\,, \tag{16}$$

is illustrated in the Figures 5 – 8.

The designations in the Figures 5 – 8 are kept the same as those ones implemented for the diagrams shown in the Figures 1 – 4.

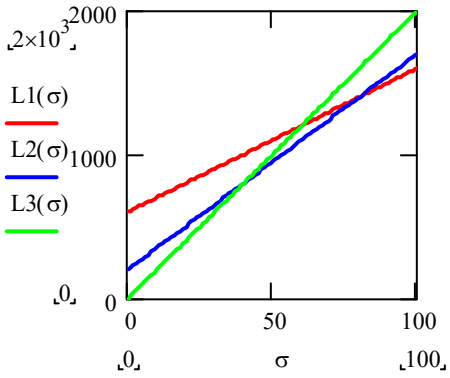


Figure 5: Example figure for modified losses functions

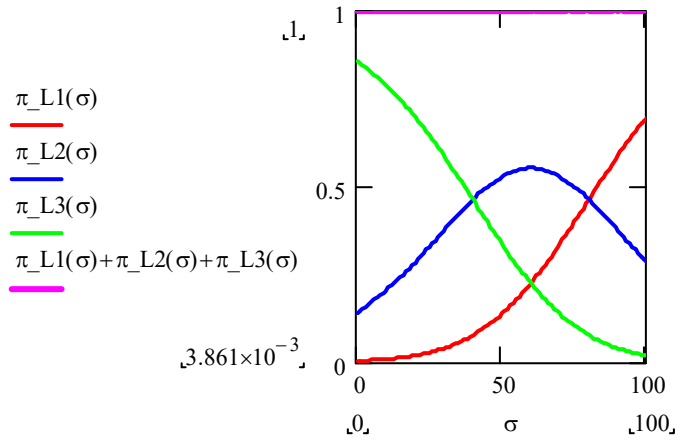


Figure 6: Example figure for modified preferences functions

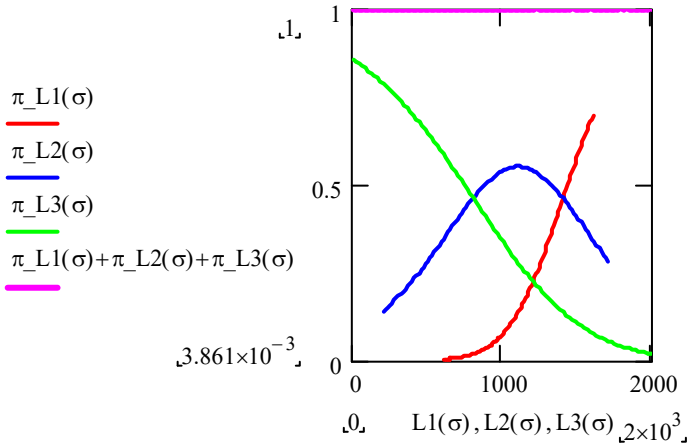


Figure 7: Example figure for modified preferences over losses functions

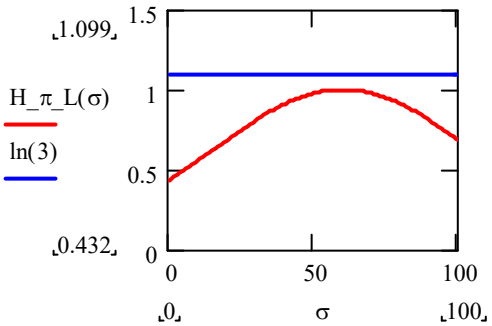


Figure 8: Example figure for modified preferences functions entropy

3. Discussion

The models based upon the second equation of (13) should provide greater subjective individuals' preferences functions values to the greater values of the utility (usefulness) functions. This is logically explainable.

As a whole, the entropy approach modeled with the linear combination for the objective intelligent air transportation management system functional (1) might include a much greater number of constraints than it has been described herewith.

Moreover, the normalization conditions for "one", introduced with relation of (6), are not so obligatory. Sometimes these conditions could be considered as the subsidiary constraints supplementing the specified problem setting for the perceptual easiness.

Endogenous parameters implemented in the presented study entropy paradigm models are presupposed to be predetermined as some external factors impact. Their influence is simplified in principle. Although, such parameters might drastically change the optimal distributions provided the parameters are different and interrelating.

The isoperimetric functions described with the expressions of (2) and (3) might ensure some extremal value (either maximum or minimum) to the objective functional or not because the solutions in the view of the canonical distributions of the subjective individuals' preferences functions (13) are being obtain just based upon the first order (necessary) extremum existence conditions (12). In order to clarify that issue (the deliverance of either maximum or minimum to the objective functional value by the found distributions of the subjective individuals' preferences functions) some additional research should be conducted.

The entropy member, described with the expression (5), of the objective functionals may have the same optimal "point" with the objective functionals themselves and may be not. That depends upon the specific cases modeled in different problem formulations.

As for the integral form of the objective functionals themselves, there are endless numbers of the problems stated in the framework of the calculus of variations that could be solved with the help of the Euler-Lagrange theory. However, the noted with the intensive versus extensive parameters models of equations indicated above as (7) and (8), as well as with the expressions of (9), allow conducting dynamical modeling.

The mean value of the uncertainty for the period of consideration is proposed to be estimated in the ideology of the equation of (8) mathematical statement.

The losses versus utilities dilemmas expressed with (10) and (11), as well as the simplest illustrated simulations results represented in the Figures 1 – 8, show the prospects of the entropy paradigm approach.

Nonetheless, the simplified problem setting discussed herewith is brightly highlighting the most important features: theoretical considerations for the optimization of the intelligent air transportation management system in conditions of the operational multi-alternativeness; and the caused because of the operational multi-alternativeness the operational situation of uncertainty measured with the entropy of the subjective preferences.

4. Conclusion

Computer modelling with the application of the subjective entropy paradigm implemented to an air transportation management intelligent system makes it possible to conduct a reasonably optimal governing activity based upon the theoretically substantiated provisions and results.

Aviation transport management rational organization should take into account the uncertainty of the decision making responsible individuals' preferences related with the alternatives of the air transportation means operation.

Specifically the theoretical developments covered herein are proposed to be used in the field of transport, management, and logistics.

Further research endeavours are envisaged to be taken in the studies dealing with the necessity of the systems intellectual potential optimal growth findings.

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.

Development of digital twin based on a model with fractional-rational uncertainty

Nataliya Pankratova[a], Igor Golinko[a]

[a] *Igor Sikorsky Kyiv Polytechnic Institute, 37, Prosp. Beresteyskiy, Kyiv, 03056, Ukraine*

Abstract

A methodology for developing the digital twin mathematical model for a cyber-physical system in the air heating process form using an electric heater is proposed. At the passive identification expense, the technique gives possibility of synthesis and adaptation of the digital twin discrete mathematical model by the air heating measured data in the operating electric heater. The influence of analytical model uncertain parameters of the electric heater on the calculation's accuracy is considered. The quality criterion for assessing the mathematical model adequacy to a physical dynamic process is proposed. The algorithm of passive identification the mathematical model uncertain parameters is developed, in which the deviations of calculated model variables from measured values in the process of air heating are minimized. A numerical study of the considered method has been carried out. It is shown that the passive identification of the uncertain model parameters belongs to the one-extreme optimization problem. The considered simulation examples prove that the proposed methodology for the digital twin development using MatLAB software is an effective and convenient tool for the rational creation of a digital twin for cyber-physical systems.

Keywords 1

Cyber-physical system, digital twin, mathematical model, identification, uncertain parameters, optimization, electric heater

1. Introduction

Modem IT technology has advanced a great deal and is making it possible to increase the efficiency of industrial production to a great extent. Experts around the world predict the final arrival of the digital age in the near future. The prospect of digitalization looks attractive, progressive and efficient. The basic technological unit of digital production is the cyber-physical system (CPS) [1]. There are different interpretations of CPS concept because these systems are simultaneously located at the different spheres intersection of human activity. The main characteristic of these systems is the interaction between physical and computational processes. The interaction mechanisms are provided by a network structure known as the Industrial Internet of Things (IIoT).

Fig. 1 (*a*) shows a classic production control hierarchy. Here PLC are program logic controllers, SCADA is Supervisory Control and Data Acquisition (dispatching system and management), MES is Manufacturing Execution System (production management system), ERP is Enterprise Resource Planning (enterprise resource management system). However, this structure does not meet the modem production requirements. Today, production management requires Industry 4.0 competences, which are relatively new and largely IT-related. Thus, the classic hierarchical pyramid of production management is being transformed into a system with an IIoT (see Fig. 1, *b*).

Changing the concept of cyber production management dictates the need to develop new methods for the analysis and synthesis of such systems. The main characteristics of CPS are its heterogeneity, uncertainty of different nature, multidisciplinary nature, and provision of functioning guaranteed

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: natalidmp@gmail.com (N. Pankratova); conis@ukr.net (I. Golinko)
ORCID: 0000-0002-6372-5813 (N. Pankratova); 0000-0002-7640-4760 (I. Golinko)

strategy throughout system life cycle [2]. CPS should be able to analyze multidimensional data, taking into account the latent factors of production. Based on this data, it can autonomously solve optimization problems and make the right decisions. A key unit in the CPS maintenance and management process is the digital twins.

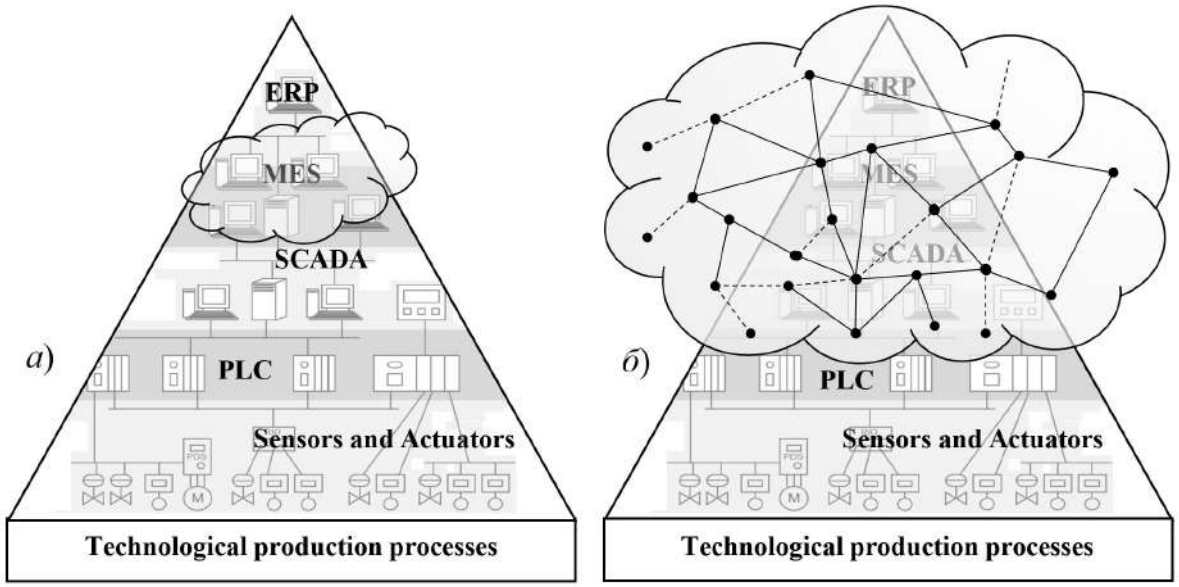


Figure 1: Technological production processes; *a*) classic production management hierarchy; *b*) managing production as a cyber-physical system

The digital twin concept as a basic prerequisite for managing a physical object throughout its life cycle (design, implementation, operation maintenance and disposal) has been proposed by Michael Grieves [3]. The digital twin concept is part of the fourth industrial revolution and is designed to help enterprises detect physical problems faster, predict their results more accurately, and produce better products. The use of a digital twin during the creation phase allows the entire life cycle of the intended object to be modelled and simulated. The digital twin use makes it possible to simulate the entire life cycle of the intended object [4]. Most often, digital twins are created to simulate objects directly related to industrial production or being an important element of technical systems [5]. The first practical definition of a digital twin was given by NASA in an attempt to improve the modelling of physical spacecraft models in 2010 [6].

According to the Industrial Internet Consortium (IIC) [7], a digital twin is a formal digital representation of some asset, process or system that captures attributes and behaviors of that entity suitable for communication, storage, interpretation or processing within a certain context. Researchers point out that digital twin's design is based on the use of simulation methods that provide the most realistic representation of a physical object in the virtual world [8]. Mathematical representation of digital twins can be obtained applying methods: statistical modeling, using machine learning or analytical modeling, which uses a mathematical description the physical processes laws.

Statistical analysis methods can be divided into three groups: regression analysis models, classification models and anomaly detection models [9]. The machine learning method choice depends on the size, quality and data nature, as well as the problems type to be solved. In [10], the authors point out that the analytical model's determinism in engineering is a valuable property that should be exploited in CPS. In [11], CPS development using deterministic models, which have proven to be extremely useful, is discussed. Deterministic mathematical models of CPS are based on differential equations and include synchronous numerical logic and single threaded imperative programs. However, CPS combine these models in such a way that determinism is not preserved.

The development of digital twins is taking place within the concept of IIoT, which encourages developers in the standardization direction. One of the fundamental works on digital twins' standardization is the Industrial Internet Reference Architecture (IIRA) reference model proposed by IIC. The document describes guidelines for the development of systems, solutions and applications

using the Internet of Things in industry and infrastructure solutions. This architecture is abstract, and provides general and consistent definitions for different stakeholders, system decomposition, design patterns, and terms glossary.

The IIRA model for the IIoT identifies at least four stakeholder perspectives: business, use, function, implementation. Each perspective focuses on the functional components of the IIoT, their structure and relationships, the interfaces and interactions between them, and the relationship and interaction of the system with external elements of the environment to support the use and CPS operation. These features are of particular interest to CPS architects, developers and integrators. Based on the IIRA model, digital twin information includes (but is not limited to) a combination of the following categories [7]: physical model and data; analytical model and data; temporal variable archives; transactional data; master data; visual models and calculations. Based on the IIRA model, digital twins are mainly created for commercial use using simulation and optimization techniques, databases, etc. to implement rational real-time production management and support strategic and operational decision making. The concept of a digital twin has a multifaceted architecture and correspondingly complex mathematical support for its implementation. The development of a digital twin mathematical model, which after machine learning or identification gives the ability to follow and predict the physical process behavior, is the subject of many scientific discussions. In this paper the approach to develop a mathematical model of a digital twin, taking into account its fractional-rational uncertainty is proposed.

2. Research problem statement

The aim of the publication is to develop digital twin mathematical model of CPS for manufacturing plants, taking into account the parametric uncertainties of the physical process mathematical description. The application of the mathematical model synthesis methodology for the electric heater digital twin is given.

3. Electric heater model uncertainty analysis

Let's consider a mathematical model of the air heating process on an electric heater

$$\begin{cases} T_E \dfrac{d\,\Delta\theta_E}{dt} + \Delta\theta_E = k_0\,\Delta N_E + k_1\,\Delta\theta_A, \\ T_A \dfrac{d\,\Delta\theta_A}{dt} + \Delta\theta_A = k_2\,\Delta\theta_E + k_3\,\Delta\theta_{A0} + k_4\,\Delta G_A, \\ T_d \dfrac{d\,\Delta d_A}{dt} + \Delta d_A = k_5\,\Delta d_{A0} + k_6\,\Delta G_A; \end{cases} \quad (1)$$

here $T_E = \dfrac{c_E M_E}{K_E}$, $K_E = \alpha_0 F_0$, $k_0 = \dfrac{1}{K_E}$, $k_1 = 1$; $T_A = \dfrac{c_A M_A}{K_A}$, $K_A = c_A G_A + \alpha_0 F_0$, $k_2 = \dfrac{\alpha_0 F_0}{K_A}$, $k_3 = 1 - k_2$, $k_4 = \dfrac{c_A(\theta_{A0} - \theta_A)}{K_A}$; $T_d = \dfrac{\omega V_A}{G_A}$, $k_5 = 1$, $k_6 = \dfrac{d_{A0} - d_A}{G_A}$.

Mathematical model (1) and its thermophysical parameters are considered in detail in work [12]. On the analysis basis of mathematical model parameters numerical values, it can be stated that thermophysical values of material flows and structural materials of electric heater are determined with high accuracy from reference books on thermophysical properties of substances and materials [13].

However, the heat transfer coefficient α_0 depends on many factors. Its numerical value can vary several times depending on: the heating element temperature θ_E; temperature θ_A, humidity d_A and air expense G_A; heat exchange surface design features and other factors. This parameter is the subject of thermal engineering research. Numerous papers have been published on methods for calculating the heat transfer coefficient, which are based on experimental studies and similarity theory [13]. The parameters T_E, T_A, k_0, $k_2 \ldots k_4$ of model (1) depend on the heat transfer coefficient α_0.

The linear properties of the mathematical model must also be taken into account. The linearization of model (1) was carried out in the basic static mode region of the heater. Therefore, the variable parameters include coefficients characterizing the basic static mode, namely: air flow rate G_A; air temperature θ_{A0} and humidity d_{A0} at the input to the electric heater; air temperature θ_A and humidity d_A at the output to the electric heater. The basic static mode variables affect parameters T_A, T_d, $k_2 \ldots k_4$, k_6 of model (1). In Fig. 2 the classification of mathematical model parameters (1) is proposed. Formally for model (1) there are six changing physical quantities α_0, G_A, θ_{A0}, θ_A, d_{A0}, d_A on which all coefficients of mathematical model (1) depend.

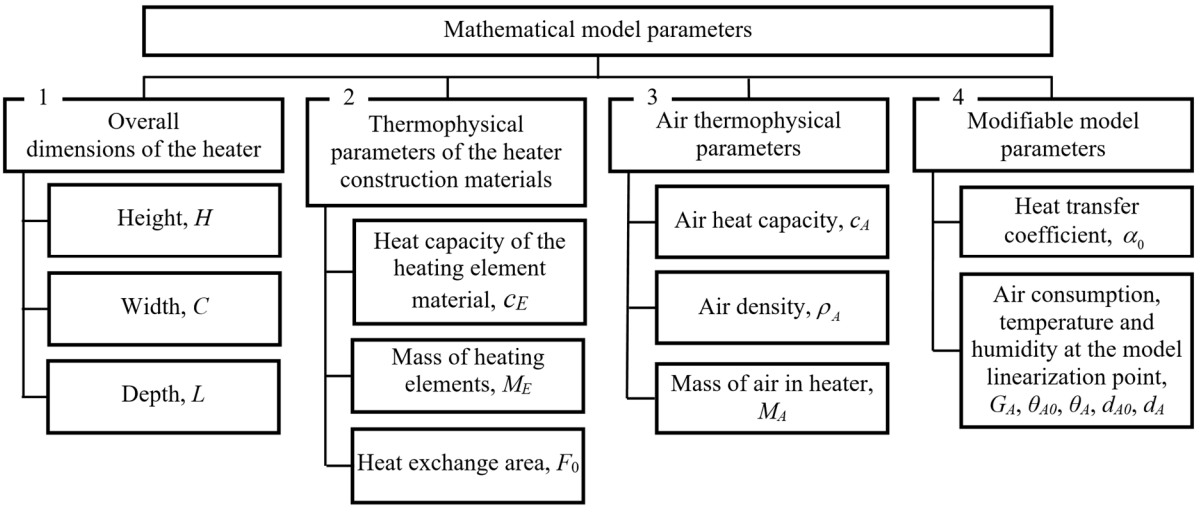


Figure 2: Classification of the electric heater model parameters

It is convenient to use Laplace transforms to find a solution to (1). Consider an implementation of model (1) in the form of transfer functions:

$$\begin{cases} \Delta\theta_A = \dfrac{1}{a_2 p^2 + a_1 p + 1}\left[\left(b_1 p + b_0\right)\Delta\theta_{A0} + \left(b_3 p + b_2\right)\Delta G_A + b_4 \Delta N_E\right], \\ \Delta d_A = \dfrac{1}{T_d p + 1}\left[k_5 \Delta d_{A0} + k_6 \Delta G_A\right]; \end{cases} \tag{2}$$

here $a_2 = \dfrac{T_E T_A}{1 - k_1 k_2}$, $a_1 = \dfrac{T_E + T_A}{1 - k_1 k_2}$; $b_0 = \dfrac{k_3}{1 - k_1 k_2}$, $b_1 = \dfrac{k_3 T_E}{1 - k_1 k_2}$, $b_2 = \dfrac{k_4}{1 - k_1 k_2}$, $b_3 = \dfrac{k_4 T_E}{1 - k_1 k_2}$,

$b_4 = \dfrac{k_0 k_2}{1 - k_1 k_2}$; p is the Laplace operator.

The mathematical model (2) can be represented by multidimensional model in the Laplace domain:

$$\mathbf{Y}(p) = \mathbf{W}(p)\mathbf{X}(p); \tag{3}$$

here $\mathbf{Y} = \begin{bmatrix} \Delta\theta_A \\ \Delta d_A \end{bmatrix}$, $\mathbf{W} = \begin{bmatrix} W_{11} & 0 & W_{13} & W_{14} \\ 0 & W_{22} & W_{23} & 0 \end{bmatrix}$, $\mathbf{X} = \left[\Delta\theta_{A0}\ \Delta d_{A0}\ \Delta G_A\ \Delta N_E\right]^{\mathrm{T}}$;

$W_{11} = \dfrac{b_1 p + b_0}{a_2 p^2 + a_1 p + 1}$, $W_{13} = \dfrac{b_3 p + b_2}{a_2 p^2 + a_1 p + 1}$, $W_{14} = \dfrac{b_4}{a_2 p^2 + a_1 p + 1}$, $W_{22} = \dfrac{k_5}{T_d p + 1}$, $W_{23} = \dfrac{k_6}{T_d p + 1}$.

Applying the inverse Laplace transform, it is easy to find an analytical solution (2), (3) for the influence channels.

Let's analyze variable parameters influence of mathematical model (3) on electric heater HE 36/2 dynamic properties, simulation of which is considered in [12]. Let's assume that the electric heater basic static mode parameters θ_{A0}, θ_A, d_{A0}, d_A are determined using CFS sensors readings. Suppose, for model (3), due to reduction of heated air flow rate from $G_A = 0.43\ kg/s$ to $\overline{G}_A = 0.2\ kg/s$ (the new

nominal operating mode of the heater, all other conditions being equal) the heat exchange coefficient is changed from $\alpha_0 = 161 \frac{W}{m^2\,{}^oC}$ to $\overline{\alpha}_0 = 100 \frac{W}{m^2\,{}^oC}$. At the same time, the model (3) matrix transfer function parameters will change

from $$\mathbf{W}(p) = \begin{bmatrix} \frac{5.6p+1}{2.4p^2+6.7p+1} & 0 & -\frac{52.1p+9.3}{2.4p^2+6.7p+1} & \frac{0.002}{2.4p^2+6.7p+1} \\ 0 & \frac{1}{0.42p+1} & 0 & 0 \end{bmatrix} \quad (4)$$

to $$\overline{\mathbf{W}}(p) = \begin{bmatrix} \frac{9p+1}{8.2p^2+11.3p+1} & 0 & -\frac{180.4p+20}{8.2p^2+11.3p+1} & \frac{0.005}{8.2p^2+11.3p+1} \\ 0 & \frac{1}{0.91p+1} & 0 & 0 \end{bmatrix}. \quad (5)$$

The numerical values of the transfer matrices (4) and (5) differ significantly. Fig. 3 shows the transient simulation results. In this figure and further on the symbols are used: the output vector $\mathbf{Y}$

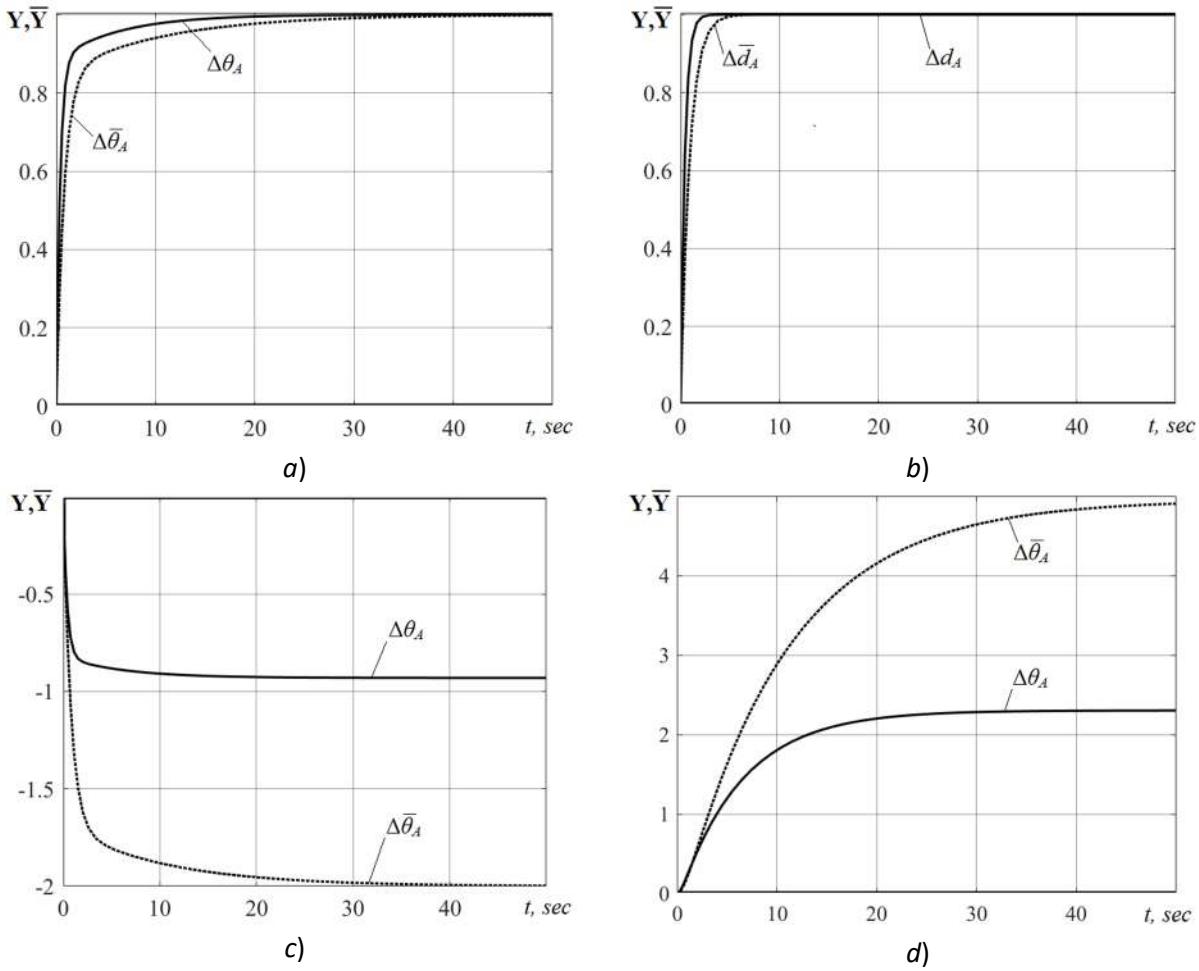


Figure 3: The heat transfer coefficient α_0 influence on the dynamic properties of the heater model HE 36/2 through the action channels $\mathbf{X} \to \mathbf{Y}$ (the graphs shows output variables received increments as the result of changing α_0);

a) $\Delta\theta_{A0} \to \mathbf{Y}$, $\Delta\theta_{A0} = 1\,{}^oC$; *b*) $\Delta d_{A0} \to \mathbf{Y}$, $\Delta d_{A0} = 1$ g/kg;

c) $\Delta G_{A0} \to \mathbf{Y}$, $\Delta G_{A0} = 0.1$ kg/s; *d*) $\Delta N_E \to \mathbf{Y}$, $\Delta N_E = 1$ kW

for the reference mathematical model with the transfer matrix (4) where the initial values of parameters α_0 and G_A were used; the output vector $\overline{\mathbf{Y}}$ for the model with the transfer matrix (5) where the new values of parameters $\overline{\alpha}_0$ and $\overline{G}_A$ were used. The output vectors $\mathbf{Y}$ and $\overline{\mathbf{Y}}$ contain the variables: $\Delta\theta_A$ and $\Delta\overline{\theta}_A$ are heated air temperature; Δd_A and $\Delta\overline{d}_A$ are heated air moisture content. In transients' simulation for HE 36/2 electric heater a control action $\mathbf{X}$ was used by successive stepwise changes in the input variables. The input vector $\mathbf{X}$ contains the variables: $\Delta\theta_{A0}$, Δd_{A0} are temperature and humidity of electric heater input air; ΔG_A is air flow through the electric heater; ΔN_E is electric power supplied to the electric heater elements.

The heat transfer coefficient α_0 affects the numerical values of the matrix transfer function of the model (3). The dynamic characteristics of the air heating process are significantly influenced by this parameter, as demonstrated in the simulation graphs (see Fig. 3). For these reasons, specialists in the field of heat exchange processes simulation should determine this parameter quite accurately, since its numerical value significantly affects the investigated process properties.

In paper [14] mathematical models uncertainties classification in the control of organizational and technological objects is considered. In general terms, the electric heater mathematical model can be represented by the input-output relation

$$\mathbf{Y}(p,q) = \mathbf{W}(p,q)\mathbf{X}(p), \tag{6}$$

here $\mathbf{W}(p,q) = \left[\mathbf{A}(p) + \delta\mathbf{A}(p,q)\right] / \left[\mathbf{B}(p) + \delta\mathbf{B}(p,q)\right]$ is transfer matrix, $\delta\mathbf{A}(p,q)$ and $\delta\mathbf{B}(p,q)$ are numerator and denominator of perturbation, p is Laplace operator, q is vector of uncertain parameters. According to the proposed classification, the electric heater mathematical model has fractional-rational uncertainty, which is proposed to be identified by numerical methods.

4. Identification of mathematical model parameters

The electric heater mathematical model (3) is developed on the basis of theoretical laws for a specific physical process. Therefore, the analytical model requires specification of some parameters. Formally for model (3) there are six changing parameters α_0, G_A, θ_{A0}, θ_A, d_{A0}, d_A (see Fig. 2). In the process of identification, it is necessary to determine two parameters α_0, G_A, parameters of the basic static mode θ_{A0}, θ_A, d_{A0}, d_{A} can be estimated, using CFS sensors readings or with sufficient accuracy these parameters can be determined from the data sheet of the electric heater.

Model (3) parameters identification is proposed to be performed in the passive experiment operation by measured signals values of electric heater input vectors $\mathbf{X}$ and output vectors $\mathbf{Y}$. As the identification criterion we use the least squares criterion, which is defined as the measured values error square of the physical process $\mathbf{Y}$ and the estimates of the identified model (3) output vector $\overline{\mathbf{Y}}$ at the same input action $\mathbf{X}$

$$I = M\left\{\int_{t_0}^{t_0+t_f} \left(\mathbf{Y} - \overline{\mathbf{Y}}\right)^{\mathrm{T}} \mathbf{Q}\left(\mathbf{Y} - \overline{\mathbf{Y}}\right) dt\right\} \to \min, \tag{7}$$

here t_0 is the initial trend time, t_f is the trend duration, $\mathbf{Q}$ is the unit square matrix, $\mathbf{T}$ is the matrix transpose operator, M is the mathematical expectation operator, which takes into account industrial perturbations when measuring variables by sensors. The general structural diagram of parameters identification of the electric heater mathematical model is shown in Fig. 4.

It is recommended to use numerical zero-order optimization methods to identify the mathematical model parameters [15], since the search function is not set analytically and is calculated directly during the search algorithm implementation. Given the substantial computational resources to implement numerical methods, passive identification can be implemented at the enterprise management middle level within a decision support system (DSS) [16].

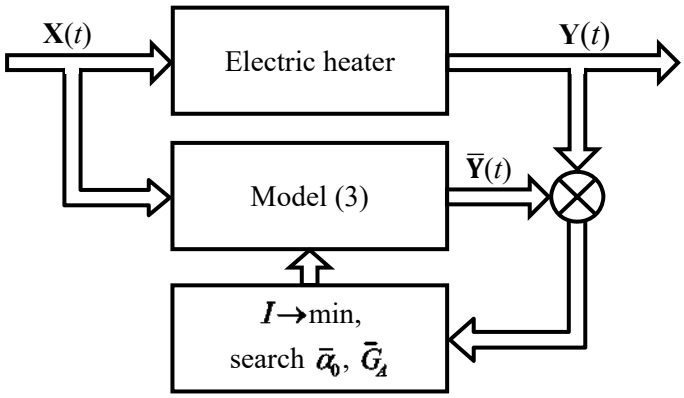


Figure 4: Schematic diagram of the heater parameter passive identification

The algorithm for identifying the parameters of a mathematical model consists of steps.

1. The CFS sensors monitor the input vector **X**(*t*) and the output state **Y**(*t*) of the heater in real time and the data is processed in the DSS, thus forming the time base of the physical process trends.
2. The output state $\overline{\mathbf{Y}}(t)$ of the identifiable model (3) with the parameters initial values $\bar{\alpha}_0$, $\bar{G}_A$ is estimated from the time trends of the input action **X**(*t*) on the electric heater.
3. Based on the values of the output **Y**(*t*) vectors and the identifiable $\overline{\mathbf{Y}}(t)$ states, the identification quality criterion (7) is determined.
4. According to criterion (7), the parameters $\bar{\alpha}_0$, $\bar{G}_A$ of the identified model (3) are optimized using numerical optimization methods.
5. If the minimum of criterion (7) is found, move on to the development of the digital twin mathematical model, else move on to step 2 and continue the process of identifying model parameters.

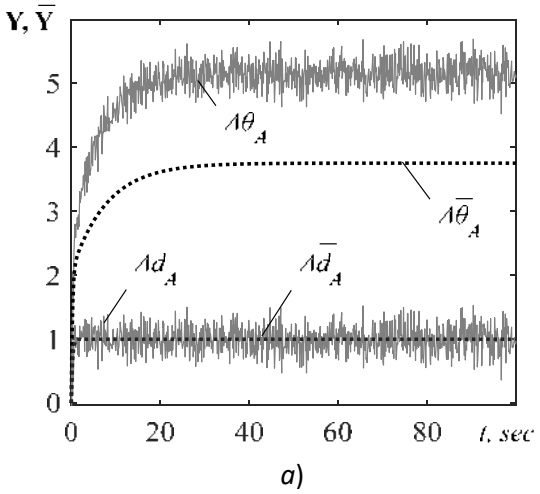
During the identification process, the optimization method (7) can have its own specific features that need to be taken into account. For qualitative identification, the time interval t_f must be several times longer than the duration of transients in the heater. Also, it is necessary to set the parameter variation limits $\bar{\alpha}_0$ and $\bar{G}_A$, based on the physical feasibility of the duct heater model.

The proposed identification algorithm was implemented using MatLAB software package. The reference model with transfer matrix (4) was used to form time trends **X**(*t*) and **Y**(*t*) of the functioning electric heater. The random signal with amplitude ±0.2 was added to the reference output variables in order to simulate production disturbances in DSS measurement channels. In the identified model the initial values of parameters $\bar{\alpha}_0$, $\bar{G}_A$ differed significantly from the reference values α_0, G_A. The MatLAB function fminsearch(...) was used for parametric identification of $\bar{\alpha}_0$ and $\bar{G}_A$ with the Nelder-Meade simplex optimization method. The main results of the numerical study are presented in Fig. 5. and Fig. 6.

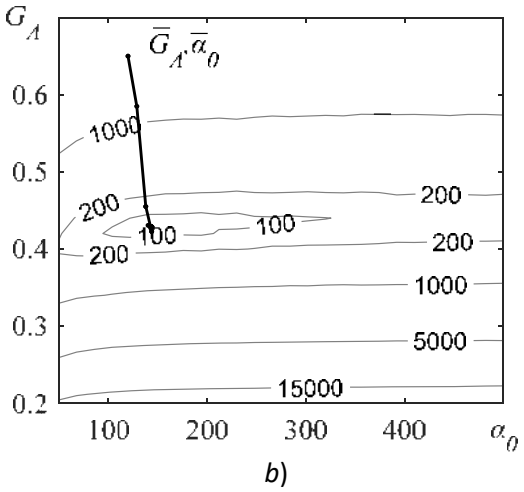
Fig. 5 shows the case where the initial values of the identification parameters are as follows: $\alpha_0 = 161$, $G_A = 0.43$; $\bar{\alpha}_0 = 120$, $\bar{G}_A = 0.65$. With a step change in the input action vector $\mathbf{X}(t) = [1,\ 1,\ -0.2,\ 1]^{\mathrm{T}}$, produces the transients for the output of the reference $\mathbf{Y}(t)$ and the identifiable $\overline{\mathbf{Y}}(t)$ model, which are shown in Fig. 5 (*a*).

The difference between the output values of vectors $\mathbf{Y}(t)$ and $\overline{\mathbf{Y}}(t)$ is significant. In Fig. 5 (b) shows the surface isolines of criterion (7) and its minimization trajectory. From Fig. 5 (*b*) shows that the quality criterion has one extremum in the area of identification parameters. Identification parameters $\bar{\alpha}_0 = 143.7$, $\bar{G}_A = 0.428$ were determined using the above algorithm. The numerical value $\bar{\alpha}_0$ was found with low accuracy, while $\bar{G}_A$ was determined quite accurately. Identification results obtained can be explained by weak sensitivity of criterion (7) to parameter α_0, that can be seen from curves of isolines in Fig. 5 (*b*). Fig. 5 (*c*) shows the output time characteristics of the reference $\mathbf{Y}(t)$

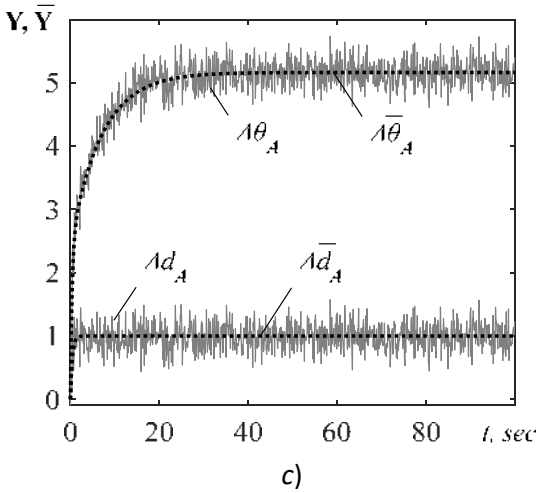
and the identified $\overline{\mathbf{Y}}(t)$ model after identification. In the graph, the variables of the identified model output vector $\overline{\mathbf{Y}}(t)$ are in the region of the reference model vector $\mathbf{Y}(t)$ with random perturbation. It can be concluded that the proposed identification algorithm has good convergence in the stepped input influences case.



a)



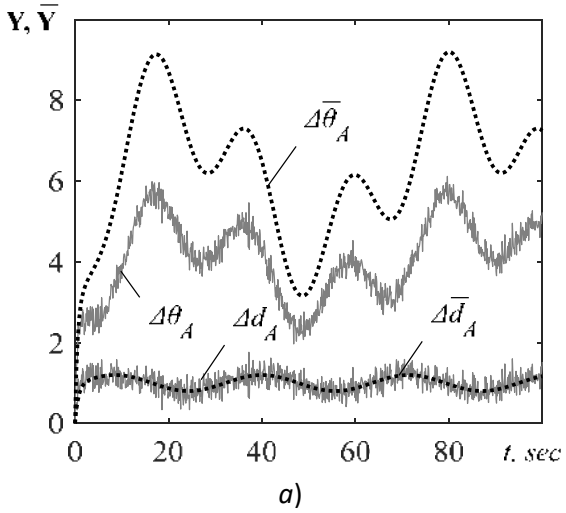
b)



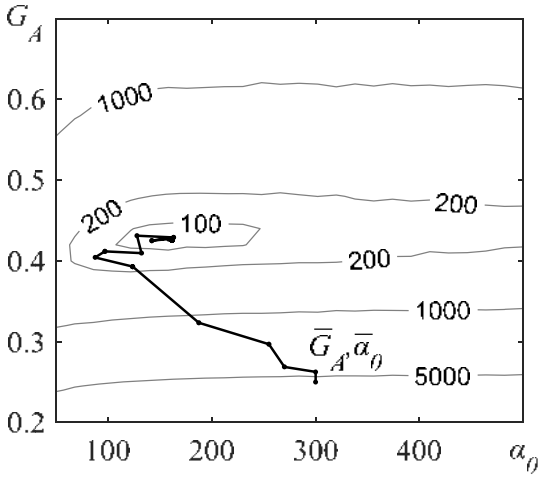
c)

Figure 5: Identification of parameters $\bar{\alpha}_0$ and $\bar{G}_A$ with the input $\mathbf{X}(t)=[1,\ \ 1,\ \ -0.2,\ \ 1]^{\mathrm{T}}$;
a) Transients modelling before identification;
b) Optimization trajectory $\bar{\alpha}_0$ and $\bar{G}_A$ according to criterion (7);
c) Transients modelling after identification $\bar{\alpha}_0$ and $\bar{G}_A$

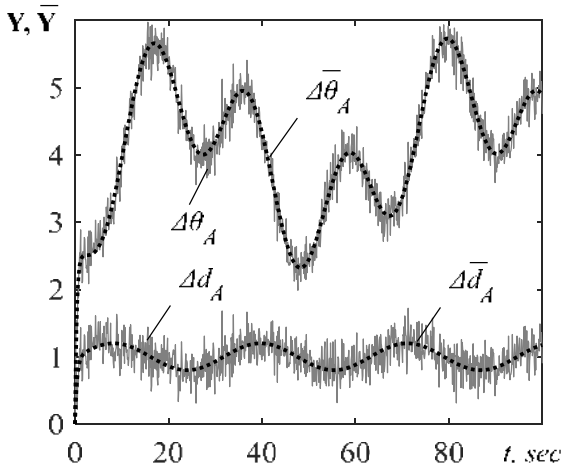
However, under real-world conditions, the input can be of any shape. Consider the more complex case where the input signal has a harmonic component. In this case, the input action $\mathbf{X}(t)=[1,\ \ 1+0.2\sin(0.2t),\ \ -0.2+0.1\sin(0.3t),\ \ 0.5+0.5\sin(0.1t)]^{\mathrm{T}}$ is applied to both models. Fig. 6 (*a*) shows the simulation case when the parameters of the identified model $\bar{\alpha}_0=300$, $\bar{G}_A=0.25$ differ significantly from the reference ones $\alpha_0=161$, $G_A=0.43$. Fig. 6 (*b*) shows surface isolines of criterion (7) and its minimization trajectory. In the process of identification parameters values $\bar{\alpha}_0=161.9$, $\bar{G}_A=0.426$ are optimized. Numerical values of identification parameters are sufficiently close to reference ones. Fig. 6 (*c*) shows temporal characteristics of the reference vector $\mathbf{Y}(t)$ and identifiable $\overline{\mathbf{Y}}(t)$ model after identification under harmonic change X(t). It can be concluded from the simulation results that the proposed passive identification algorithm has good convergence in the harmonic component presence the vector $\mathbf{X}(t)$ and the random noise presence.



a)



b)



c)

Figure 6: Parametric identification $\bar{\alpha}_0$ and $\bar{G}_A$ under harmonic variation of the input vector $\mathbf{X}(t)$;
a) Transients modelling before identification;
b) Optimization trajectory $\bar{\alpha}_0$ and $\bar{G}_A$ according to criterion (7);
c) Transients modelling after identification $\bar{\alpha}_0$ and $\bar{G}_A$

5. Application of mathematical model for the electric heater digital twin

The development of the digital twin mathematical model will be based on the continuous model (3), which previously passed the uncertain parameters identification stage using control samples of trends **X**(*t*) and **Y**(*t*) for the operating electric heater. The digital twin model must reflect the physical process dynamics in real time. To ensure this condition, the mathematical model must be discrete, with the sampling time ensuring the information distribution over CFS network.

The transition from the continuous model (3) to the discrete analogue can be obtained by using z-transformations [17]

$$\mathbf{Y}(z) = \mathbf{W}(z)\mathbf{X}(z), \tag{8}$$

here $\mathbf{W}(z) = \frac{z-1}{z} Z\left\{\frac{\mathbf{W}(p)}{p}\right\}$ is the discrete transfer matrix of the multidimensional system, where the multiplier $\frac{z-1}{z}$ indicates the presence of zero-order extrapolator for fixing the signal between quantization moments T_{KV}.

Thus, the application of mathematical model for s digital twin for electric heater consists of steps.

1. The uncertain parameters ($\bar{\alpha}_0$ and $\bar{G}_A$) identification of electric heater mathematical model (3) by the considered algorithm.
2. Transition from continuous model (3) to discrete model (8), which is digital twin.

3. If during operation, the digital twin accuracy has deteriorated due to the non-stationarity of the physical process, it is necessary to go to step 1 to determine the model parameters.

In accordance with the proposed methodology, the digital twin mathematical model simulation of the electric heater was carried out using the MatLAB software package. The reference model (3) with transfer matrix (4) was used as a basic model. The function c2d(...) of the MatLAB software package was used to calculate the matrix $\mathbf{W}(z)$ of digital twin (8). The simulation results are shown in Fig. 7.

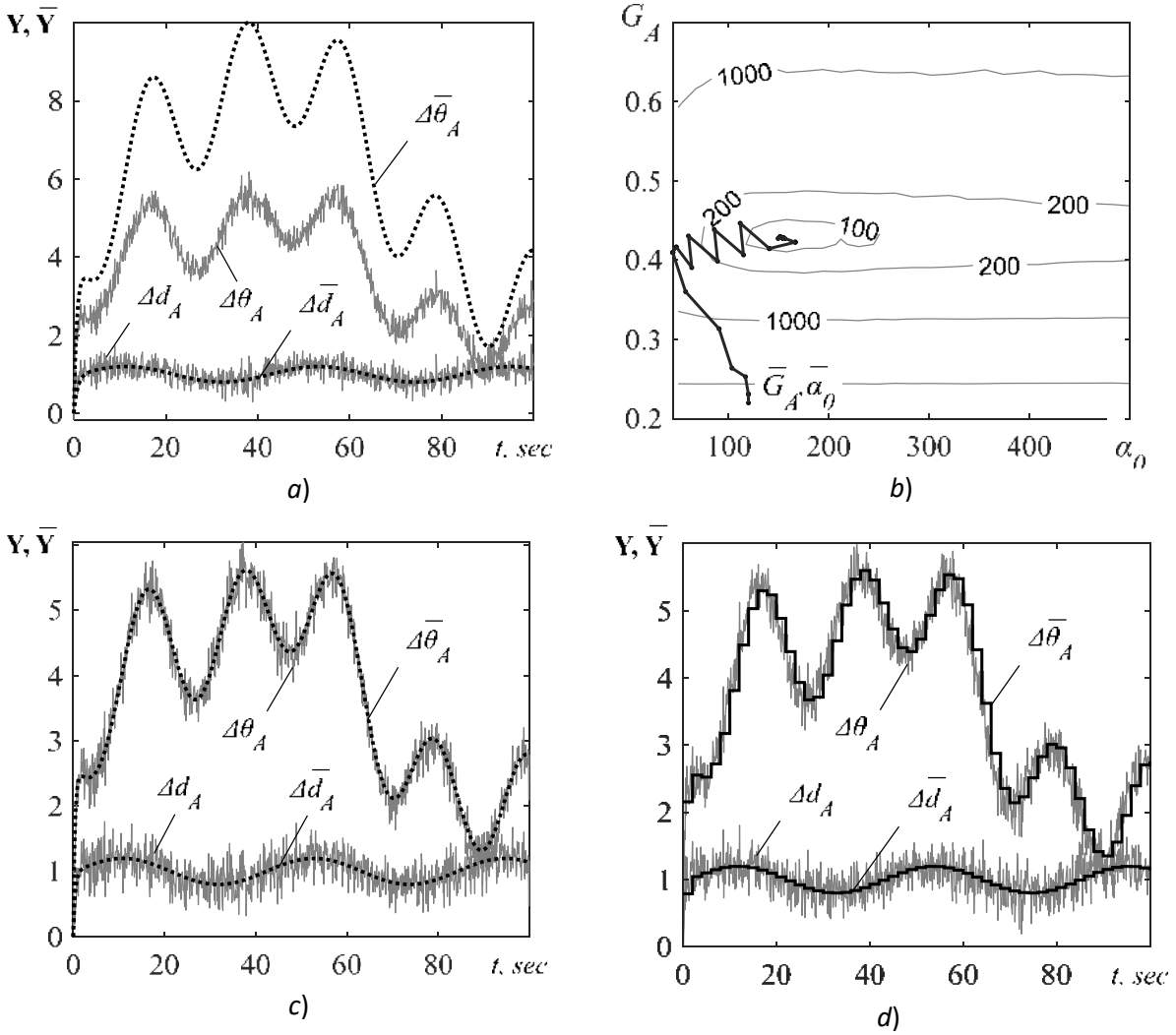


Figure 7: Modeling the stages of digital twin synthesis for electric heater;
a) Transients before identification; *b*) Optimization trajectory $\bar{\alpha}_0$ and $\bar{G}_A$;
c) Transients after identification $\bar{\alpha}_0$ and $\bar{G}_A$; *d*) Transients for model (3) and digital twin (8)

In Fig. 7 (*a*) simulates the case with initial conditions for the reference model $\alpha_0 = 161$, $G_A = 0.43$ and the identifiable model $\bar{\alpha}_0 = 120$, $\bar{G}_A = 0.22$ and at the input step action $\mathbf{X}(t) = \left[1 + 0.5\sin(0.15t) \quad 1 + 0.2\sin(0.15t) \quad -0.2 + 0.1\sin(0.3t) \quad 0.2 + 0.8\sin(0.05t)\right]^{\mathrm{T}}$. In Fig. 7 (*b*) shows the surface isolines of criterion (7) and parameters minimization trajectory $\bar{\alpha}_0$, $\bar{G}_A$, resulting in $\bar{\alpha}_0 = 154$, $\bar{G}_A = 0.428$. From these parameters the model transfer matrix being identified is calculated

$$\overline{\mathbf{W}}(p)=\begin{bmatrix} \dfrac{5.9p+1}{2.5p^2+6.9p+1} & 0 & -\dfrac{54.7p+9.3}{2.5p^2+6.9p+1} & \dfrac{0.0023}{2.5p^2+6.9p+1} \\ 0 & \dfrac{1}{0.426p+1} & 0 & 0 \end{bmatrix},$$

its numerical values are close to the reference transfer matrix (4). Next, using MatLAB function c2d(...), the discrete transfer matrix (8) for the sampling period is found $T_{KV}=2$

$$\overline{\mathbf{W}}(z)=\begin{bmatrix} \dfrac{0.9127z-0.6516}{z^2-0.742z+0.004} & 0 & \dfrac{-8.539z+6.09}{z^2-0.742z+0.004} & \dfrac{0.0005z+0.0001}{z^2-0.742z+0.004} \\ 0 & \dfrac{0.991}{z-0.009} & 0 & 0 \end{bmatrix}.$$

Fig. 7 (*d*) shows the time characteristics of the reference model output vector $\mathbf{Y}$ and the digital twin $\overline{\mathbf{Y}}$. Based on the simulation results, it can be concluded that the proposed methodology makes it possible to implement sufficiently exact the digital twin of air heating.

A distinctive feature of the proposed digital synthesis technique is the model parameters identification that are imprecisely defined at the model development stage and are refined in the passive identification process. As a rule, in modern identification methods, the model structure is specified and all parameters are determined from its dynamics [17, 18]. In the classical case, more software resources are required for identification, and the identification result may be unsatisfactory if the model structure is incorrect. In the proposed methodology, the model structure is known, for which only uncertain parameters are identified and not the model as a whole.

6. Conclusions

The introduction of real-time industrial control systems leads to the complex mathematical models use with uncertain parameters and the complexity of their identification tasks. As an example, the methodology of analytical model passive identification of electric heater with subsequent digital twin synthesis for CFS is considered. For the considered model changing parameters analysis is made and their influence on modelling results is demonstrated. An algorithm for passive identification of the mathematical model uncertain parameters is proposed that uses criterion (7) to numerically optimize the uncertain model coefficients values using the accumulated trend base of the physical process.

Several examples of parameter identification $\bar{\alpha}_0$ and $\bar{G}_A$ for the analytical model (3) are demonstrated. The examples show that the uncertain model coefficients identification should be classified as a one-extremes optimization problem. Application of the mathematical model for the electric heater digital twin CPS is proposed and numerically investigated. The simulation results confirmed the effectiveness of the considered digital twin design methodology using MatLAB application package. In the future, using available data trends of reference model (3), it is supposed to synthesize electric heater digital twin based on ARCS model and analyze the obtained results.

The use of digital twins in CPS makes it possible to identify process bottlenecks, improve product quality, and reduce the risks of abnormal operation throughout its equipment lifecycle. Digital twins can be used to optimize equipment modes, predict and detect faults, and find-modifications to the structure of a real physical system based on its observed effects in the future. This approach provides a highly accurate assessment of a plant's production capacity when drawing up a production program.

7. References

[1] D. Jone, C. Snider, A. Nassehi, J. Yon, B. Hicks, Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology, (2020) 36-52. doi:10.1016/j.cirpj.2020.02.002

[2] N.D. Pankratova, Creation of Physical Models for Cyber-Physical, Systems Lecture Notes in Networks and Systems (2020) 68-77. doi:10.1007/978-3-030-34983-7.

[3] M. Grieves, Completing the Cycle: Using PLM Information in the Sales and Service Functions [Slides], SME Management Forum, Troy, MI. (2002).
[4] M. Grieves, Can the digital twin transform manufacturing, World Economic Forum, 2015. https://www.weforum.org/agenda/2015/10/can-the-digital-twin-transform-manufacturing/?DAG=3&gclid=Cj0KCQjwwtWgBhDhARIsAEMcxeCFyfAQIbgE1WIoIQUT1wAeywKBphlxnxPUaNBTN5mtDPykV7GLwlcaAho2EALw_wcB
[5] V. Slyusar, The concept of networked distributed engine control system of future air vehicles, Proceedings of AVT-357 STO NATO Workshop on Technologies for future distributed engine control systems (2021) 1-12. doi:10.14339/STO-MP-AVT-357
[6] E. Negria, L. Fumagalli, M. Macchia, A review of the roles of Digital Twin in CPS-based production systems, 27th International Conference on Flexible Automation and Intelligent Manufacturing (2017) 939-948. doi:10.1016/j.promfg.2017.07.198
[7] Digital Twins for Industrial Applications, An Industrial Internet Consortium White Paper, 2020. https://www.iiconsortium.org/pdf/IIC_Digital_Twins_Industrial_Apps_White_Paper_2020-02-18.pdf
[8] B.A. Talkhestani, T. Jung, B. Lindemann, N. Sahlab, N. Jazdi, W. Schloegl, M. Weyrich, An architecture of an intelligent digital twin in a cyber-physical production system. Automatisierungstechnik, (2019) 762–782. doi:10.1515/auto-2019-0039.
[9] P.I. Bidyuk, O.L. Tymoshchuk, A.E. Kovalenko, L.O. Korshevniuk, Decision support systems and methods. Kyiv, Igor Sikorsky Kyiv Polytechnic Institute, 2022.
[10] E.A. Lee, Fundamental Limits of Cyber-Physical Systems Modeling. ACM Transactions on Cyber-Physical Systems. (2016) 1-26. doi:10.1145/2912149
[11] E.A. Lee, The Past, Present and Future of Cyber-Physical Systems: A Focus on Models, Sensors (2015) 4837–4869. doi:10.3390/s150304837
[12] N. Pankratova, I. Golinko, Electric heater mathematical model for cyber-physical systems, System research and information technologies (2021) 7–17. doi:10.20535/SRIT.2308-8893.2021.2.01
[13] Y.M. Kornienko, Yu.Yu. Lukach, I.O. Mikulonok, V.L. Rakytskyi, G.L. Ryabtsev, Processes and equipment of chemical technology, NTUU "KPI", Kyiv, 2012.
[14] T.O. Prokopenko, Classification of uncertainties in the management of organizational and technological objects, Technological audit and production reserves (2014) 23–25. doi:10.15587/2312-8372.2014.30336
[15] S.D. Shtovba, Optimization methods in the MatLab environment, VDTU, Vinnytsia, 2001.
[16] N. Pankratova, P. Bidyuk, I. Golinko, Decision support system for microclimate control at large industrial enterprises, CMIS-2020: Computer Modeling and Intelligent Systems (2020) 489–498. https://ceur-ws.org/Vol-2608/paper37.pdf
[17] V.M. Dubovoi, Identification and modeling of technological objects and control systems, VNTU, Vinnytsia, 2012.
[18] P.I. Bidyuk, O.M. Trofymchuk, A.V. Fedorov, Information system of decision-making support for forecasting financial-economic processes based on structural-parameter adaptation of models, KPI Science News (2011) 42-53.

On Efficient Single-Core Execution of Agent-Based Epidemiological Models with Contact-Tracing Transmission

Vladyslav Sarnatskyi[a] *and* Igor Baklan[a]

[a] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Prosp. Peremohy, Kyiv, 03056, Ukraine*

Abstract

In this work, we present our research on efficiency of single-core execution of agent-based epidemiological models with contact-tracing transmission. We performed an analysis of existing epidemiological agent-based modeling tools from the perspective of their performance, which reflects computational time needed to perform a single simulation step of a model with fixed number of agents.

We developed several simulation algorithms, based on different model types, and showed some optimizations to maximize the performance of them.

We designed a metric to compare simulation step execution times on a single CPU core and used to estimate a performance of underlying simulation algorithms in the existing methods with developed one. Even though, as it will be discussed below, it is very difficult to estimate an execution time of some algorithm without actual access to it, we present the results of our method on both modern and old CPU.

Keywords 1

Modeling, epidemiology, agent-based models, individual-based models, IBMs, infectious disease spread modeling, performance, single-thread, optimization

1. Introduction

Global epidemilogical events can make significant damage to global economy [1]. In order to reduce both epidemiological and economic impact, active measures must be considered. They include, but not limited to full or partial quarantine, obligatory face mask regime, vaccinations etc. However, even though some measures like full quarantine can significantly lower disease spread rate by flattening the curve of new cases, economic impact can be unbearable for certain communities. This is why a good balance between epidemiological and economic risks must be hold. But, determining the exact threshold of disease spread countermeasures severity is problematic without modeling particular epidemics.

First research in the field of infectious disease spread modeling is dated early XX century. In their work, Anderson McKendrick and Willian Kermack [2] considered modeling an epidemic by splitting the population into several compartments with different stages of a disease. These stages can be, for example: (S)usceptible — individuals, who can be infected in the future, (I)nfectious — individuals, who were infected, show symptoms and can infect others. After assigning the initial number of individuals in each compartment, the dynamics of its change can be modeled as a system of differential equations.

Work of McKendrick and Kermack served as the foundation for the research and development of compartmental epidemiological models, which are used these days to model various diseases like dengue fever, COVID-19, etc [3-11].

[1]The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: vladysarn@protonmail.com (V. Sarnatskyi); iaa@ukr.net (I. Baklan)
ORCID: 0000−0001−5231−6136 (V. Sarnatskyi); 0000−0002−5274−5261 (I. Baklan)

With the invention of computers and increase in available computation power, another type of epidemiological model appeared — agent- or individual-based. This type bears some resemblance to compartmental by considering the number of individuals in each compartment. But, instead of modeling it via equations, each individual is modeled separately. It means, that each individual is a separate entity, which can move inside an environment and spread the disease by contacting other individuals. Such significant increase in model granularity allows the users to model complex stochastic behaviors, common for the real world. But, on the other hand, required computational resources, needed to run the model, also increased rapidly. This can be explained easily by the following example: in a community of individuals, which can be considered a medium-sized model, a single simulation step requires computation of agent-to-agent contact events.

In this work, we explore the possibilities to optimize both time and space algorithmic complexity of simulation algorithm.

2. Problem statement

The present study aims to address the challenges of infectious disease modeling through the development of a mathematical framework for a generalized agent-based epidemiological model. As opposed to compartmental epidemiological modeling, agent-based considers modeling population as a set of distinct entities – agents. These agents can perform some activities and interact with each other. This agent-to-agent interaction is the main way of infection disease transmission. However, analysis of it requires iterating through all of agents pairs, making its algorithmic complexity quadratic, which is a problem for a large models with significant number of agents. For example, single simulation of FluTE model with number of agents equal to the population of United States of America takes about 6 hours to run on a cluster with 32x CPUs [12]. The ultimate goal of this research is to enhance the computational efficiency of this process. Specifically, the study has the following objectives:

Firstly, we aim to formulate a mathematical apparatus for a general agent-based epidemiological model. To achieve this, we will identify critical factors that influence disease transmission and develop a mathematical framework that accurately captures these dynamics.

Secondly, we aim to explore ways to optimize the complexity of the simulation algorithm, which can be computationally expensive, especially when dealing with large populations. The optimization of the simulation algorithm will involve identifying ways to reduce its computational complexity while maintaining its accuracy.

Thirdly, we aim to compare the performance of the model, based on optimized algorithm with previously developed models. By conducting a comparative analysis, we aim to demonstrate the effectiveness of the proposed optimization strategies in enhancing the efficiency of simulations. We will use standard evaluation metrics, such as the time required for simulation, to compare the performance of the optimized model with that of existing models.

3. Literature review

In this section, we describe our findings on existing tools and methods of agent-based epidemiological modeling. These tools can be divided into three distinct groups:

- Programming language modules/libraries;
- General-purpose modeling environments;
- Specialized software for epidemiological modeling.

We review each group separately.

3.1. Programming language modules/libraries

Swarm [13] is a set of libraries for Java programming languages which defines a model as a set of interacting agents, dynamics of their evolution and the schedule of events. This package is no longer maintained and has its successor — Ascape [14], which was simplified from the perspective of

programming interface. Ascape also delivers a simple graphical interface for model adjustments and inspection and spreadsheet data export.

3.2. General-purpose modeling environments

NetLogo [15] is modeling environment, consisting of the following elements:

- Subject-oriented programming language, used for describing agents behavior, their inner states evolution and interactions between each other;
- Tool set for the analysis of experiments data as the result of modeling.

Overall, NetLogo allows one to flexibly define models, but this flexibility comes with certain limitations. First of all, as showed in [16], the overall performance of NetLogo simulation engine is significantly lower, compared to similar models implemented using other tools. NetLogo features fixed modeling space geometry in the form of rectangular grid, where each agent is ``attached'' to a certain cell of it and can interact with agent on adjacent cells. Despite the interaction of agents positioned far from each other can be implemented, it is not supported internally, can be cumbersome and limits computational performance of a simulation.

RePast [17] is a software tool set, designed for modeling of complex systems. There are several implementations of it, based on different programming languages such as Java, C++. Repast also features an implementation, suitable for high-parallel environments [18].

3.3. Specialized software for epidemiological modeling

As each model with software implementation can be viewed as specialized software for epidemiological modeling, we review only those, offering some amount of configurability.

A Framework for Reconstructing Epidemic Dynamics (FRED) [19] is a software for agent-based epidemiological modeling, which uses a synthetic population database to represent every individual in a specific geographic region. The database contains geographically located synthetic households, group quarters, schools, and workplaces that reflect the actual spatial distribution of the area. Each agent has associated demographic and socioeconomic information and locations for their activities. The synthetic population closely matches the available census data for the United States with high spatial resolution. The model runs a discrete-time simulation with a time step of one day, allowing agents to interact with other agents at their shared activity locations, potentially transmitting diseases. The simulation records each infection transmission event to evaluate control measures and their impact on sub-populations. Agents can dynamically alter their daily activities. The fixed simulation step of 1 day allows for performance optimizations and is not a severe limitation for diseases with long latency periods, but it may be a limitation for diseases with short latency and infectious periods or short-term interventions. FRED was originally developed to study influenza, but it can be customized to investigate other infectious diseases, such as measles, by making adjustments to the configuration files that describe the natural history of the disease.

Unfortunately, to our knowledge, standalone software for agent-based epidemiological modeling with wide range of configurability options similar to GLEaMviz [20] doesn't exist.

4. Method

4.1. General notation

In this subsection, we describe a general mathematical apparatus of agent behavior within an epidemiological model with contact-tracing transmission.

Simulation is done in discrete steps, which can be, for example, days. These steps form the simulation step set $T = 1 \ldots N_T$.

Agents, representing individuals, who are somehow related to a modeled disease (can be infected, serve as vectors etc.) create the finite set A. Each agent has its schedule, dictating the geographical position of it. This position shouldn't be though of as coordinates in some space, as it would limit the

model, but rather an occupancy of some entity, for example the household or the school. In order to model the change of agents' location during the day, we split each simulation step into a finite number of smaller time steps from the set $H = 1 \dots N_H$. Thus, the schedule function can be defined as:

$$\phi: A \times T \times H \to P, \tag{1}$$

where P is a finite set of possible agents' location[2] (henceforth, we assume each function can depend on some external model parameters, even though it's not displayed in the notation).

Instead of using the compartments, agent-based models assign infection states to each agent, often mirroring respective compartments of compartmental models. This assignment can be expressed as infection state function:

$$\xi: A \times T \to S, \tag{2}$$

where S is a finite set of infection states.

Naturally, the infection state of an agent can change by two separate processes: intrinsic and extrinsic. Intrinsic one considers events, occurring inside the organism, which is modeled by the agent. These events can be driven by an immunological response or nature of the disease. We call this process *infection progress*. Extrinsic process involves inter-agent communication and majorly represents the transmission of a disease. We call this process *infection spread*.

Mathematically, these processes can be described by infection progress (Equation 3) and infection transmission (Equation 4) functions.

$$\Phi_p: A \times S \to [0,1] \tag{3}$$

$$\Phi_t: A^2 \times S \to [0,1] \tag{4}$$

For a given agent $a \in A$ and infection state $s \in S$, Φ_p describes the probability, that at next simulations step agent's a infection state will become equal to s as the result of intra-agent processes.

Similarly, for given agents $r, d \in S$ and infection state $s \in S$, Φ_t describes the probability, that at next simulation step agent's r infection state will become equal to s as the result of inter-agent communication with agent d. Discussing transmission function Φ_t, we refer to r as recipient, to d as donor.

Thus, the goal of an agent-based epidemiological model at simulation step $t \in T$ is to estimate

$$\xi(a, t+1), \forall a \in A \tag{5}$$

This can be done by evaluating the probability of each agent $a \in A$ to change its state to $s \in S, s \neq \xi(a,t)$[3], which can be expressed by the following expression:

$$\underset{t}{Pr}(\xi(a,t+1) = s) = 1 - \Big(1 - \Phi_p(a,s)\Big) \prod_{h \in H} \prod_{d \in A: \phi(a,t,h) = \phi(d,t,h)} \Big(1 - \Phi_t(a,d,s)\Big) \tag{6}$$

Given this equation, a trivial algorithm for computing $Pr_t(\xi(a,t+1) = s)$ is presented in Algorithm 1. As one may notice, this evaluation involves iteration over A, which has to be done for each agent as nested loop. Optimization of computation of this expression is the main concern of this work.

4.2. Optimizations

A significant part of literature discuss models, with transmission function independent of donor's and recipient's inner states, which can be formulated as following constraints of function Φ_t:

$$\begin{gathered} \forall a, a_1, a_2 \in A, s \in S: \xi(a_1, t) = \xi(a_2, t) \Rightarrow \\ \Phi_t(a, a_1, s) = \Phi_t(a, a_2, s) \wedge \Phi_t(a_1, a, s) = \Phi_t(a_2, a, s) \end{gathered} \tag{7}$$

This assumption can be used to perform a significant optimization. The main idea of it is to estimate «contagiousness» of each location at each time step by tracking infectious agents' movements, which is defined as:

$$\forall t \in T: I(p, h, s_1, s_2) = 1 - \prod_{a \in A, \phi(a,t,h) = p} \Big(1 - \overline{\Phi_t}(s_1, \xi(a,t), s_2)\Big), \tag{8}$$

[2]Henceforth, we assume each function can depend on some external model parameters, even though it's not displayed in the notation

[3]$P_t\big(\xi(a,t+1) = \xi(a,t)\big)$ can be computed as $1 - \sum_{s \in S, s \neq \xi(a,t)} Pr_t\,(\xi(a,t+1) = s)$

where:

$$\forall r, d \in A, s \in S, t \in T: \Phi_t(r, d, s) = \overline{\Phi_t}(\xi(r, t), \xi(d, t), s) \tag{9}$$

Algorithm 1: $\Pr_t(\xi(a, t+1) = s)$ computation algorithm (trivial)

Data: $A, H, S, \Phi_t, \Phi_p, t \in T,$
Result: $\Pr_t(\xi(a, t+1) = s)$
for $a \in A$ **do**
  for $s \in S$ **do**
    $\Pr_t(\xi(a, t+1) = s) \leftarrow \Phi_p(a, s)$
  end
  for $h \in H$ **do**
    $p \leftarrow \phi(a, t, h)$
    for $d \in A$ **do**
      if $p = \phi(d, t, h)$ **then**
        for $s \in S, s \neq \xi(a, t)$ **do**
          $\Pr_t(\xi(a, t+1) = s) \leftarrow 1 - (1 - \Pr_t(\xi(a, t+1) = s))(1 - \Phi_t(a, d, s))$
        end
      end
    end
  end
  $\Pr_t(\xi(a, t+1) = \xi(a, t)) \leftarrow 1 - \sum_{s \in S, s \neq \xi(a,t)} \Pr_t(\xi(a, t+1) = s)$
end

Figure 1: Description of a trivial algorithm for computation of $P_t(\xi(a, t+1) = s)$.

Subsequently, susceptible agents' movements are used together with these estimates to compute the probabilities of their state change:

$$\forall t \in T, \underset{t}{Pr}(\xi(a, t+1) = s) = 1 - \Big(1 - \Phi_p(a, s)\Big) \prod_{h \in H} \Big(1 - I(\phi(a, t, h), h, \xi(a, t), s)\Big) \tag{10}$$

Even though, there are no evidence that Equation 7 holds for real world, it can be slightly adjusted, so that the scope of possible use cases of this model increases. If Φ_t can be decomposed into functions Φ_{tr}, Φ_{td} so that:

$$\forall r \in A, s \in S: \prod_{d \in A} \Big(1 - \Phi_t(a, d, s)\Big) = \Phi_{tr}\left(r, \prod_{d \in A} \Big(1 - \Phi_{td}(d, s)\Big), s\right), \tag{11}$$

the Algorithm 2 can be used.

We don't try to solve this equation in the scope of this work, but the following solution is used in our experiments:

$$\begin{gathered} \forall f: A \to R_+, \Phi_{tr}(r, x, s) = x^{f(x)} \\ \forall \Phi_{td}(d, s): A \times S \to [0,1] \end{gathered} \tag{12}$$

Algorithm 2: $\Pr_t(\xi(a,t+1)=s)$ computation algorithm (linear)

Data: $A, H, S, P, \Phi_t, \Phi_p, t \in T,$
Result: $\Pr_t(\xi(a,t+1)=s)$
for $p \in P$ **do**
  for $h \in H$ **do**
    for $(s_1, s_2) \in S^2$ **do**
      $I(p,h,s_1,s_2) \leftarrow 0$
    end
  end
end
for $a \in A$ **do**
  for $h \in H$ **do**
    $p \leftarrow \phi(a,t,h)$
    for $(s_1, s_2) \in S^2$ **do**
      $I(p,h,s_1,s_2) \leftarrow 1-(1-I(p,h,s_1,s_2))(1-\Phi_t(s_1,\xi(a,t),s_2))$
    end
  end
end
for $a \in A$ **do**
  for $s \in S$ **do**
    $\Pr_t(\xi(a,t+1)=s) \leftarrow \Phi_p(a,s))$
  end
  for $h \in H$ **do**
    $p \leftarrow \phi(a,t,h)$
    for $s \in S$ **do**
      $\Pr_t(\xi(a,t+1)=s) \leftarrow 1-(1-\Pr_t(\xi(a,t+1)=s))(1-I(p,h,\xi(a,t),s))$
    end
  end
  $\Pr_t(\xi(a,t+1)=\xi(a,t)) \leftarrow 1-\sum_{s\in S, s\neq\xi(a,t)} \Pr_t(\xi(a,t+1)=s)$
end

Figure 2: Description of a linear algorithm for computation of $P_t(\xi(a,t+1)=s)$.

In a general case, when Algorithm 2 cannot be applied, there is still room for an optimization. Algorithm 1 considers iterating over all possible pairs of agents, even though most of them weren't even in contact. By iterating over pairs of agents, which are in the same contact group, the overall number of computation can be decreased. Here, a contact group is a subset of the agent set *A*, which were at the same location at a specified simulation step *t* and time step *h*:

$$\forall a_1, a_2 \in G_{t,h,i} \subseteq A: \phi(a_1,t,h) = \phi(a_2,t,h) \tag{13}$$

This computational optimization can be explained by the following:

$$\begin{aligned} \sum_{h\in H}\sum_{i}\left|G_{t,h,i}\right| &= |A\times H| \\ \sum_{h\in H}\sum_{i}\left|G_{t,h,i}\right|^2 &\leq |A\times H|^2 \end{aligned} \tag{14}$$

Algorithm 3: $\Pr_t(\xi(a,t+1)=s)$ computation algorithm (with grouping)

Data: $A, H, S, P, \Phi_t, \Phi_p, t \in T$,
Result: $\Pr_t(\xi(a,t+1)=s)$
$G \leftarrow \{\emptyset | \forall (p,h) \in P \times H\}$
for $a \in A$ **do**
  for $s \in S$ **do**
    $\Pr_t(\xi(a,t+1)=s) \leftarrow \Phi_p(a,s)$
  end
  for $h \in H$ **do**
    $p \leftarrow \phi(a,t,h)$
    $G_{p,h} \leftarrow G_{p,h} \cup \{a\}$
  end
end
for $(p,h) \in P \times H$ **do**
  for $(r,d) \in G_{p,h}, r \neq d$ **do**
    for $s \in S$ **do**
      $\Pr_t(\xi(r,t+1)=s) \leftarrow 1-(1-\Pr_t(\xi(r,t+1)=s))(1-\Phi_t(r,d,s))$
    end
  end
end
for $a \in A$ **do**
  $\Pr_t(\xi(a,t+1)=\xi(a,t)) \leftarrow 1-\sum_{s\in S, s\neq\xi(a,t)} \Pr_t(\xi(a,t+1)=s)$
end

Figure 3: Description of a grouping-based algorithm for computation of $P_t(\xi(a,t+1)=s)$.

As one may notice, the worst-case computational complexity of Algorithm 3 is $O(n^2)$, depending on the number of agents when $|P|=1$. But, when using with real-world models, it's easy to show that it's equal to $O(n)$. In the case of real world, we can assume, the number of places is proportional to the number of agents:

$$|A| \propto |P| \tag{15}$$

This can be explained by the fact, that, for example, the average number of residents of a household is independent of the total number of agents in the model[4].

Given this assumption, the model can be divided into a set of contact groups types, where i-th type has a number of associated contact groups, proportional to the number of agent's with coefficient k_i and number of agents in each group equal to x_i from some probability distribution X_i. Given this, computational complexity of Algorithm 3 can be estimated as:

$$O\left(g_T(n) + E\left[\sum_i n\, k_i x_i^2\right]\right) = O\left(g_T(n) + n\sum_i k_i\, E\left[x_i^2\right]\right) = O(g_T(n)+n), \tag{16}$$

where $g_T(n)$ is algorithmic complexity of the grouping algorithm itself.

Both space and time complexity of the grouping algorithm depends on the choice and implementation of the underlying data structure which holds values of function Φ. It must support the following operations:

- Add tuple (a,h,p) into collection $(a \in A, h \in H, p \in P)$;
- For tuple (h,p) return all a_i, so that (a_i,h,p) was added into collection before;
- Clear all data from the collection.

[4]There can be some variation for the models, differing in size by orders of magnitude. For example, rural settlement with population of 10^3 cannot be modeled by scaling down the model of a large city with 10^6 citizens — the economical and sociodemographic gap will be too significant.

Moreover, the following constraints are present:

- Overall number of tuples (a, h, p) is equal to $|H| \cdot |A|$;
- Number of unique tuples (h, p) in the collection is at most $|H| \cdot |P|$;
- For a given tuple (h, p), maximum number of a_i, so that (a_i, h, p) was added into the collection is at most $|A|$.

Such a data structure is a multimap, which is an extension of the associative array for the case of storing several values by one key. Being an abstract data type, a multimap can be implemented in several different ways, however, having a common feature: dividing the structure into two components. The first component is a data structure that maps a key to a reference to a container containing a list of values; the second is the type of this container. So, for example, the first data structure can be a hash table [21], and the second — a linked list [22]. Based on this, the computational algorithm complexity g_T can be given as:

$$\begin{gathered} g_T(n) = g_{T1}(n) + g_{T2}(n) \\ g_{T1}(n) = O\big(n \cdot ind_1(n) ins_2(n)\big) \\ g_{T2}(n) = O\big(itr_1(n) itr_2(n)\big), \end{gathered} \tag{17}$$

where ind_1 is the algorithmic complexity of searching by the index in the first data structure, ins_2 is the algorithmic complexity of inserting at the end of the second data structure, itr_1,itr_2 is the algorithmic complexity of iterating the elements of the first and second data structures, respectively.

Based on the aforementioned constraints and the fact that all keys can be numbered from 1 to $|P| \cdot |H|$, the obvious candidates for the first data structure are the static array and the list because it is valid for them: $ind_1(n)=O(1)$,$itr_1(n)=O(n)$. However, due to the use of the CPU cache, in practice the use of a static array is more efficient (Figure 4).

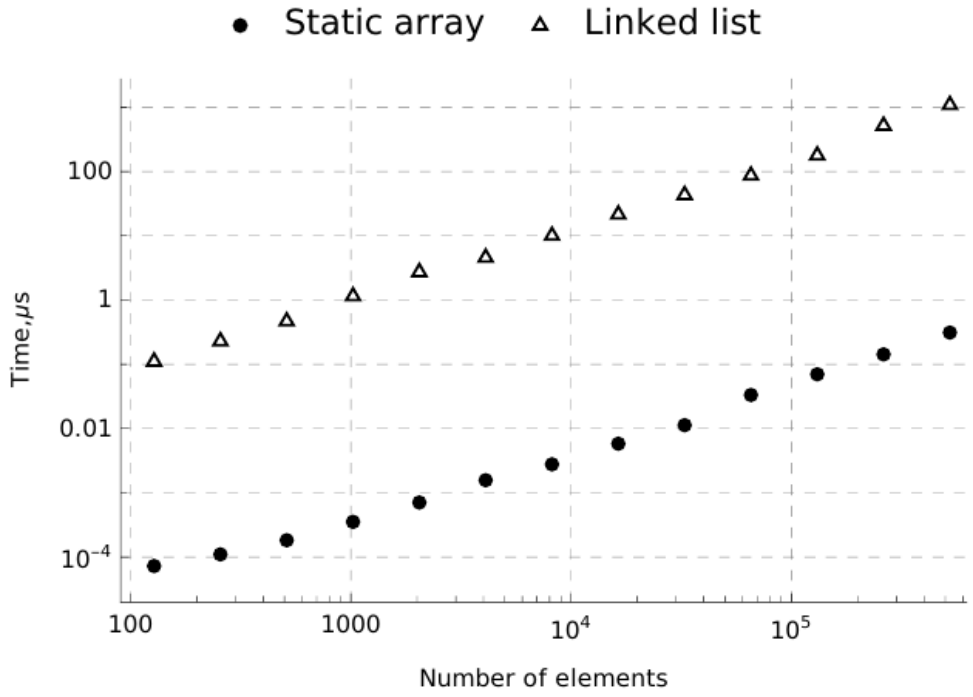


Figure 4: Comparison of container iteration speed. Cargo 1.56.0 compiler, compilation optimizations disabled

In addition to implementing a multiset using a container of containers, an approach based on hash tables can be used. In this case, it involves two data structures, the first of which maps the value of the key to the number of elements by this key stored in the structure, and the second - the key and the index of the element to its value.

This approach is appropriate, because the operations of adding and obtaining a value by key in such a structure will have an algorithmic complexity equal to $O(1)$. This is justified due to the fact that

similar operations on the hash table also have a constant complexity (in the absence of collisions) [23].

To evaluate the time performance of each approach, a number of experiments were conducted, which involved the calculation of the execution time of the grouping algorithm for each approach with different numbers of agents and the ratio of the number of agents to the number of places. The results are shown in figures 5, 6, 7.

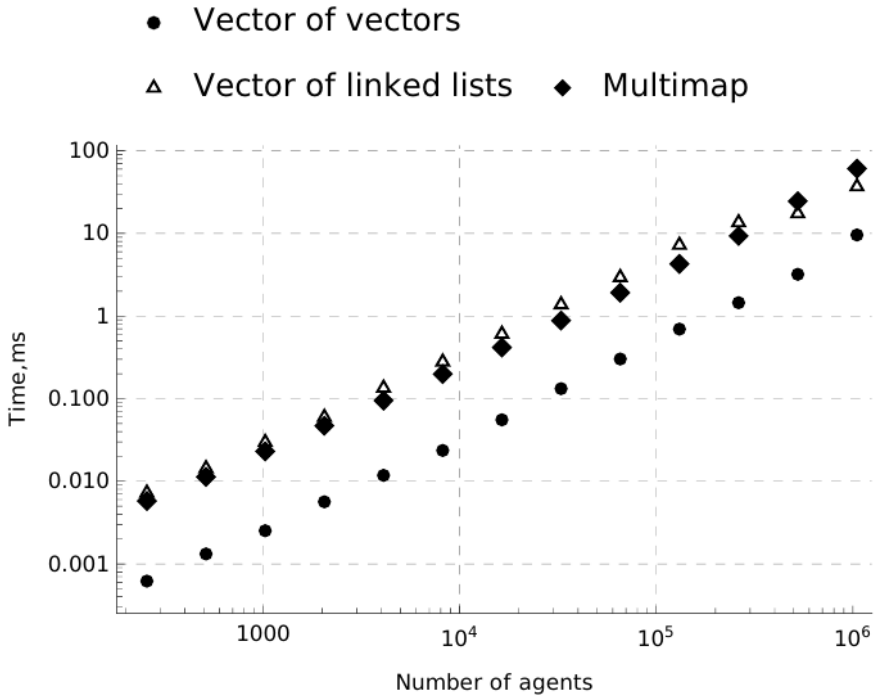


Figure 5: The execution time of the grouping algorithm for the average ratio of the number of agents to the number of places equal 16:1

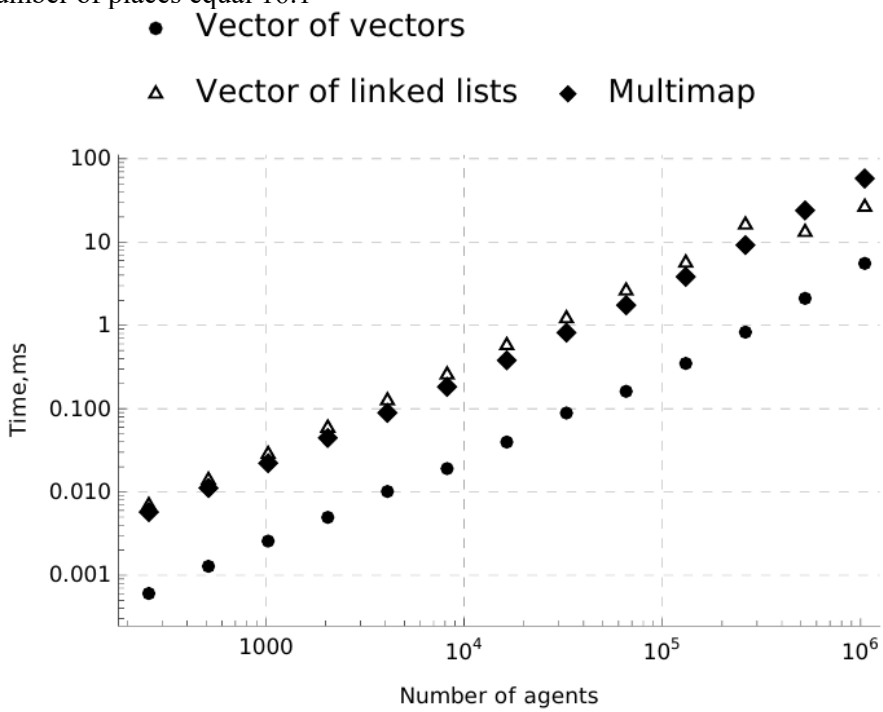


Figure 6: The execution time of the grouping algorithm for the average ratio of the number of agents to the number of places equal 256:1

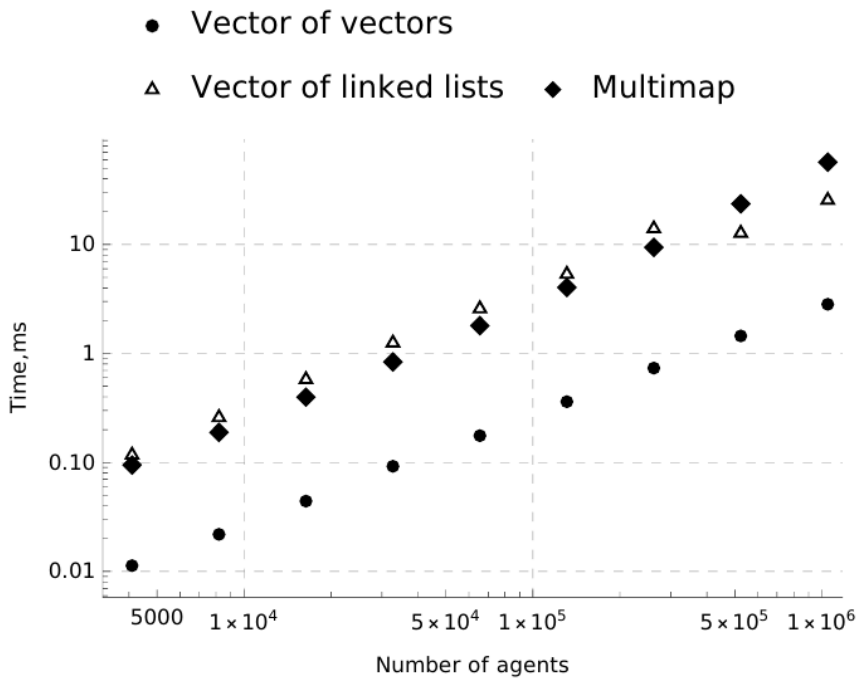


Figure 7: The execution time of the grouping algorithm for the average ratio of the number of agents to the number of places equal 4096:1

The results show that the implementation of a multiset through a vector of vectors is much more effective. This can be explained by the fact that the number of memory allocation can be reduced to a minimum with an optimally configured schedule for changing the size of the vector. And although it is impossible to predict the number of elements for each key in advance without having prior information about it, in the case of using a multiset to count visits of agents to places in the framework of infection modeling, it can be assumed that this number remains approximately the same. That is:

$$\forall p \in P, \forall t \in T, \forall h \in H: \sum_{a} 1\left(p = \phi(a,t,h)\right) \approx \sum_{a} 1\left(p = \phi(a,t+1,h)\right) \quad (18)$$

Obviously, in the real world, this statement is generally not true. For example, it is valid to assume that the number of visitors of shopping malls increases significantly on Saturdays, compared to Fridays in countries where the working week ends on Friday. However, this can be partially solved by keeping the size of the buffers equal to at least the last number of elements multiplied by some constant.

In the general case, one can achieve a minimum number of memory allocations equal to zero if set the size of each buffer equal to the number of agents. However, due to the assumption 15, the number of buffers is proportional to the number of agents. This makes the space complexity of such an algorithm equal to $O(n^2)$, which is unacceptable.

Apart from aforementioned algorithmic optimizations, computational optimizations can also be applied. One may notice, the evaluation of state change probabilities involves large amount of multiplication operations if the form:

$$\underbrace{x_1 x_1 \dots x_1}_{p_1} \dots \underbrace{x_n x_n \dots x_n}_{p_n} \quad (19)$$

Computation of such expression can be reformulated as:

$$\prod_{i=1}^{n} x_i^{p_i} = exp\left(\sum_{i=1}^{n} p_i ln x_i\right) \quad (20)$$

If we denote by $t_x, t_+, t_{exp}, t_{ln}$ the calculation time of the product, sum, exponential function, and natural algorithm, respectively, the calculation time of the left part will be equal to $t_x(\sum p_i - 1)$, the right one - $(t_{ln} + t_+)\sum p_i - t_+ + t_{exp}$. In the general case, the latter value is greater because the calculation of the natural logarithm is a time-consuming operation. However, in the case where the values of x_i are constants known during the compilation, the computation time of the right-hand side

will be equal to $t_+(\sum p_i - 1) + t_{exp}$. Because computing the product of two floating-point numbers requires more CPU cycles (for most architectures) than computing the sum [24], computing the optimized version is more efficient when:

$$t_* - t_+ > \frac{t_{exp}}{\sum p_i - 1}, \tag{21}$$

which is valid for sufficiently large values of $\sum p_i$. Thus, if the disease distribution function depends only on the resulting state and the recipient and donor states, the specified optimization can be used. To check the effect of this optimization, an experiment was conducted, during which execution times with and without it were measured for various model sizes. We observed some difference between optimized and default approach on the experimental machine for small number of agents (Fig. 5). As, the test was done on a single machine with Intel Core i7-8700K, for more general results, additional experiments should be performed. We leave this problem for a future research.

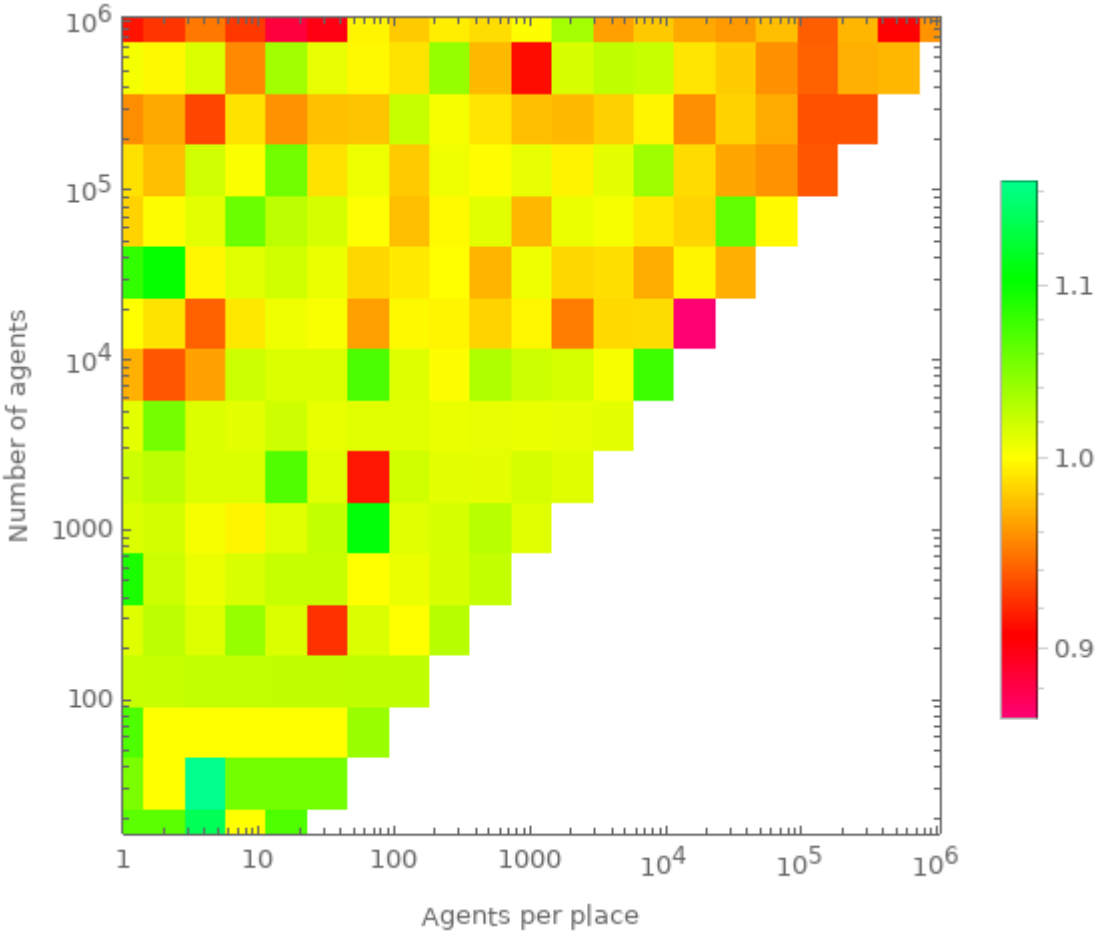


Figure 8: Simulation step performance increase (higher values are better). White areas correspond to absent data points.

4.3. Performance estimation

Comparing the performance of algorithms across different machines is not a trivial task. Modern CPUs are able to reach significant performance boost by using caching, instruction pipelines and other technologies, the impact of which on a specific algorithm is difficult to predict [25]. So, to compare out approach to existing ones, we perform two sets of measurements: first one on a desktop machine with Intel Core i7-8700K and second — on a 12 years old laptop with Intel Core i3-2330M.

In order to account for various model sizes and CPU core numbers, we use the following metric:

$$\frac{T}{10^{-6} N_a * N_d * N_s * N_c}, \quad (22)$$

where T - simulation execution time in seconds, N_a,N_d,N_s - number of agents, simulation steps and time steps in a model respectively, N_c - number of CPU cores. This metric is only suitable for algorithms with linear computational complexity as function of number of agents.

5. Results

We compared several models and their implementation approaches with CTrace [26] modeling language, whose translator implements aforementioned algorithms. Our test model didn't captured all details of each model, but instead was based around their common features like agent activity scheduling, several different place types, etc. To compute the aforementioned metric for it, a number of experiments were conducted with various model sizes.

Table 1
Performance comparison of various agent-based epidemiological models/approaches

Method	Metric, s	Hardware
FluTe[12]	4.00 — 13.71	32x Intel Core2 Duo T9400
AvilovNetLogo[16]	511.36 — 10960.15	Intel Core i3 330M 2.13Ghz
AvilovOracle[16]	5.75 — 13.71	
FRED[19]	1.20 — 7.29	SGI Altix UV supercomputer
MontañolaNetLogo[27]	2.47 — 24.71	Intel Core i5 3.20 GHz
GiacopelliLombardy[28]	240	AMD 3900X 12-core
CTrace[26]	**0.67**	**Intel Core i7-8700K 3.70 GHz**
	1.38	**Intel Core i3-2330M 2.20 GHz**

To be able to compare the developed approach to previously developed, we measured it's performance on both modern CPU and 12-years old one. Even though, the CPU generation range is significantly wide, we can assume that our approach is at least as performant as previously developed, offering the most flexible model specification. This flexibility is explained by CTrace's independence of modeling space structure, number and types of places (and other modeling entities), dynamics of a disease, etc. Given these results, we can conclude that usage of the aforementioned optimization approaches can provide significant benefits for agent-based epidemiological model computational performance on a single CPU core.

6. Conclusions

This paper presents a discussion of the challenges encountered in the implementation of agent-based modeling for infectious diseases. Specifically, the focus is on the time complexity of the simulation algorithm, which is identified as a key issue. To address this problem, a mathematical framework for agent-based epidemiological modeling is developed. This framework enables exploration of various optimization techniques for the simulation process. Several algorithms for contact tracing, which have linear time complexity, are proposed, depending on the nature of the infectious disease being modeled. These algorithms are tested and compared with existing models. Results demonstrate that the proposed approach is at least as effective as other implementations, while offering increased model flexibility through the use of the CTrace language interface. Described approach scales up by running multiple models, each per CPU core, which is often desirable for this kind of models as it leads to statistically more significant results. The question of single model parallelization we leave for a future research

7. References

[1] R. Dandekar, G. Barbastathis, Quantifying the effect of quarantine control in Covid-19 infectious spread using machine learning, medRxiv, 2020, pp. 1-13. doi: 10.1101/2020.04.03.20052084.

[2] W. O. Kermack, A. G. McKendrick, A contribution to the mathematical theory of epidemics, Proceedings of the royal society of London, Vol. 115, 1927, pp. 700–721. doi: 10.1098/rspa.1927.0118.

[3] K. S. Sharov, Creating and applying SIR modified compartmental model for calculation of COVID-19 lockdown efficiency, Chaos, Solitons & Fractals, Vol. 141, 2020, pp. 1-14. doi: 10.1016/j.chaos.2020.110295.

[4] S. Nie, W. Li, Using lattice SIS epidemiological model with clustered treatment to investigate epidemic control, Biosystems, Vol. 191, 2020, doi: 10.1016/j.biosystems.2020.104119.

[5] A. Ceria, K. Köstler, R. Gobardhan, H. Wang, Modeling airport congestion contagion by heterogeneous SIS epidemic spreading on airline networks, PLoS One, Vol. 16, 2021. doi: 10.1371/journal.pone.0245043.

[6] W. Sanusi, N. Badwi, A. Zaki, S. Sidjara, N. Sari, M. I. Pratama, S. Side, Analysis and Simulation of SIRS Model for Dengue Fever Transmission in South Sulawesi, Indonesia, Journal of Applied Mathematics, Vol. 2021, 2021, pp. 1-8. doi: 10.1155/2021/2918080.

[7] H.-Y. Shin, A multi-stage SEIR(D) model of the COVID-19 epidemic in Korea, Annals of Medicine, Vol. 53, 2021, pp. 1160–1170. doi: 10.1080/07853890.2021.1949490.

[8] S. Annas, M. I. Pratama, M. Rifandi, W. Sanusi, S. Side, Stability analysis and numerical simulation of SEIR model for pandemic COVID-19 spread in Indonesia, Chaos, Solitons & Fractals, Vol. 139, 2020, pp. 1-7. doi: 10.1016/j.chaos.2020.110072.

[9] Z. Ahmad, M. Arif, F. Ali, I. Khan, K. S. Nisar, A report on COVID-19 epidemic in Pakistan using SEIR fractional model, Scientific Reports, Vol. 10, 2020, pp. 1–14. doi: 10.1038/s41598-020-79405-9.

[10] W. Li, J. Gong, J. Zhou, L. Zhang, D. Wang, J. Li, C. Shi, H. Fan, An evaluation of COVID-19 transmission control in Wenzhou using a modified SEIR model, Epidemiology & Infection Vol. 149, 2021, pp. 1-12. doi: 10.1017/S0950268820003064.

[11] M. W. Levin, M. Shang, R. Stern, Effects of short-term travel on COVID-19 spread: A novel SEIR model and case study in Minnesota, PLoS One, Vol. 16, 2021, pp. 1-16. doi: 10.1371/journal.pone.0245919.

[12] D. L. Chao, M. E. Halloran, V. J. Obenchain, I. M. Longini Jr, FluTE, a publicly available stochastic influenza epidemic simulation model, PLoS computational biology, Vol. 6, 2010, pp. 1-8. doi: 10.1371/journal.pcbi.1000656.

[13] Minar, N., Burkhart, R., Langton, C., & Askenazi, M. The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations, ACM Transactions on Modeling and Computer Simulation, 1996.

[14] P. Patlolla, V. Gunupudi, A. R. Mikler, R. T. Jacob, Agent-based simulation tools in computational epidemiology, International Workshop on Innovative Internet Community Systems, Springer, 2004, pp. 212–223. doi: 10.1007/11553762_21.

[15] Tisue, S., & Wilensky, U. (2004, May). Netlogo: A simple environment for modeling complexity. In International conference on complex systems, Vol. 21, pp. 16-21.

[16] K. Avilov, O. Y. Solovey, Agent-Based Models: Approaches and Applicability to Epidemiology, 2012, pp. 425-443. doi: 10.1146/annurev-publhealth-040617-014317.

[17] Collier, Nick. Repast: An extensible framework for agent simulation, The University of Chicago's Social Science Research, Vol. 36, 2003, pp. 1-18.

[18] N. Collier, M. North, Parallel agent-based simulation with Repast for high performance computing, Simulation, Vol. 89, 2013, pp. 1215–1235. doi:10.1177/0037549712462620. doi: 10.1177/0037549712462620.

[19] J. J. Grefenstette, S. T. Brown, R. Rosenfeld, J. DePasse, N. T. Stone, P. C. Cooley, W. D. Wheaton, A. Fyshe, D. D. Galloway, A. Sriram, et al., FRED (a Framework for Reconstructing Epidemic Dynamics): an open-source software system for modeling infectious diseases and

control strategies using census-based populations, BMC public health, Vol. 13, 2013, pp. 1–14. doi: 10.1186/1471-2458-13-940.
[20] W. Van den Broeck, C. Gioannini, B. Gonçalves, M. Quaggiotto, V. Colizza, A. Vespignani, The GLEaMviz computational tool, a publicly available software to explore realistic epidemic spreading scenarios at the global scale, BMC infectious diseases, Vol. 11 2011, pp. 1–14. doi: 10.1186/1471-2334-11-37.
[21] D. P. Mehta, S. Sahni, Handbook of Data Structures and Applications, Chapman and Hal l/CRC, 2004, p. 163. doi: 10.1201/9781420035179.
[22] D. P. Mehta, S. Sahni, Handbook of Data Structures and Applications, Chapman and Hall/CRC, 2004, p. 47. doi: 10.1201/9781420035179.
[23] T. Van Dijk, Analysing and Improving Hash Table Performance, 10th Twente Student Conference on IT. University of Twente, Faculty of Electrical Engineering and Computer Science, Citeseer, 2009.
[24] A. Fog, et al., Instruction tables: Lists of instruction latencies, throughputs and micro-operation breakdowns for Intel, AMD, and VIA CPUs, Copenhagen University College of Engineering, 2011.
[25] J. Domínguez, E. Alba, A Methodology for Comparing the Execution Time of Metaheuristics Running on Different Hardware, European Conference on Evolutionary Computation in Combinatorial Optimization, Springer, 2012, pp. 1–12. doi: 10.1007/978-3-642-29124-1_1.
[26] Sarnatskyi V., Baklan I. CTrace: Language for Definition of Epidemiological Models with Contact-Tracing Transmission, International Scientific Conference “Intellectual Systems of Decision Making and Problem of Computational Intelligence”. – Springer, Cham, 2023, pp. 426-448. doi: 10.1007/978-3-031-16203-9_25.
[27] C. Montañola-Sales, J. F. Gilabert-Navarro, J. Casanovas-Garcia, C. Prats, D. López, J. Valls, P. J. Cardona, C. Vilaplana, Modeling tuberculosis in Barcelona. A solution to speed-up agent-based simulations, in: 2015 Winter Simulation Conference (WSC), IEEE, 2015, pp. 1295–1306. doi: 10.1109/WSC.2015.7408254.
[28] G. Giacopelli, et al., A Full-Scale Agent-Based Model to Hypothetically Explore the Impact of Lockdown, Social Distancing, and Vaccination During the COVID-19 Pandemic in Lombardy, Italy: Model Development, Jmirx med, Vol. 2, 2021, pp. 1-11. doi: 10.2196/24630.

On linear regression for fuzzy data of different quality

Serhii Mashchenko[a], Oleksandr Marchenko[a]

[a] *Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska Street, City of Kyiv, 01601, Ukraine*

Abstract

The present paper is devoted to the linear regression for a fuzzy set of fuzzy data samples. This model allows one to take into account the data of different quality. It is shown that regression parameters are type-2 fuzzy sets. Furthermore, the corresponding type-2 membership functions are given. The decomposition approach is used to investigate the T2FSs of linear regression parameters. It is shown that each T2FS of regression parameter can be decomposed according to secondary membership grades into a finite collection of fuzzy numbers. Each of these fuzzy number is the corresponding fuzzy regression parameter for a set of data numbers. This set is the corresponding α-cut of the original fuzzy set of fuzzy data samples. The illustrative example is given.

Keywords [1]

Linear regression, fuzzy least squares estimator, fuzzy number, type-2 fuzzy set.

1. Introduction

The classical regression analysis is based on crisp data and a crisp relationship between the dependent variable and the independent variables. In practice, there are many situations in which observations cannot be measured as crisp quantities, because the information is often fuzzy, incomplete, linguistic or noisy. Fuzzy regression analysis is a non-statistical method based on a fuzzy set theory rather than probability theory (see [1]). In a general model of a fuzzy regression both input and output are fuzzy. In this regard, the fuzzy regression model contains fuzzy parameters instead of error terms.

The three main fields can be distinguished in fuzzy regression analysis. These are possibilistic regression analysis, fuzzy least squares methods and machine learning techniques. The probabilistic approach in fuzzy regression analysis was first proposed by Tanaka et al. [2]. Unlike conventional regression analysis, where deviations between observed and predicted values reflect a measurement error, deviations in fuzzy regression reflect the uncertainty of the system structure expressed by fuzzy parameters of the regression model. Fuzzy parameters of the model are considered to be distributions of possibilities and determined by solving a linear programming problem that allows one to minimize fuzzy deviations subject to membership degrees constraints. Since the membership functions (MFs) of fuzzy sets (FSs) can be viewed as probability distributions, this approach was called 'possibilistic regression analysis.' The possibilistic approach was explored and improved by many authors. Reviews of possibilistic regression analysis can be found in D'Urso [3].

Fuzzy regression analysis was also considered from the viewpoint of generalizing the classical least squares method to the case of fuzzy data. The idea of this approach is to minimize in some sense the different distance measures between the predicted fuzzy values and the given fuzzy data. Fuzzy least squares methods were first proposed by Celmins [4]. Later, this approach was significantly developed by many researchers. In [1] one can find qualitative reviews on the fuzzy least squares and fuzzy least absolute methods.

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: s.o.mashchenko@gmail.com (S. Mashchenko); rozenkrans17@gmail.com (O. Marchenko)
ORCID: 0000-0003-4863-2763 (S. Mashchenko); 0000-0002-5408-5279 (O. Marchenko)

Machine learning techniques made it possible to generalize fuzzy regression analysis through the use of genetic algorithms, neural networks, and support vector machines. Relevant references can be found in Chukhrova and Johannssen [1] and Hastie et al. [5].

Often, when solving applied problems, data of different quality can be used [6]. For example, according to [7], wind tunnel experiments provide high simulation accuracy (a source of high-fidelity data). Also, experiments based on computational physical models have a higher error (source of low fidelity data). In some applications, a significantly more accurate regression model can be built if low-precision data are also used. In this case, the problem arises of constructing a regression based on data of different quality.

Using data of different quality with the aim of improving model accuracy is not a new concept. For instance, Hevesi et al. [8] predict average annual precipitation values near a potential nuclear waste disposal site using a set of precipitation measurements from the region along with more easily obtainable elevation map of the area. Kennedy and O'Hagan [9] approach the subject from the perspective of model construction using data resulting from computational simulations of varying fidelities and costs. In [7], to construct a Gaussian regression model, the problem of planning an experiment is solved with the choice of the ratio between the sizes of samples of low-precision and high-precision data. Also, to process data with a variable degree of certainty the methods of transfer learning [10], space mapping [11], and others [12] are used.

Data quality (degree of usability of data) is a complex concept. It is characterized by objectivity, integrity, relevance, measurability, controllability, etc. In some cases, the quality of data may not be crisp defined [13]. According to [14], where using data without expert knowledge, the choice of a representative sample becomes an NP-complete problem. Therefore, samples have to be found within a reasonable time, and this justifies the use of fuzzy methods that formalize expert knowledge expressed in natural language words. For instance, in the framework of quality control fuzzy expert assessments are used in [15] to construct acceptance sampling plans (how many units can be selected from a consignment and how many defective units are allowed in this sample).

In this article, we intend to investigate the method of constructing a fuzzy linear regression in the case when fuzzy data samples are of different qualities. Furthermore, the degrees of membership to a FS are known for these samples. This leads to a possibility that the regression takes into account fuzziness of the quality assessments of different samples, rather than just uncertainty of data. Examples of such FSs data samples could be: 'High quality data samples', 'Questionable data samples', 'Actual data samples', etc.

The main result of the article justifies the fact that a FS of different quality samples of type-1 fuzzy data generates a type-2 fuzzy regression model. In this model, the regression parameters are T2FSs on the real line with constant secondary grades. Although, in general, a T2FS is a rather complicated mathematical object, T2FSs with constant secondary grades are simple enough for practical use. This feature allows us to decompose this set by secondary grades into a collection of corresponding fuzzy numbers. Each of them represents the corresponding fuzzy regression parameter for a crisp set of data samples. The set in the focus is the α-cut of the original fuzzy set of data samples. We note that the well-known type 2 fuzzy regression models use crisp collections of type 2 data sets, while the model proposed in the article uses a fuzzy collection of type 1 fuzzy data samples. This is a principal difference between them. It should also be added that this article continues the line of research in the field of mathematical operations with a fuzzy set of operands, first introduced in the context of intersections and unions of fuzzy sets [16, 17].

2. Materials and Methods

In this section, we briefly review some existing theories and definitions.

2.1. Linear regression analysis for fuzzy input and output data using the extension principle

The article focuses on a linear regression in the case when data samples form a FS. We stress that one could have exploited different known methods of constructing a linear regression for fuzzy data. As an alternative, we intend to modify the method [18] based on the extension principle.

Let $K = \{1,...,|K|\}$ be the set of indices of data samples $\{y_i, x_{i1},...,x_{ip}\}$, $i \in K$, where $|K|$ is the cardinality of K. A crisp statistical linear regression has the form

$$y_i = x_i\,(\varphi(K))^T + \varepsilon_i,\ i \in K, \tag{1}$$

where for each $i \in K$, y_i is the dependent variable; $x_i = (1, x_{i1},...,x_{ip})$ is the vector of independent variables (factors, regressors) x_{il}, $l = 1,...,p$; ε_i is the independent normal random variable. The symbol ^{T} denotes the transpose, p is the number of independent variables, $\varphi(K) = (\varphi_0(K),...,\varphi_p(K)) = (\varphi_l(K))_{l=0,...,p}$ is the vector of regression parameters. Let $y(K) = X(K)(\varphi(K))^T + \varepsilon(K)$ be the matrix notation of equations (1) with $X(K) = \{x_{is}\}_{i\in K, s=0,...p}$ and $x_{i0} = 1$ for all $i \in K$, $y(K) = (y_i)^T_{i\in K}$ and $\varepsilon(K) = (\varepsilon_i)^T_{i\in K}$. For the convenience of presentation, the set K of sample indices is indicated hereinafter as a parameter in these formulae. According to the least squares method, the estimate $\hat{\varphi}(K) = (\hat{\varphi}_l(K))_{l=0,...,p}$ of the parameter vector $\varphi(K) = (\varphi_l(K))_{l=0,...,p}$ has the form

$$\hat{\varphi}(K) = [X^T(K)X(K)]^{-1}X^T(K)y(K). \tag{2}$$

Assume that the data is fuzzy. To generalize formula (2) we denote by

$$f_l(X(K), y(K)) = \hat{\varphi}_l(K) \tag{3}$$

the l-th element, $l = 0,...,p$ of the estimate $\hat{\varphi}(K)$. We also denote by

$$\tilde{X}(K) = \{\tilde{x}_{is}\}_{i\in K, s=0,...p},\ \tilde{y}(K) = (\tilde{y}_i)^T_{i\in K} \tag{4}$$

the matrix of independent variables and the vector of dependent variables, respectively, where for each $i \in K$, $\tilde{x}_{i0} = 1$ is the crisp number which is equal to 1; $\tilde{x}_{is}$, $s = 1,...,p$ and $\tilde{y}_i$ are fuzzy numbers (FNs) with the MFs $\mu_{\tilde{x}_{is}}(x_{is})$, $x_{is} \in \mathbb{R}$, $s = 1,...,p$ and $\mu_{\tilde{y}_i}(y_i)$, $y_i \in \mathbb{R}$, $i \in K$, respectively. Here, $\mathbb{R}$ is the real line.

Remark 1. Recall that a FN is a normal FS on $\mathbb{R}$ with the upper semicontinuous and quasi-concave MF (for example, see [19]).

The vector $\tilde{\varphi}(K) = (\tilde{\varphi}_l(K))_{l=0,...,p}$ of fuzzy parameters of the regression has the form $\tilde{\varphi}(K) = [\tilde{X}^T(K)\tilde{X}(K)]^{-1}\tilde{X}^T(K)\tilde{y}(K)$ according to the least squares method [18]. For each $l = 0,...,p$, $\tilde{\varphi}_l(K) = f_l(\tilde{X}(K), \tilde{y}(K)) = \{(r, \mu_{\tilde{\varphi}_l(K)}(r)) : r \in \mathbb{R}\}$ is the FN with the MF

$$\begin{aligned}&\mu_{\tilde{\varphi}_l(K)}(r) = \max_{X(K),y(K)}\{\min_{i\in K, s=0,...,p}\{\mu_{\tilde{x}_{is}}(x_{is}), \mu_{\tilde{y}_i}(y_i)\} :\ r = f_l(X(K), y(K)),\\ &X(K) = \{x_{is}\}_{i\in K; s=0,...,p} \in \mathbb{R}^{p|K|}, y(K) = \{y_i\}_{i\in K} \in \mathbb{R}^{|K|}\}\end{aligned} \tag{5}$$

by Zadeh's extension principle [20]. Here, $f_l(X(K), y(K))$ is the l-th element of the vector $\hat{\varphi}(K) = [X^T(K)X(K)]^{-1}X^T(K)y(K)$ by (2) and (3). According to Remark 1, the maximum in (5) exists. As shown in [21], the representation of FNs by u-cuts is simpler for calculations than the functional approach. Therefore, for each $l = 0,...,p$, we represent the MF of the FN $\tilde{\varphi}_l(K)$ in the form

$$\mu_{\tilde{\varphi}_l(K)}(r) = \max_{u\in[0,1]} u 1_{[\tilde{\varphi}_l(K)]_u}(r), \tag{6}$$

where $[\tilde{\varphi}_l(K)]_u$ is the u-cut of the FN $\tilde{\varphi}_l(K)$. This u-cut is the set $[\tilde{\varphi}_l(K)]_u = \{r \in \mathbb{R} : \mu_{\tilde{\varphi}_l(K)}(r) \ge u\}$ with the MF

$$1_{[\tilde{\varphi}_l(K)]_u}(r) = \begin{cases}1, & r \in [\tilde{\varphi}_l(K)]_u;\\ 0, & r \notin [\tilde{\varphi}_l(K)]_u;\end{cases} \tag{7}$$

$r \in \mathbb{R}$, $u \in [0,1]$. According to [18] and Remark 1, formula (5) implies that u-cut $[\tilde{\varphi}_l(K)]_u$ of the FN $\tilde{\varphi}_l(K)$ has the form

$$[\tilde{\varphi}_l(K)]_u = \{f_l(X(K), y(K)) : x_{is} \in [\tilde{x}_{is}]_u, s = 0,..., p;\ y_i \in [\tilde{y}_i]_u,\ i \in K\}, \quad (8)$$

where for each $i \in K$, the u-cuts of the FNs $\tilde{x}_{is}$, $s = 0,...p$ and $\tilde{y}_i$ are closed intervals

$$[\tilde{x}_{il}]_u = [[\tilde{x}_{is}]_u^D, [\tilde{x}_{is}]_u^H],\ s = 0,..., p \text{ and } [\tilde{y}_i]_u = [[\tilde{y}_i]_u^D, [\tilde{y}_i]_u^H], \quad (9)$$

respectively, of the real line $\square$. Since $\tilde{\varphi}_l(K)$ is a FN, then its u-cut $[\tilde{\varphi}_l(K)]_u$ is the interval $[\tilde{\varphi}_l(K)]_u = [[\tilde{\varphi}_l(K)]_u^D, [\tilde{\varphi}_l(K)]_u^H] \subset \square$ too. Equality (8) entails

$$[\tilde{\varphi}_l(K)]_u^D = \min\{f_l(X(K), y(K)) :\ x_{is} \in [\tilde{x}_{is}]_u, s = 0,..., p;\ y_i \in [\tilde{y}_i]_u,\ i \in K\}, \quad (10)$$

$$[\tilde{\varphi}_l(K)]_u^H = \max\{f_l(X(K), y(K)) :\ x_{is} \in [\tilde{x}_{is}]_u,\ , s = 0,..., p;\ y_i \in [\tilde{y}_i]_u,\ i \in K\}. \quad (11)$$

Thus, (6) ensures that formula (5) and the FN $\tilde{\varphi}_l(K)$ have the forms

$$\mu_{\tilde{\varphi}_l(K)}(r) = \max\{u \in [0,1] :\ [\tilde{\varphi}_l(K)]_u^D \le r \le [\tilde{\varphi}_l(K)]_u^H\},\ r \in \square, \quad (12)$$

$$\tilde{\varphi}_l(K) = \{(r,u) : r \in [[\tilde{\varphi}_l(K)]_u^L, [\tilde{\varphi}_l(K)]_u^H], u \in [0,1]\} = \{([[\tilde{\varphi}_l(K)]_u^L, [\tilde{\varphi}_l(K)]_u^H], u) : u \in [0,1]\},$$

respectively. Problems (10) and (11) are rather complicated. In view of this, it is suggested in [18] to use, for $l = 0,..., p$, the approximate value of the l-th fuzzy parameter

$$\tilde{\beta}_l(K) = \{([[\tilde{\beta}_l(K)]_u^L, [\tilde{\beta}_l(K)]_u^H], u) : u \in [0,1]\}, \quad (13)$$

with the MF

$$\mu_{\tilde{\beta}_l(K)}(r) = \max\{u \in [0,1] :\ [\tilde{\beta}_l(K)]_u^D \le r \le [\tilde{\beta}_l(K)]_u^H\},\ r \in \square, \quad (14)$$

where

$$[\tilde{\beta}_l(K)]_u^D = \min\{f_l([\tilde{X}(K)]_u^D, [\tilde{y}(K)]_u^D), f_l([\tilde{X}(K)]_u^D, [\tilde{y}(K)]_u^H, f_l([\tilde{X}(K)]_u^H, [\tilde{y}(K)]_u^D), f_l([\tilde{X}(K)]_u^H, [\tilde{y}(K)]_u^H)\}, \quad (15)$$

$$[\tilde{\beta}_l(K)]_u^H = \max\{f_l([\tilde{X}(K)]_u^D, [\tilde{y}(K)]_u^D), f_l([\tilde{X}(K)]_u^D, [\tilde{y}(K)]_u^H, f_l([\tilde{X}(K)]_u^H, [\tilde{y}(K)]_u^D), f_l([\tilde{X}(K)]_u^H, [\tilde{y}(K)]_u^H)\}. \quad (16)$$

Here, the matrices $[\tilde{X}(K)]_u^D$, $[\tilde{X}(K)]_u^H$ and the vectors $[\tilde{y}(K)]_u^D$, $[\tilde{y}(K)]_u^H$ are comprised in of the elements $[\tilde{x}_{is}]_u^D, [\tilde{x}_{is}]_u^H$, $s = 0,..., p$; $[\tilde{y}_i]_u^D, [\tilde{y}_i]_u^H$, $i \in K$, respectively (see (9)). It is clear that $[\tilde{\varphi}_l(K)]_u^D \le [\tilde{\beta}_l(K)]_u^D$, $[\tilde{\varphi}_l(K)]_u^H \ge [\tilde{\beta}_l(K)]_u^H$. Therefore, the inclusion $[\tilde{\beta}_l(K)]_u \subseteq [\tilde{\varphi}_l(K)]_u$ is hold for the closed interval $[\tilde{\beta}_l(K)]_u = [[\tilde{\beta}_l(K)]_u^D, [\tilde{\beta}_l(K)]_u^H]$. This entails $\mu_{\tilde{\beta}_l(K)}(r) \le \mu_{\tilde{\varphi}_l(K)}(r)$ and thereupon the value $\mu_{\tilde{\beta}_l(K)}(r)$ is a lower bound of $\mu_{\tilde{\varphi}_l(K)}(r)$ for $r \in \square$.

2.2. Type-2 fuzzy sets

Zadeh [22] introduced the notion of type-2 fuzzy set (T2FS) as a generalization of type-1 fuzzy set (T1FS) (that is, an ordinary FS). Unlike a T1FS, the membership degree of elements in a T2FS is a FS on closed interval [0, 1]. Based on the ideas of Karnik and Mendel [23], Mendel and John [24] gave a different definition. The T2FS $\tilde{C}$ on a set X is the collection

$$\tilde{C} = \{((x,u), \eta_{\tilde{C}}(x,u)) :\ x \in X,\ u \in J_x \subseteq [0,1]\}, \quad (17)$$

where $\eta_{\tilde{C}}(x,u)$ is a type-2 membership function (T2MF), J_x is the set of primary membership degrees u of $x \in X$ to $\tilde{C}$. The value $\eta_{\tilde{C}}(x,u)$ is a crisp number from the closed interval [0, 1] which is called the secondary grade of the pair (x,u).

Remark 2. According to the comments of Harding et al. [25] and Aisbett et al. [26], one has to define the T2MF $\eta_{\tilde{C}}(x,u)$ for all $x \in X$ and $u \in [0,1]$. To this end, one should put $\eta_{\tilde{C}}(x,u) = 0$ for all $u \notin J_x$, $x \in X$.

Remark 3. The primary membership degree *u* is deemed as the degree of manifestation of some property (which determines the given FS) of $x \in X$. According to [27], we interpret the secondary grade $\eta_{\tilde{C}}(x,u)$ as the degree of truth of the corresponding primary degree *u* of this property for *x*.

Following [24], we define embedded T2FSs and T1FSs for a T2FS

$\tilde{C} = \{((x,u),\eta_{\tilde{C}}(x,u)) : x \in X, u \in [0,1]\}$. Assume that $u_x = \mu_{C^{e1}}(x) \in [0,1]$ is a unique primary degree of membership, for each $x \in X$, where $\mu_{C^{e1}}(x)$, $x \in X$ is the MF of the T1FS $C^{e1} = \{(x, \mu_{C^{e1}}(x)) : x \in X\}$. This T1FS is called embedded in the T2FS $\tilde{C}$. We define the embedded T2FS $\tilde{C}^{e2}$ in $\tilde{C}$ in the form $\tilde{C}^{e2} = \{((x,u_x),\eta_{\tilde{C}^{e2}}(x,u_x)) : x \in X\}$ with $\eta_{\tilde{C}^{e2}}(x,u_x)) = \eta_{\tilde{C}}(x, \mu_{C^{e1}}(x)))$ for all $x \in X$.

Remark 4. According to [24], each element of the type-2 fuzzy collection $\tilde{C} = \{((x,u),\eta_{\tilde{C}}(x,u)) : x \in X, u \in [0,1]\}$ is interpreted as a subset. Thus, the collection is represented as the classical union of its elements in the sense of T1FSs. In [24], Mendel and John stated that each T2FS can be represented as a collection of all possible embedded T2FSs.

2.3. Collections of T2FSs with constant secondary grades

We shall need one special case of a T2FS to be defined according to [28, 29, 30]. Let $A = \{\eta_{\tilde{C}}(x,u) : \eta_{\tilde{C}}(x,u) > 0, x \in X, u \in [0,1]\}$ be the set of all possible positive values of secondary grades for the T2FS $\tilde{C} = \{((x,u),\eta_{\tilde{C}}(x,u)) :\ x \in X,\ u \in [0,1]\}$. Assume that the set A is finite.

Definition 1. We say that an embedded T2FS $\tilde{C}^{e2}(\alpha) = \{((x,u_x),\eta_{\tilde{C}^{e2}(\alpha)}(x,u_x)) : x \in X\}$ in the T2FS $\tilde{C}$ has a constant secondary grade $\alpha \in A$ if, for each $x \in X$, the unique primary degree $u_x = \mu_{C^{e1}(\alpha)}(x) \in [0,1]$ exists for which $\eta_{\tilde{C}^{e2}(\alpha)}(x,u_x) \equiv \alpha$, i.e. $\tilde{C}^{e2}(\alpha) = \{((x, \mu_{C^{e1}(\alpha)}(x)),\alpha) :\ x \in X\}$.

Here $\mu_{C^{e1}(\alpha)}(x)$, $x \in X$ is the MF of the embedded T1FS $C^{e1}(\alpha) = \{(x, \mu_{C^{e1}(\alpha)}(x)) : x \in X\}$ in the T2FS $\tilde{C}$.

Remark 5. Obviously, for the T2FS $\tilde{C}$ and each $\alpha \in A$, there is the unique embedded T1FS $C^{e1}(\alpha) = \{(x, \mu_{C^{e1}(\alpha)}(x)) :\ x \in X\}$ which is corresponding to the embedded T2FS $\tilde{C}^{e2}(\alpha)$ with constant secondary grade α. Hence, $\tilde{C}^{e2}(\alpha) = \{(C^{e1}(\alpha),\alpha)\} = \{(\{(x, \mu_{C^{e1}(\alpha)}(x)) :\ x \in X\},\alpha)\} =$
$= \{((x, \mu_{C^{e1}(\alpha)}(x)),\alpha) :\ x \in X\}$.

3. Formulation of the problem and the main idea

Let $N = \{1,...,n\}$ be the set of indices of fuzzy data samples $\{\tilde{y}_i, \tilde{x}_{i1},...,\tilde{x}_{ip}\}$, $i \in N$ in the form of FNs with the MFs $\mu_{\tilde{x}_{is}}(x_{is})$, $x_{is} \in \mathbb{R}$, $s = 1,...,p$, $\mu_{\tilde{y}_i}(y_i)$, $y_i \in \mathbb{R}$, $i \in N$, respectively. The matrix $\tilde{X}(N)$ of independent variables and the vector $\tilde{y}(N)$ of dependent variables are given by formula (4). Assume that qualities of data samples $\{\tilde{y}_i, \tilde{x}_{i1},...,\tilde{x}_{ip}\}$, $i \in N$ are different. Furthermore, the degrees $\mu_I(j), j \in N$ of membership to the FS $I = \{(j, \mu_I(j)) : j \in N\}$ of data samples indices are known. The following question arises: 'what are linear regression parameters in the case when fuzzy data samples are involved in the calculation with the corresponding degrees $\mu_I(j)$, $j \in N$ of membership?' In other words: 'what are the fuzzy parameters of the regression for the FS I of the data samples indices? We investigate this problem for the l-th regression parameter. First, we generalize formula (12) for the case of an arbitrary subset $K \subseteq N$ of sample indices and represent it in a convenient form for us. For each $r \in \mathbb{R}$, we consider the mapping $U_l^r : 2^N \to [0,1]$ given by

$$U_l^r(K) = \max\{u \in [0,1] :\ [\tilde{\beta}_l(K)]_u^D \le r \le [\tilde{\beta}_l(K)]_u^H\},\ K \subseteq N. \qquad (18)$$

According to (12), the mapping U_l^r associates each subset $K \subseteq N$ of data sample indices with the value of the MF $\mu_{\tilde{\beta}_l(K)}(r)$ of the fuzzy parameter $\tilde{\beta}_l(K)$ (see (13)). The latter is the FN

$$\tilde{\beta}_l(K) = \{(r, \mu_{\tilde{\beta}_l(K)}(r)) : r \in \square\}, \tag{19}$$

with the MF

$$\mu_{\tilde{\beta}_l(K)}(r) = U_l^r(K), \ r \in \text{supp}(\tilde{\beta}_l(K)) = \{r \in \square : U_l^r(K) \neq 0\}, \tag{20}$$

where $\text{supp}(\tilde{\beta}_l(K))$ is the support of the FN $\tilde{\beta}_l(K)$. Next, we generalize formulae (19) and (20) to the case of the FS I of sample indices. We denote by $\tilde{B}_l$ the l-th regression parameter, and by $M_{\tilde{B}_l}(r)$, $r \in \square$ its MF for the FS I of data samples indices. In this case, the value of the MF $M_{\tilde{B}_l}(r)$ for each fixed $r = r*$ coincides with the image $U_l^{r*}(I)$ of the FS I under U_l^{r*}, i.e. $M_{\tilde{B}_l}(r*) = U_l^{r*}(I)$. According to Zadeh's extension principle [20], the image of the FS I under the mapping $U_l^{r*} : 2^N \to [0,1]$ (see (18)) is the FS $U_l^{r*}(I) = \{(u, \mu_{U_l^{r*}(I)}(u)) : u \in [0,1]\}$ with the MF

$$\mu_{U_l^{r*}(I)}(u) = \max\{\alpha \in [0,1] : \ u = U_l^{r*}(I(\alpha))\}, \ u \in [0,1]. \tag{21}$$

Here, $I(\alpha) = \{j \in N : \mu_I(j) \geq \alpha\}$ is the α-cut, $\alpha \in [0,1]$ of the FS $I = \{(j, \mu_I(j)) : j \in N\}$ of sample indices;

$$U_l^{r*}(I(\alpha)) = \mu_{\tilde{\beta}_l(I(\alpha))}(r*), \tag{22}$$

is the image of the α-cut $I(\alpha)$, $\alpha \in [0,1]$ of the FS I of the samples indices in the mapping U_l^{r*} (see (18)). The value $U_l^{r*}(I(\alpha))$ is equal to the MF value $\mu_{\tilde{\beta}_l(I(\alpha))}(r*)$ of the l-th fuzzy parameter $\tilde{\beta}_l(I(\alpha))$ for the set $I(\alpha)$ of sample indices.

Remark 6. Let $A = \{\mu_I(j) : j \in N\}$ be the set of membership degrees values of the fuzzy set $I = \{(j, \mu_I(j)) : j \in N\}$ of sample indices. Note that the cardinality of the set A is |A| ≤ n. The situation |A| < n may occur if the degrees of membership $\mu_I(j)$ coincide for different indices $j \in N$ of samples. It is clear that while obtaining α-cuts $I(\alpha) = \{j \in N : \mu_I(j) \geq \alpha\} \neq \varnothing$ of the fuzzy set I we can assume that $\alpha \in A$ rather than $\alpha \in [0,1]$.

Thus, according to (20) and (21), for fixed $r = r*$, values of $M_{\tilde{B}_l}(r*)$ form the T1FS $\{(u, \mu_{M_{\tilde{B}_l}(r*)}(u)) : u \in [0,1]\}$ on [0, 1] with the MF $\mu_{M_{\tilde{B}_l}(r*)}(u) = \mu_{U_l^{r*}(I)}(u) = \max\{\alpha \in A : u = U_l^{r*}(I(\alpha))\}$, $u \in \text{supp}(M_{\tilde{B}_l}(r*)) = \{u \in [0,1] : \ u = U_l^{r*}(I(\alpha)), \ \alpha \in A\}$. Then (22) entails

$$\mu_{M_{\tilde{B}_l}(r*)}(u) = \max\{\alpha \in A : \ u = \mu_{\tilde{\beta}_l(I(\alpha))}(r*)\}, \tag{23}$$

where $u \in \text{supp}(M_{\tilde{B}_l}(r*)) = \{u \in [0,1] : \ u = \mu_{\tilde{\beta}_l(I(\alpha))}(r*), \ \alpha \in A\}$. Therefore, we conclude that $\tilde{B}_l$ is a FS on $\square$ with the MF whose values form T1FS on $[0,1]$. Then, according to [22], $\tilde{B}_l$ is the T2FS on $\square$. In the manner of vertical slices [27] the T2FS $\tilde{B}_l$ on $\square$ has the form:

$$\tilde{B}_l = \{(r, M_{\tilde{B}_l}(r)) : r \in R\} = \{(r, \{(u, \mu_{M_{\tilde{B}_l}(r)}(u)) : \ u \in J_r\}) : \ r \in \square\}, \tag{24}$$

where $\mu_{M_{\tilde{B}_l}(r)}(u)$, $u \in [0,1]$ is the MF of the T1FS $M_{\tilde{B}_l}(r) = \{\{(u, \mu_{M_{\tilde{B}_l}(r)}(u)) : \ u \in [0,1]\}$ of values of fuzzy degree of membership of the element $r \in \square$ to the T2FS $\tilde{B}_l$ and $J_r = \text{supp}(M_{\tilde{B}_l}(r))$ is the set of primary membership degrees, where $\text{supp}(M_{\tilde{B}_l}(r))$ is the support of the T1FS $M_{\tilde{B}_l}(r)$ for $r \in \square$. According to Section 2.2, we can also characterize the T2FS $\tilde{B}_l$ of the l-th regression parameter by means of the T2MF $\eta_{\tilde{B}_l}(r,u) = \mu_{M_{\tilde{B}_l}(r)}(u)$, $u \in J_r$ and $\eta_{\tilde{B}_l}(r,u) = 0$, $u \notin J_r$. This conclusion allows us to introduce the following notion.

Definition 2. By the regression parameter with index $l = 0,..., p$ for the FS I of sample indices is meant the T2FS

$$\tilde{B}_l = \{((r,u), \eta_{\tilde{B}_l}(r,u)) : \ u \in [0,1], \ r \in \square\}, \tag{25}$$

with the T2MF

$$\eta_{\tilde{B}_l}(r,u)=\begin{cases}\max\{\alpha\in A:\ u=\mu_{\tilde{\beta}_l(I(\alpha))}(r)\},\ u\in J_r;\\ 0,\qquad\qquad\qquad u\notin J_r.\end{cases} \tag{26}$$

Here,

$$J_r=\{u\in[0,1]:\ u=\mu_{\tilde{\beta}_l(I(\alpha))}(r),\ \alpha\in A\}, \tag{27}$$

is the set of primary membership degrees $u\in[0,1]$ with strictly positive secondary grades $\eta_{\tilde{B}_l}(r,u)$ which coincides with the support $\text{supp}(M_{\tilde{B}_l}(r))$ (see (23)) of the T1FS $M_{\tilde{B}_l}(r)$ of fuzzy membership degrees of the element $r\in\square$;

$$\mu_{\tilde{\beta}_l(I(\alpha))}(r)=\max\{u\in[0,1]:\ [\tilde{\beta}_l(I(\alpha))]_u^D\le r\le[\tilde{\beta}_l(I(\alpha))]_u^H\}, \tag{28}$$

is the MF of the l-th fuzzy parameter

$$\tilde{\beta}_l(I(\alpha))=\{(r,\mu_{\tilde{\beta}_l(I(\alpha))}(r)):r\in\square\}, \tag{29}$$

for the set $I(\alpha)$ of sample indices (see (19)-(20) for $K=I(\alpha)$);

$$\begin{aligned}[\tilde{\beta}_l(I(\alpha))]_u^D=\min\{&f_l([\tilde{X}(I(\alpha))]_u^D,[\tilde{y}(I(\alpha))]_u^D),f_l([\tilde{X}(I(\alpha))]_u^D,[\tilde{y}(I(\alpha))]_u^H),\\ &f_l([\tilde{X}(I(\alpha))]_u^H,[\tilde{y}(I(\alpha))]_u^D),f_l([\tilde{X}(I(\alpha))]_u^H,[\tilde{y}(I(\alpha))]_u^H)\}\end{aligned} \tag{30}$$

and

$$\begin{aligned}[\tilde{\beta}_l(I(\alpha))]_u^H=\max\{&f_l([\tilde{X}(I(\alpha))]_u^D,[\tilde{y}(I(\alpha))]_u^D),f_l([\tilde{X}(I(\alpha))]_u^D,[\tilde{y}(I(\alpha))]_u^H),\\ &f_l([\tilde{X}(I(\alpha))]_u^H,[\tilde{y}(I(\alpha))]_u^D),f_l([\tilde{X}(I(\alpha))]_u^H,[\tilde{y}(I(\alpha))]_u^H)\}\end{aligned} \tag{31}$$

are the estimates of the lower and upper bounds (see (15)-(16) for $K=I(\alpha)$) of the u-cut $[\tilde{\beta}_l(I(\alpha))]_u$ of the FN $\tilde{\beta}_l(I(\alpha))$;

$$I(\alpha)=\{j\in N:\mu_I(j)\ge\alpha\} \tag{32}$$

is the α-cut of the FS I of sample indices;

A is the set of the membership degrees values $\mu_I(j),j\in N$ of the FS $I=\{(j,\mu_I(j)):j\in N\}$ of sample indices (see Remark 6). According to (26), the set A includes all possible positive values of secondary grades for the T2FS $\tilde{B}_l$ of the l-th regression parameter.

4. Regression for a fuzzy set of sample indices

4.1. Decomposition of T2FSs of regression parameters

For each l-th regression parameter, $l=0,...,p$, we apply a decomposition approach to represent the T2FS $\tilde{B}_l$ in a more convenient form. Theorem 1 justifies the representation of the T2FS $\tilde{B}_l$ in the form of a collection of the embedded T2FSs with constant secondary grades.

Theorem 1. The T2FS $\tilde{B}_l$ of the l-th regression parameter is represented in the form of the collection $\tilde{B}_l=\{\tilde{B}_l^{e2}(I(\alpha)):\ \alpha\in A\}$ of embedded T2FSs

$$\tilde{B}_l^{e2}(I(\alpha))=\{(\tilde{\beta}_l(I(\alpha)),\alpha))\}, \tag{33}$$

with the constant secondary grades $\alpha\in A$. For each $\alpha\in A$, the embedded T1FS $\tilde{\beta}_l(I(\alpha))=\{(r,\mu_{\tilde{\beta}_l(I(\alpha))}(r)):r\in\square\}$ is the FN which is the l-th fuzzy parameter for the set $I(\alpha)=\{j\in N:\mu_I(j)\ge\alpha\}$ of sample indices, with the MF $\mu_{\tilde{\beta}_l(I(\alpha))}(r)$ in form (28).

Proof. According to (25), the T2FS of the regression parameter with the index $l=0,...,p$ has the form $\tilde{B}_l=\{((r,u),\mu_{\tilde{B}_l}(r,u)):\ u\in[0,1],\ r\in\square\}$. Hence,

$$\tilde{B}_l=\{\{((r,u),\max\{\alpha\in A:\ u=\mu_{\tilde{\beta}_l(I(\alpha))}(r)\}):\ u\in J_r\}\cup\{((r,u),0):\ u\notin J_r\}:\ r\in\square\}, \tag{34}$$

by (26). Remark 4 allows us to ignore the pairs (r,u) that have secondary grades equal to 0. Thus,

$$\tilde{B}_l = \{((r,u), \max\{\alpha \in A:\ u = \mu_{\tilde{\beta}_l(I(\alpha))}(r)\}):\ u \in J_x,\ r \in \square\}, \quad (35)$$

and thereupon $\tilde{B}_l = \{((r, \mu_{\tilde{\beta}_l(I(\alpha))}(r)):\ \alpha \in A):\ r \in \square\}$ by (27). Note that the collection $\{(\mu_{\tilde{\beta}_l(I(\alpha))}(r), \alpha):\ \alpha \in A\}$ is the T1FS which is formed by the unique value $u = \mu_{\tilde{\beta}_l(I(\alpha))}(r)$ of the fuzzy degree of membership of $r \in \square$. The different values $\alpha \in A$ may correspond to $u = \mu_{\tilde{\beta}_l(I(\alpha))}(r)$. Therefore, the equality $\{(\mu_{\tilde{\beta}_l(I(\alpha))}(r), \alpha): \alpha \in A\} = \{(u, \max\limits_{\alpha \in A}\{\alpha:\ u = \mu_{\tilde{\beta}_l(I(\alpha))}(r)\})\}$ holds true. Further, regrouping elements yields

$$\tilde{B}_l = \{(r, (\mu_{\tilde{\beta}_l(I(\alpha))}(r), \alpha)): \alpha \in A, r \in \square\} = \{\{((r, \mu_{\tilde{\beta}_l(I(\alpha))}(r)), \alpha): r \in \square\}: \alpha \in A\}. \quad (36)$$

Finally, by virtue of formula (29) we conclude that $\tilde{B}_l = \{(\tilde{\beta}_l(I(\alpha)), \alpha):\ \alpha \in A\}$.

The proof of Theorem 1 is complete.

4.2. Calculation of T2FSs of regression parameters

First, we construct the set $A = \{\mu_I(j): j \in N\}$ of membership degrees values of the FS $I = \{(j, \mu_I(j)): j \in N\}$ of sample indices. For each $\alpha \in A$, according to (32), we construct the α-cut $I(\alpha) = \{j \in N: \mu_I(j) \geq \alpha\}$ of the FS I. Further, for each $l = 0, ..., p$, we use the representation of the T2FS $\tilde{B}_l$ in the form of a collection of embedded T2FSs with constant secondary grades (see Theorem 1). This leads to the following sequence of calculations for each $\alpha \in A$.

We construct the embedded T1FS $\tilde{\beta}_l(I(\alpha)) = \{(r, \mu_{\tilde{\beta}_l(I(\alpha))}(r)): r \in \square\}$. This is the FN, which is the l-th fuzzy parameter of the regression for the set $I(\alpha)$ of sample indices. To construct the FN $\tilde{\beta}_l(I(\alpha))$ one can use any known methods. An application of the method worked out in Section 2.1 of [18] yields $\tilde{\beta}_l(I(\alpha)) = \{([[\tilde{\beta}_l(I(\alpha))]_u^D, [\tilde{\beta}_l(I(\alpha))]_u^H], u): u \in [0,1]\}$ (see (13), (30), (31) for $K = I(\alpha)$) with the MF

$$\mu_{\tilde{\beta}_l(I(\alpha))}(r_l) = \max\{u \in [0,1]:\ [\tilde{\beta}_l(I(\alpha))]_u^D \leq r_l \leq [\tilde{\beta}_l(I(\alpha))]_u^H\},\ r_l \in \square. \quad (37)$$

Formula (37) is justified by (14) with $K = I(\alpha)$. Then, the corresponding embedded T2FS with the constant secondary grade α has the form $\tilde{B}_l^{e2}(I(\alpha)) = \{(\tilde{\beta}_l(I(\alpha)), \alpha))\}$ according to (33).

Once all embedded T2FSs $\tilde{B}_l^{e2}(I(\alpha))$ with constant secondary grades $\alpha \in A$ have been obtained, the resulting T2FS of the l-th regression parameter has the form $\tilde{B}_l = \{\tilde{B}_l^{e2}(I(\alpha)):\ \alpha \in A\} = \{(\tilde{\beta}_l(I(\alpha)), \alpha)):\ \alpha \in A\}$ by Theorem 1. According to Remark 3, the T2FSs $\tilde{B}_l$ can be interpreted as follows. For fuzzy data of different quality, the l-th regression parameter $\tilde{B}_l$ is equal to the l-th parameter $\tilde{\beta}_l(I(\alpha))$ of the regression for the corresponding crisp set $I(\alpha)$ of data samples with the degree of truth being equal to α, $\alpha \in A$.

5. Illustration and discussion

5.1. Example

This example is devised to illustrate our approach. We stress that this example does not use real data. All data are given to the second decimal place.

To conduct the historical research, we need to find out the dependence of weight of a middle-aged person on he or her tall in the 10th century. The data borrowed from four authentic historical documents are located in lines 1-4 of Table 1. There is a reason to believe that the values in line 3 are slightly less reliable than the rest. We assume that in those days the height and the weight were measured with an accuracy to ±1Kg and ±5%, respectively. We denote by $\tilde{a} = (a^L, a^C, a^U)$ a 'triangular' FN with the MF

$$\mu_{\tilde{a}}(r)=\begin{cases}(r-a^L)/(a^C-a^L), & r\in[a^L-a^C];\\ (a^R-r)/(a^R-a^C), & r\in[a^C-a^R];\\ 0, & \text{otherwise.}\end{cases} \quad (38)$$

Table 1 contains the fuzzy data in forms of the 'triangular' FNs $\tilde{x}_{i1}=(x_{i1}^L,x_{i1}^C,x_{i1}^U)$ and $\tilde{y}_{i1}=(y_i^L,y_i^C,y_i^U)$, $i\in\{1,...,4\}$. The fuzzy data $\tilde{x}_{i1}$ and $\tilde{y}_i$ be denoted as $\tilde{x}_{i1}=appr(x_{i1}^C)$ and $\tilde{y}_i=appr(y_i^C)$, and can be interpreted as 'approximately x_{i1}^C and y_i^C', respectively. The dependence of the weight on the height for majority of data in rows 1-4 of Table 1 may seem strange, since weight gain decreases as the height increases. We assume that this can be explained by the small sample size and its non-representativeness.

Table 1

Weight depending on height

i	Height $\tilde{x}_{i1}$	Weight $\tilde{y}_i$
1	(149,150,151)	(55,1;58,60,9)
2	(159,160,161)	(63,65;67;70,35)
3	(169,170,171)	(63,65;67;70,35)
4	(189,190,191)	(80,75;85;89,25)
5	150	50
6	160	60
7	170	70
8	180	80

To solve this problem, we supplement the data sample. We know from the medical sources [31] that the Broca's formula for determining the height and the optimal weight has the form $W = H - 100$, where H is the height in cm and W is the optimal weight in Kg. According to this formula, we calculate the optimal weights for four different heights and place these data in rows 5-8 of Table 1. Thus, we have three sources of data of different quality. The high-quality source is historical, which we fully trust. We evaluate the degree of reliability of data from this source. The results which are located in rows 1,2,4 of Table 1 have the values of the degree of reliability are equal to 1. The degree of less reliable data from the second source of historical data which are located in line 3 of Table 1 is estimated at 0,9. The third source of data is medical. It is known that modern persons are taller than persons living in the 10th century and having the same weight. As a consequence, the degree of reliability of data from a medical source is not very high (for example, we estimate it as 0,7), since this source only provides information on the ratio of the weight and the height for modern people. Thus, we construct the FS 'Reliable data samples' $I=\{(1;1),(2;1),(3;0,9),(4;1),(5;0,7),(6;0,7),(7;0,7),(8;0,7)\}$ on the set $N=\{1,...,8\}$ of data sample indexes with the MF values $\mu_I(1)=\mu_I(2)=\mu_I(4)=1$, $\mu_I(3)=0,9$ и $\mu_I(5)=...=\mu_I(8)=0,7$.

In view of Remark 6, the set of membership degrees values of the FS I takes the form $A=\{0,7;0,9;1\}$. For $\alpha=1,0$; $\alpha=0,9$ and $\alpha=0,7$, according to (32), we construct the corresponding α-cuts $I(1)=\{1,2,4\}$, $I(0,9)=\{1,...,4\}$ and $I(0,7)=\{1,...,8\}$ of the FS I of data samples indices. Next, we construct FNs of parameters of linear least-squares regressions (see Section 2.1). In Example 1, these FNs are of the 'triangular' type (in general case, this is not necessary). Here, these FNs are of the 'triangular' type (in general case, this is not necessary).

For $\alpha=1$, we get embedded T1FSs $\tilde{\beta}_0(I(1))=(-86,51;-82,2;-79,6)=appr(-82,2)$ and $\tilde{\beta}_1(I(1))=(0,88;0,91;0,94)=appr(0,91)$ in the T2FSs $\tilde{B}_0^{e2}(I(1))=\{(\tilde{\beta}_0(I(1)),1))\}$ and $\tilde{B}_1^{e2}(I(1))=\{(\tilde{\beta}_1(I(1)),1))\}$, respectively, with the constant secondary grade (the degree of truth) being equal to 1.

For $\alpha=0,9$, we get embedded T1FSs $\tilde{\beta}_0(I(0,9))=(-68,47;-64,4;-60,41)=appr(-64,4)$ and $\tilde{\beta}_1(I(0,9))=(0,77;0,81;0,85)=appr(0,81)$ in the T2FSs $\tilde{B}_0^{e2}(I(0,9))=\{(\tilde{\beta}_0(I(0,9));0,9))\}$ and

$\tilde{B}_1^{e2}(I(0,9)) = \{(\tilde{\beta}_1(I(0,9)); 0,9))\}$, respectively, with the degree of truth being equal to 0,9.

For $\alpha = 0,7$, we get embedded T1FSs $\tilde{\beta}_0(I(0,7)) = (-81,79; -77; -72,3) = appr(-77)$ and $\tilde{\beta}_1(I(0,7)) = (0,85; 0,9; 0,95) = appr(0,9)$ in the T2FSs $\tilde{B}_0^{e2}(I(0,7)) = \{(\tilde{\beta}_0(I(0,7)); 0,7))\}$ and $\tilde{B}_1^{e2}(I(0,7)) = \{(\tilde{\beta}_1(I(0,7)); 0,7))\}$, respectively, with the degree of truth) being equal to 0,7.

According to Theorem 1, the resulting T2FSs of regression parameters have the forms

$$\tilde{B}_0 = \{(appr(-82,2);1), (appr(-64,4); 0,9), (appr(-77); 0,7)\}, \quad (39)$$

$$\tilde{B}_1 = \{(appr(0,91);1), (appr(0,81); 0,9), (appr(0,9); 0,7)\}. \quad (40)$$

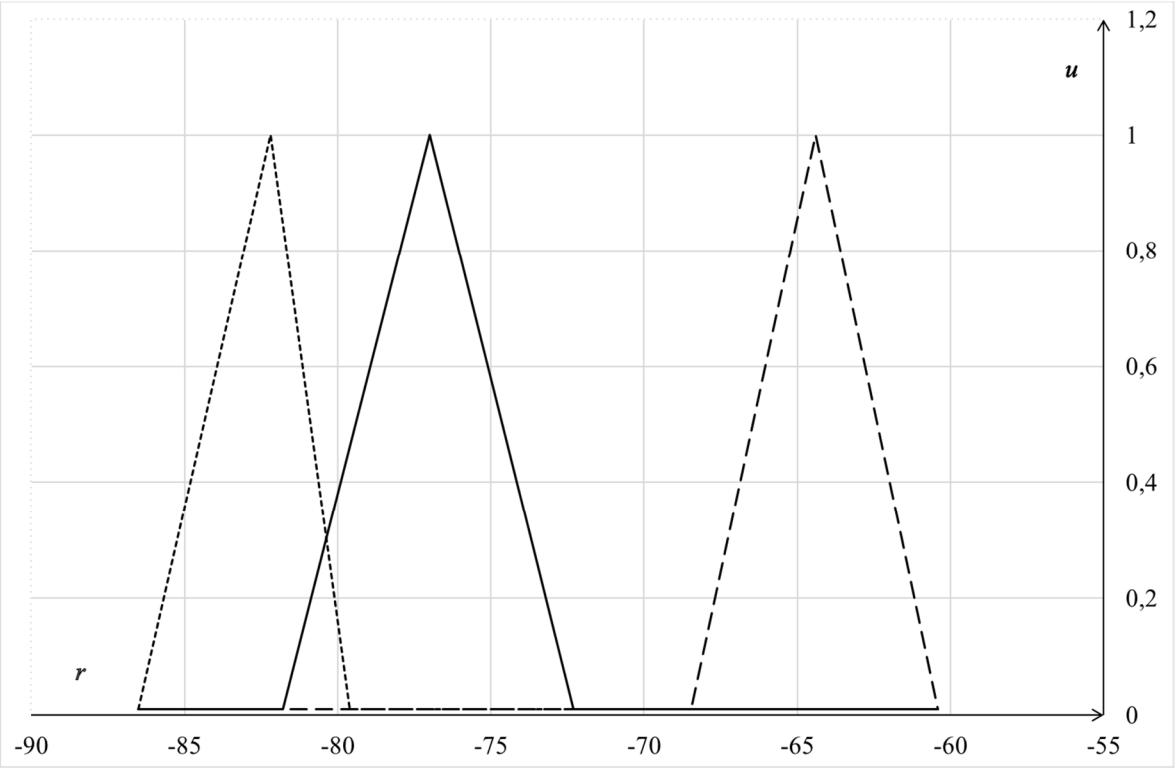


Figure 1. The lines of the levels $\alpha = 0,7$; $\alpha = 0,9$ and $\alpha = 1,0$ of the T2MF $\eta_{\tilde{B}_0}(r,u)$

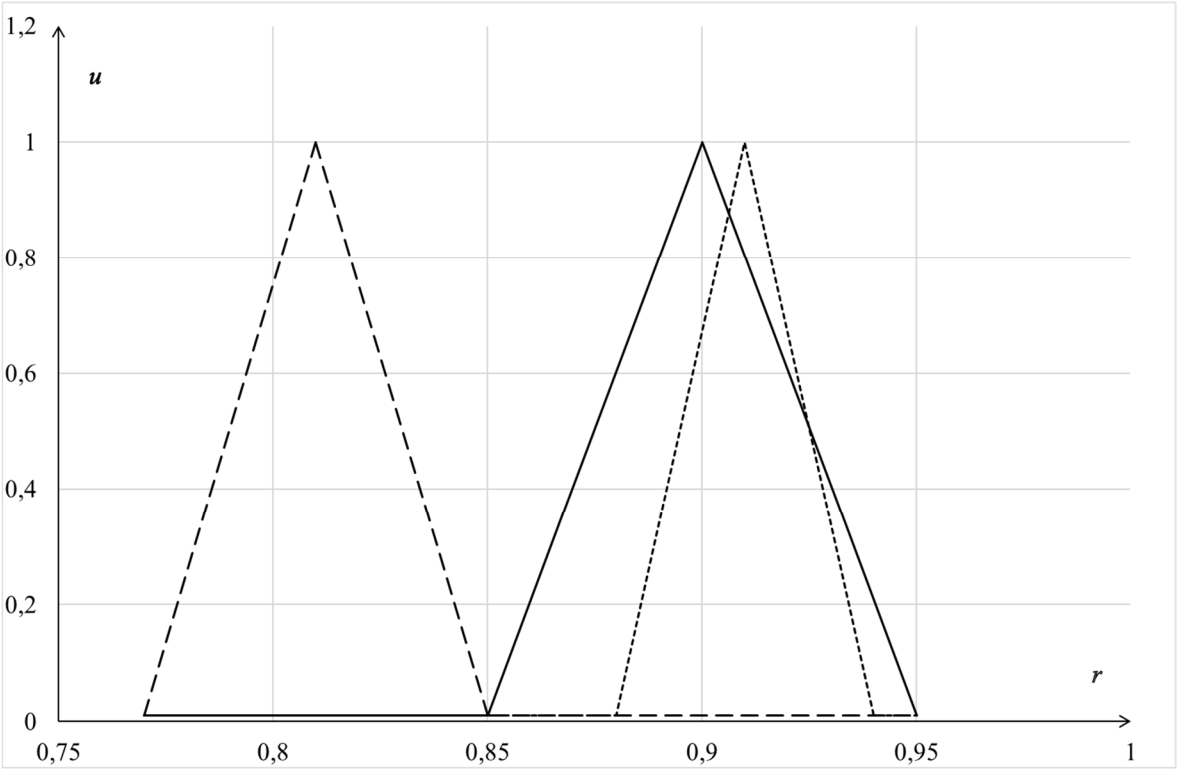


Figure 2. The lines of the levels $\alpha = 0,7$; $\alpha = 0,9$ and $\alpha = 1,0$ of the $\eta_{\tilde{B}_1}(r,u)$

The T2MFs $\eta_{\tilde{B}_0}(r,u)$ and $\eta_{\tilde{B}_1}(r,u)$ can be calculated with the help of formulae (26) and (27). Their levels $\alpha \in A = \{0,7;0,9;1\}$ are represented by solid (for $\alpha = 1$), dashed (for $\alpha = 0,9$) and dotted (for $\alpha = 0,7$) lines in Figure 1 for $\eta_{\tilde{B}_0}(r,u)$ and in Figure 2 for $\eta_{\tilde{B}_1}(r,u)$.

The obtained T2FSs can be interpreted as follows. For fuzzy data of different quality, the T2FSs $\tilde{B}_0$ and $\tilde{B}_1$ of fuzzy regression parameters values are equal to:

- the FNs $\tilde{\beta}_0(I(0,7)) = appr(-77)$ and $\tilde{\beta}_1(I(0,7)) = appr(0,9)$, respectively, for the crisp set $I(0,7) = \{1,...,8\}$ of data sample indices with the degree of truth being equal to 0,7;
- the FNs $\tilde{\beta}_0(I(0,9)) = appr(-64,4)$ and $\tilde{\beta}_1(I(0,9)) = appr(0,81)$, respectively, for the crisp set $I(0,9) = \{1,...,4\}$ of data sample indices with the degree of truth being equal to 0,9 and
- the FNs $\tilde{\beta}_0(I(1)) = appr(-82,2)$ and $\tilde{\beta}_1(I(1)) = appr(0,91)$, respectively, for the crisp set $I(1) = \{1,2,4\}$ of data sample indices with the degree of truth being equal to 1.

5.2. The discussion of the results

Let us consider a graphical interpretation obtained regression with T2FSs parameters. On Figure 3 thin solid lines demonstrate the graphs of the regression functions

$$y = (\tilde{\beta}_1(I(1))_0^L + (\tilde{\beta}_0(I(1))_0^L x = -86,51 + 0,88x\,, \tag{41}$$

$$y = (\tilde{\beta}_1(I(1)))_0^H + (\tilde{\beta}_0(I(1)))_0^H x = -79,6 + 0,94x \tag{42}$$

with parameters which correspond to the lower and upper bounds of the 0-cuts $[(\tilde{\beta}_0(I(1))_0^L, (\tilde{\beta}_0(I(1))_0^H] = [-86,51; -79,6]$ and $[(\tilde{\beta}_1(I(1))_0^L, (\tilde{\beta}_1(I(1))_1^H] = [0,88; 0,94]$ of the FNs $\tilde{\beta}_0(I(1)) = appr(-82,2)$ and $\tilde{\beta}_1(I(1)) = appr(0,91)$, respectively. These FNs are corresponding

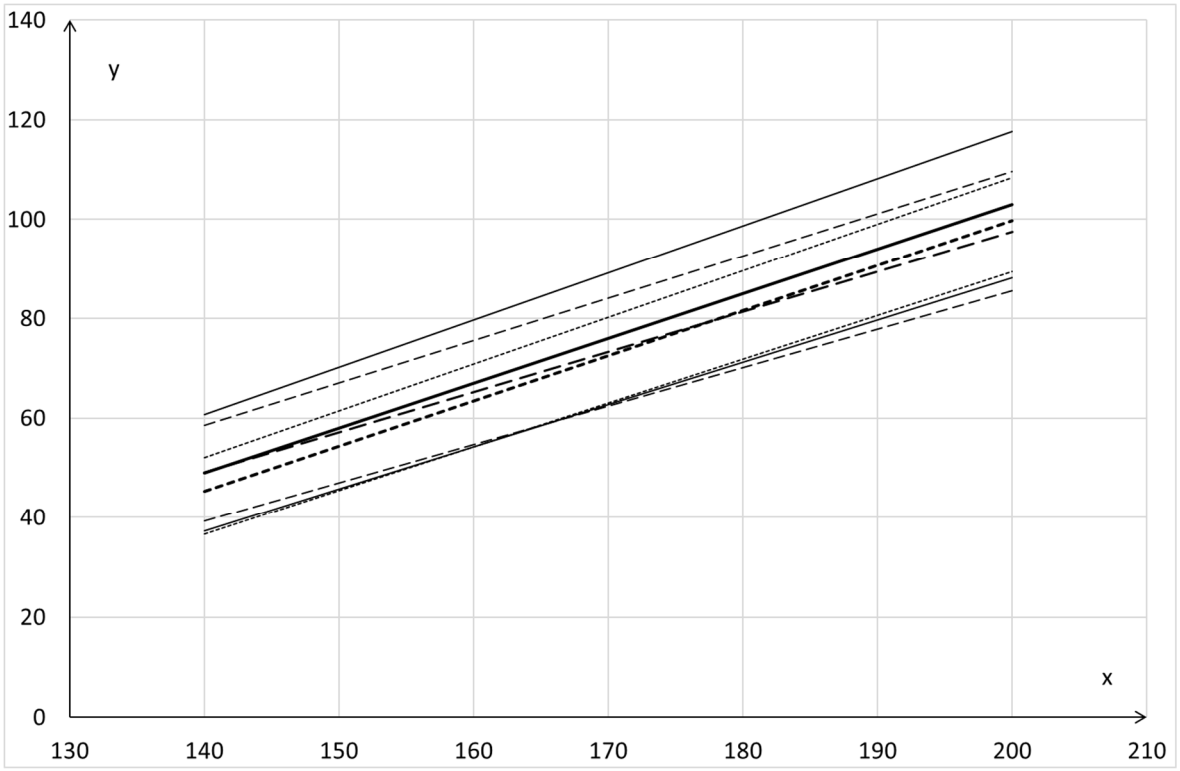


Figure 3. The illustration of the regression with the parameter T2FSs

parameters of the fuzzy linear regression for crisp set $I(1)=\{1,2,4\}$ of data sample indices. For the same set of sample indices, the thick solid line shows the graph of the regression function $y=(\tilde{\beta}_1(I(1))_1^L+(\tilde{\beta}_0(I(1))_1^L x=-82,2+0,91x$ with parameters which correspond to the 1-cuts $[-82,2;-82,2]$ and $[0,91;0,91]$ of the FNs $\tilde{\beta}_0(I(1))=appr(-82,2)$ and $\tilde{\beta}_1(I(1))=appr(0,91)$, respectively. For the crisp set $I(0,9)=\{1,...,4\}$ of data sample indices, the dashed lines demonstrate the graphs of the regression functions $y=(\tilde{\beta}_1(I(0,9))_0^L+(\tilde{\beta}_0(I(0,9))_0^L x=-68,47+0,77x$ and

$$y=(\tilde{\beta}_1(I(0,9))_0^H+(\tilde{\beta}_0(I(0,9))_0^H x=-60,4+0,85x \quad (43)$$

(thin lines), and

$$y=(\tilde{\beta}_1(I(0,9))_1^L+(\tilde{\beta}_0(I(0,9))_1^L x=-64,4+0,81x \quad (44)$$

(a thick line). For the crisp set $I(0,7)=\{1,...,8\}$ of data sample indices, the dotted lines demonstrate the graphs of the regression functions $y=(\tilde{\beta}_1(I(0,7))_0^L+(\tilde{\beta}_0(I(0,7))_0^L x=-81,79+0,85x$ and

$$y=(\tilde{\beta}_1(I(0,7))_0^H+(\tilde{\beta}_0(I(0,7))_0^H x=-72,3+0,95x \quad (45)$$

(thin lines), and

$$y=(\tilde{\beta}_1(I(1)))_0^H+(\tilde{\beta}_0(I(1)))_0^H x=-79,6+0,94x \quad (46)$$

(a thick line).

Figure 3 allows us to draw the following conclusions. Regressions corresponding to data of different quality may differ from each other and express a different relationship between the independent variables and predicted output. Therefore, the resulting regression with T2FSs of parameters should be taken as a whole as a collection of regressions corresponding to α-cuts $I(\alpha)$, $\alpha\in[0,1]$ of the FS I of data sample indices. Only in this case we get an idea about dependence between the regression and the quality of the data used. Sometimes understanding this is important.

If we need some 'maximizing' regression, which has parameters values with the primary membership degrees equal to 1, then it should be considered as a collection of regressions

corresponding to α-cuts $I(\alpha)$, $\alpha \in [0,1]$ of the FS I of data sample indices. For these regressions, we take the parameters values with the membership degrees equal to 1. In Example 1, these are regressions with graphs depicted by thick lines in Figure 3.

The regression for fuzzy data of different quality can be represented as a set of fuzzy regressions with the degree of truth $\alpha \in A$. These fuzzy regressions correspond to data samples of different quality with indices belonging to α-cuts $I(\alpha)$, $\alpha \in A$ of the FS I. Thus, the degree of truth of the regression is determined by the quality of the data for which it is constructed. The higher the quality of data samples, the greater the degree of membership of their indices to the FS I and the higher the degree of truth of the corresponding fuzzy regression in the collection that forms the regression for fuzzy data of different quality. On the other hand, increasing the amount of low-quality data reduces the degree of truth of the corresponding fuzzy regression in the collection.

6. Conclusion

According to the proposed approach, the linear regression parameters for fuzzy data of different quality are T2FSs with constant secondary grades. Although, in general, a T2FS is a rather complicated mathematical object, T2FSs with constant secondary grades are simple enough for practical use. Therefore, to represent these sets in a form which is convenient for understanding and applications, we have used a decomposition method. The results obtained have allowed us to decompose the T2FSs of linear regression parameters according to secondary grades into finite sets of FNs. These are fuzzy parameters of the regressions which correspond to different α-cuts of FSs of data sample indices. One direction of future investigation of regressions with a fuzzy set of data sample indices may be related to the development of a similar approach for possibilistic regression analysis.

7. Acknowledgements

This work has been supported by Ministry of Education and Science of Ukraine: R&D Project 0122U001844 for the period of 2022-2024 at Taras Shevchenko National University of Kyiv.

8. References

[1] N. Chukhrova, A. Johannssen, Fuzzy regression analysis: Systematic review and bibliography, Appl. Soft Comput. J. 84 (2019) 1–29. doi:10.1016/j.asoc.2019.105708.

[2] H. Tanaka, S. Uejima, K. Asai, Linear regression analysis with Fuzzy model, IEEE Trans. Syst. Man Cybern. 12(6) (1982) 903–907.

[3] P. D'Urso, Exploratory multivariate analysis for empirical information affected by uncertainty and modeled in a fuzzy manner: a review, Granular Comput. 2(4) (2017) 225–247. doi:10.1007/s41066-017-0040-y.

[4] A. Celmins, Least squares model fitting to fuzzy vector data, Fuzzy Sets and Syst. 22(3) (1987) 245–269. doi:10.1016/0165-0114(87)90070- 4.

[5] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, NY, 2009. doi:10.1007/978-0-387-84858-7.

[6] A. Forrester, A. Sobester, A. Keane, Multi-fidelity optimization via surrogate modelling, Proceedings of the Royal society A: Mathematical, Physical and Engineering Sci. 463 (2007) 3251-3269.

[7] A. Zaytsev, E. Burnaev, Minimax Approach to Variable Fidelity Data Interpolation, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR, volume 54, Fort Lauderdale, FL, USA, 2017, pp.652–661. https://proceedings.mlr.press/v54/zaytsev 17a.html.

[8] J. Hevesi, A. Flint, J. Istok, Precipitation estimation in mountainous terrain using multivariate geostatistics. Part II: isohyetal maps. J. Appl. Meteorol. 31 (1992) 677–688. doi:10.1175/1520-0450(1992)031!0677:PEIMTUO2.0.CO;2.
[9] M. Kennedy, A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika 87(1) (2000) 1–13. doi:10.1093/biomet/87.1.1.
[10] S. Pan, Q. Yang, A survey on transfer learning. IEEE Trans. on Knowledge and Data Engineering 22(10) (2010) 1345–1359. doi:10.1109/TKDE.2009.191.
[11] J.W. Bandler, Q.S. Cheng, S.A. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, J. Sondergaard, Space mapping: the state of the art, IEEE Trans. on microwave theory and techniques 52(1) (2004) 337–361. https://ieeexplore.ieee.org/document/ 1262727.
[12] C. E. Rasmussen, C. K. I. Williams, Gaussian processes for machine learning, The MIT Press, 2006.
[13] N. Arruda, J. Alcantara, V. Vidal, A. Brayner, M. Casanova, V. Pequeno, W.A. Franco, Fuzzy Approach for Data Quality Assessment of Linked Datasets, in: Proceedings of the 21th International Conference on Enterprise Information Syst. (ICEIS 2019), volume 1, 2019, pp. 399–406. doi:10.5220/0007718803990406.
[14] J. Gamez, F. Modave, O. Kosheleva, Selecting the Most Representative Sample is NP-Hard: Need for Expert (Fuzzy) Knowledge, in: Proceedings of the IEEE International Conference on Fuzzy Syst. (IEEE World Congress on Computational Intelligence), 2008, pp. 1069–1074. doi:10.1109/FUZZY.2008. 4630502.
[15] M. Z. Khan, M. F. Khan, M. Aslam, A. R. Mughal, Design of Fuzzy Sampling Plan Using the Birnbaum-Saunders Distribution, Mathematics 7(1) (2019) 9. doi:10.3390/math7010009.
[16] S. Mashchenko, Intersections and unions of fuzzy sets of operands, Fuzzy Sets and Syst. 352(1) (2018) 12–25. doi:10.1016/j.fss.2018.04.006.
[17] S.O. Mashchenko, D.O. Kapustian, Decomposition of intersections with fuzzy sets of operands, in: V.A. Sadovnichiy, M.Z. Zgurovsky (Ed.), Contemporary Approaches and Methods in Fundamental Mathematics and Mechanics. Understanding Complex Systems, Springer, Cham, 2020, pp. 417–432. https://link.springer.com/book/10.1007/ 978-3-030-50302-4.
[18] H.-C. Wu, Linear regression analysis for fuzzy input and output data using the extension principle, Computers and Mathematics with Applications 45(12) (2003) 1849–1859. doi:10.1016/S0898-1221(03)90006-X.
[19] S. Heilpern, Representation and application of fuzzy numbers, Fuzzy Sets and Syst. 91 (1997) 259–268.
[20] L. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – I, Inform. Sci. 8 (1975) 199–249. doi:10.1016/0020-0255(75)90036-5.
[21] D. Dubois, H. Prade, Operations on fuzzy numbers, International J. of Syst. Sci. 9(6) (1978) 613–626.
[22] L. A. Zadeh, Quantitative fuzzy semantics, Inform. Sci. 3 (1971) 159–176. doi:10.1016/S0020-0255(71)80004-X.
[23] N. N. Karnik, J.M. Mendel, Introduction to type-2 fuzzy logic systems, IEEE International Conference on Fuzzy Syst. 2 (1998) 915–920.
[24] J. M. Mendel, R. I. John, Type-2 fuzzy sets made simple, IEEE Trans. on Fuzzy Syst. 10 (2002) 117–127.
[25] J. Harding, C. Walker, E. Walker, The variety generated by the truth value algebra of T2FSs, Fuzzy Sets and Syst. 161 (2010) 735–749.
[26] J. Aisbett, J.T. Rickard, D.G. Morgenthaler, Type-2 fuzzy sets as functions on spaces, IEEE Trans. on Fuzzy Syst. 18(4) (2010) 841–844.
[27] J. M. Mendel, Type-2 fuzzy sets: some questions and answers, IEEE Connections. Newsletter of the IEEE Neural Networks Society 1 (2003) 10–13.
[28] S.O. Mashchenko, Sums of fuzzy set of summands, Fuzzy Sets and Syst. 417 (2020) 140–151. doi:10.1016/j.fss.2020.10.006.
[29] S.O. Mashchenko, Sum of discrete fuzzy numbers with fuzzy set of summands, Cybernetics and Syst. Analysis 57(3) (2021) 374–382. doi:10.1007/s10559-021-00362-w.
[30] S.O. Mashchenko, Minimum of fuzzy numbers with a fuzzy set of operands, Cybernetics and Syst. Analysis 58(2) (2022) 210–219. doi: 10.1007/s10559-021-00362-w.

[31] J. C. Segen, Concise dictionary of modern medicine, McGraw-Hill, New York, NY,2006.

Application of Polarization-Time Model Seismic Signal for Remote Monitoring of Potential Sources Emergencies by Three-Component Seismic Station

Tetiana Vakaliuk[a], Ihor Pilkevych[b], Yurii Hordiienko[b] and Veronika Loboda[b]

[a] *Zhytomyr Polytechnic State University, Chudnivska str., 103, Zhytomyr, 10005, Ukraine*
[b] *Korolev Zhytomyr Military Institute, Mira ave., 22, Zhytomyr, 10004, Ukraine*

Abstract

The aim of the work is to develop theoretical foundations aimed at improving efficiency of detecting hazard factors related to natural and man-made disasters based on the results a seismic observation. An analysis of seismic monitoring equipment operated by the Main Center for Special Control of State Space Agency of Ukraine and methods of seismic data processing was carried out. Various components of seismic signals from events with epicenters in the regional zone was analyzed. A relationship was established between location of seismic event location relative to seismic observation station and angular characteristics of main seismic signal components for events with regional epicenters. In order to reduce the time of providing preliminary information to users about the fact of a seismic event and its seismic source parameters, it is proposed to move from a search task of determining focal point location to remote monitoring of potential sources of natural and man-made disasters. In order to realize continuous remote monitoring for potential sources of emergency events, a polarization-time model was proposed for expected seismic signal from an event with a focal point in a controlled area (object). Results of applying the proposed approach for detecting seismic signals from earthquakes with foci in the regional zone are presented. Relative simplicity of implementing our approach allows monitoring potential sources an emergency events in a time mode close to real time. Parameters of a polarization-time model of seismic signals for nuclear power plants in Ukraine and hydraulic structures of the Dnipro cascade are presented. At the same time, this work is included in a broader set of studies aimed at improving existing and creating new theoretical foundations for detecting seismic hazards of natural and man-made origin. In addition, this research covers processes of occurrence and propagation of seismic emergencies that may threaten society's vital activity.

Keywords [1]

Emergency events, Earthquake, seismic signal, three-component seismic station, signal detection, seismic waves, polarization-time model of seismic signal, seismic monitoring, automatically controlled monitoring system, source of emergency

1. Introduction

The interaction of natural, technological, and social systems on Earth leads to the emergence of numerous emergency situations that can have natural, technological, or military origins. Among these emergencies, the most dangerous for the Earth's biosphere are earthquakes, floods, hurricanes, accidents at nuclear power facilities, military actions, and others that can have a serious impact on human life and the ecological balance of the planet.

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: tetianavakaliuk@gmail.com (T. Vakaliuk); igor.pilkevich@meta.ua (I. Pilkevych); ua_gordienko@ukr.net (Y. Hordiienko); veronichkaloboda@gmail.com (V. Loboda)
ORCID: 0000-0001-6825-4697 (T. Vakaliuk); 0000-0001-5064-3272 (I. Pilkevych); 0000-0002-6141-4675 (Y. Hordiienko); 0000-0002-3535-0233 (V. Loboda)

Recently, disasters have become global in nature. Population growth increases the scale of consequences from natural disasters as more and more people are forced to live in dangerous places, which are located in areas of flooding, landslides, earthquakes, etc. Examples of this are catastrophic consequences of powerful earthquakes in recent decades – China (2008), Japan (1995, 2011), Haiti (2010), Indonesia (2004), Turkey (1999, 2023), and others.

Ukraine is no exception. In certain regions of Ukraine with a high population density, there are high-risk facilities, which dramatically increases risk of possible natural disasters, accidents and catastrophes. Every year, natural and technological ES occur in our country, leading to loss of life and significant material damage [1-4]. Hostilities as a result of Russia's armed aggression increase risks at potentially hazardous facilities, such as nuclear power plants, dams of the Dnipro cascade, and others. In addition, earthquakes are one of the hazardous natural phenomena that can occur on Ukrainian territory and lead to an emergency. Earthquakes cover large areas and are characterized by destruction of buildings and structures, massive fires and industrial accidents, as well as negative psychological impact.

Tendency of sharp increase in numbers and destructive force caused by disasters of different nature causes deterioration of social, economic and environmental consequences and points to the need to develop effective measures for prevention and elimination of disasters of different nature. Therefore, an urgent way to solve this problem is to develop an effective system for detecting hazards at stage of their inception, establishing causes of their manifestation and impact on them in order to prevent disasters and reduce their consequences.

In order to organize and ensure protection of citizens from the consequences of disasters, Civil Protection System (CPS) was created in Ukraine [1,2]. Tasks of the CPS are: prevention of emergencies and elimination their consequences; warning population about threat and occurrence of emergencies; protection of population from consequences thereof. A main component of qualitative fulfillment of tasks assigned to the CPS is timely receipt of information about disasters and their consequences, for which purpose a disaster monitoring system (DMNS) is created. At same time, the task of enhancing capabilities of the CPS by expanding monitoring methods becomes urgent.

One of these methods is seismic, its main advantages being high efficiency of establishing fact of a seismic event and ability to simultaneously monitor several potentially hazardous objects (areas) remotely, which reduces risk for technical means of observation and service personnel.

One of information segments of the DMNS is the Main Center for Special Monitoring (MCSM) of the State Space Agency (SSA) of Ukraine, which, through National Seismic Observation System and Governmental Information and Analytical System for Emergency Situations, provides information to the CPS on seismic situation in Ukraine and neighboring countries [5]. The seismic observation network (SON) of MCSM includes three-component seismic stations (TCSS), a seismic grouping system (SGS), which was included in the International Seismic Monitoring Network as PS45 station, and a National Data Center (Figure 1).

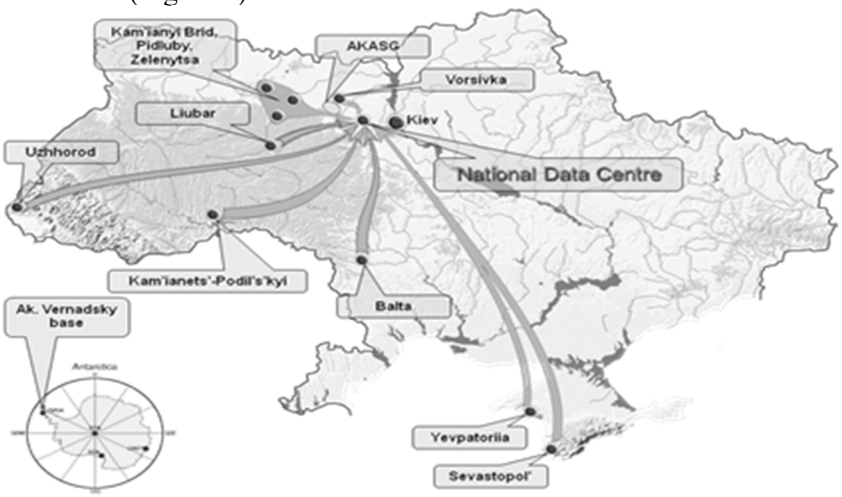


Figure 1: Network of seismic observation MCSM SSA of Ukraine

Seismic data processing is performed in manual and automatic modes. Decision on the parameters a seismic event and possible consequences is made by the operational duty officer of the MCSM based on results of analysis information on seismic signal parameters (time arrival of main types a seismic wave, their amplitude and period) received from each observation point [6].

Methods of seismic data processing currently implemented in the MCSM allow providing preliminary information to users about seismic event parameters and possible consequences in 15 minutes, and final information in 40 minutes after event occurrence [6,7].

Modernization of seismic observation equipment, transmission and processing of measurement data, transition to digital information processing, which is carried out at the MCSM within the framework of national and international programs, allows to move to a qualitatively new level of seismic monitoring. However, these capabilities are currently used to a limited extent, since seismic data processing methods implemented in the MCSM are based on algorithms for "manual" processing of seismograms by operators. At the same time, time limits for providing information to users are caused primarily by capabilities of existing approaches to processing seismic detection data.

2. Review of the literature

At present, in international and national data centers other countries, the main trend in detecting a seismic event in automatic mode is to use relatively simple measurement data processing procedures (such as STA/LTA), which allow for rapid data analysis and detection of seismic signals, but at the same time increase the density of the seismic observation network [6,8]. Territorial limitations of the seismic observation network of the MCSM, especially after temporary loss of observation stations "Sevastopol" and "Yevpatoriya" due to the occupation of the Crimea by the Russian Federation, necessitates development of methodological principles for solving seismic monitoring tasks by individual observation stations (OS) and TCSS.

Polarization analysis apparatus is used to process TCSS measurement data, but usually to determine angular characteristics of seismic wave exit to ground surface of seismic recording sections, which are defined as a signal based on preliminary detection results [6]. Another approach to applying the polarization analysis apparatus is polarization filtering, which consists in increasing signal to noise ratio of seismic vibrations from a certain direction [9]. Application of this approach makes it possible to detect seismic signals from sources with cells along a propagation path (beam), which is determined by angular characteristics – azimuth α and angle of arrival at daylight surface γ, but does not determine position of source within beam [10].

For analyzing data from seismic grouping system measurements, a method of controlled directional reception is used [6]. However, its application allows detecting signals over entire area covered by a generated beam (maximum of the seismic grouping system radiation pattern formed by time delays for each element of a seismic group for a corresponding area of Eart's surface), and also doesn't allow determining position of a seismic source within a beam.

Development of theoretical foundations for continuous remote monitoring potential sources of emergencies (PSE) natural and man-made disasters based on observations by a separate TCSS is one possible way to overcome this problem. In addition, such a need is caused by problem of ensuring the functional stability of seismic observation network as a whole, especially in cases when it is impossible to use seismic data from several stations (equipment failure, communication interruption, maintenance, etc.). Another prerequisite for improving seismic data processing methods is modernization of the MCSM seismic observation network, which is being carried out within international and national projects. Therefore, development as well as improvement of the existing theoretical foundations for automated processing of seismic signals recorded by a separate TCSS is a relevant issue.

Aim of the work is development of methodological principles for implementing continuous remote monitoring a potential source on natural and man-made emergencies based on results of observations by a three-component seismic station, taking into account kinematic and dynamic features, seismic component of signal from events centered in regional area.

3. Research methods

Reducing time for providing preliminary information to users about a seismic event and its parameters can be realized by moving from a search task of determining location a seismic source in absence of a priori information to remote monitoring potential sources by natural and man-made hazards.

PSE include potentially dangerous objects (civilian and military) and seismically active zones, earthquakes with centers in which pose a real potential threat to territory of Ukraine. Therefore, consideration of developing appropriate methodological principles for implementing continuous remote monitoring of PSE is important.

Problem of PSE monitoring by seismic means consists in detecting seismic signal, identifying main types of seismic waves, determining parameters a seismic signal components and making a decision on whether the detected signal belongs to a source from controlled area (facility).

Nowadays, the main direction for solving problems of PSE monitoring based on the results on seismic observations is to determine compliance of adopted implementation with previously registered seismic signals from events in controlled area [6]. Main problem in applying existing approaches is lack of reference signals for each of monitoring objects. So, to implement continuous remote monitoring of PSE, it is necessary to use additional information criteria due to seismic signals peculiarities. Such criteria can be used as dynamic (polarization of soil particle oscillations) and kinematic (propagation time) properties of seismic signal components.

3.1. Polarization properties of seismic signal components

Solving seismic monitoring tasks by a single observation point (OP) consists of following stages: seismic signal detection, identification, a seismic signal component (seismic wave types identification), seismic event center location, seismic source parameters estimation. In case a single-position observation is used, the last three stages are solved if the problem of determining the main components a seismic record is confidently solved. So, when analyzing existing methods of seismic data detection and processing, the main attention will be paid to possibility of solving this problem. Another criterion should be simplicity of software and algorithmic implementation for processing methods, which in turn will ensure possibility real-time processing a measurement data.

Currently, most of implemented approaches to detecting seismic signals based on TCSS observations use a criterion for exceeding amplitude thresholds (such as STA/LTA) [11], which is quite effective at an energy signal-to-noise ratio of at least 2÷3. However, using this approach leads to a limitation of magnitude sensitivity of a seismic station. In addition, this method doesn't allow to accurately determine components of seismic signal, since the arrival of next seismic wave occurs against a background of tail part of previous one. On the Figure 2 shows results of processing seismic signals from earthquakes in Ukraine using STA/LTA detection method. An example of numbered list is as following.

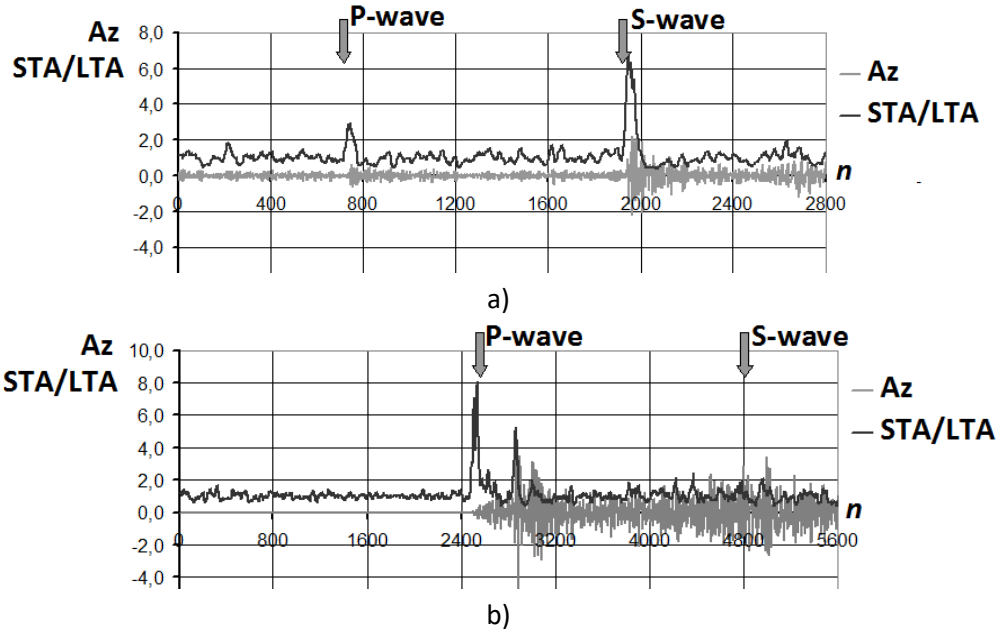


Figure 2: Results processing of seismic signals from earthquakes on territory of Ukraine, using STA/LTA detection method: a – earthquake with center in Chernivtsi region (January 16, 2020, M=2.4); b – earthquake with center in Poltava region (February 3, 2015, M=4.6)

Applying a detection method based on amplitude criterion doesn't always allow to determine seismic signal components, since the arrival of a next seismic wave occurs against a background of a tail part from the previous one (Figure 2b, several sections a seismic record after seismic signal arrival for which the STA/LTA ratio exceeds threshold).

After a seismic event, such as an earthquake or explosion, elastic ground vibrations begin to spread in all directions from the epicenter. These vibrations can travel significant distances from epicenter and provide an important source for determining parameters of a seismic event. Depending on ground motion during seismic wave propagation, seismic waves can be classified into volumetric (longitudinal P-waves and transverse S-waves) and surface (Rayleigh waves and Love waves).

P-waves cause compression and rarefaction of soil particles along a direction of wave propagation, as shown in Figure 3. Arrows in this figure show areas of compression and rarefaction. In case of S-waves, soil displacement occurs perpendicular to direction of wave propagation. In addition to bulk waves, surface waves can also propagate across the Earth's surface. These waves are divided into two types: Rayleigh (L_r) and Love (L_q). In a Rayleigh wave, soil particles move in a vertical plane that is directed along a wave propagation direction, and their trajectories form ellipses. In a Lova wave, particles move in a horizontal plane perpendicular to the wave propagation direction.

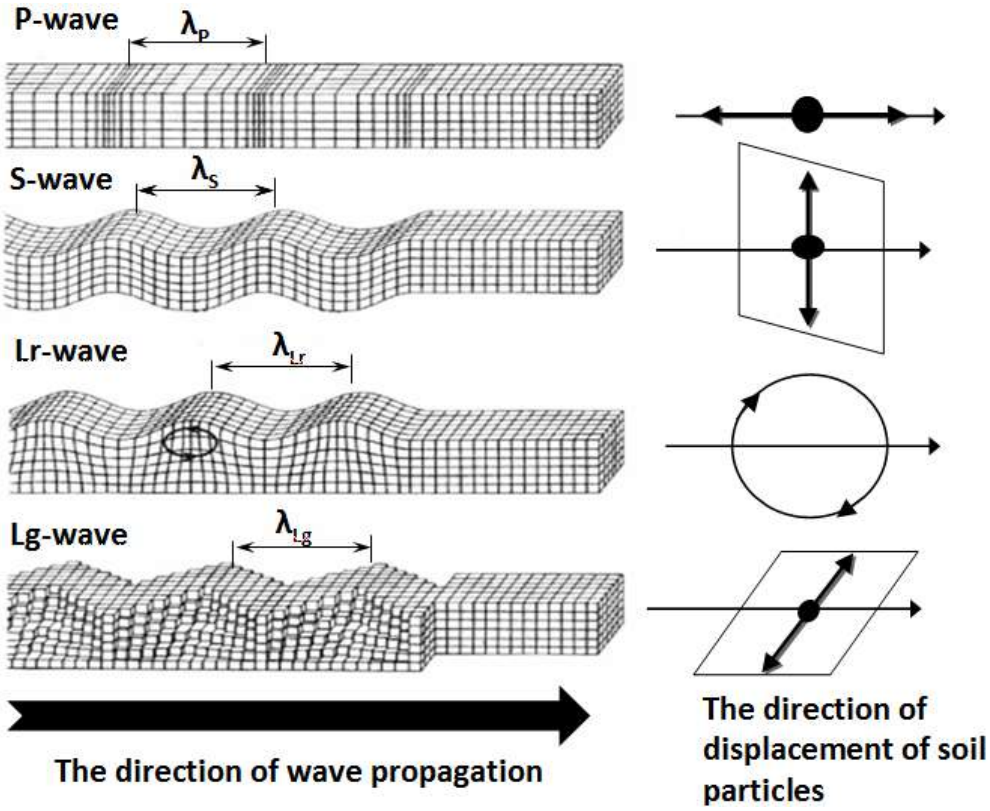


Figure 3: The nature of soil displacement for different types of seismic waves

Taking into account specific features of soil particle displacement for each main type of seismic waves in an event with centers in regional areas, angle characteristics of such waves will be related to the position of the seismic source relative to the OP as follows [10]:

– *P*-wave – since oscillation of particles in soil occurs along seismic wave propagation direction, the calculated angular characteristics coincide with position of seismic event center relative to the OP *($\alpha_P=\alpha_{sec}, \gamma_P=\gamma_{sec}$);*

– *S*-wave – due to soil oscillation occurs perpendicular to direction of wave propagation at this phase, angular characteristics are calculated (α_S, γ_S) will be different from direction to seismic event center relative to OP at 90°;

– *Lr*-wave – a superficial wave with elliptical polarization focused perpendicular to propagation direction, so its calculated azimuth will be different from input azimuth one on 90°,that is $\alpha_{Lr} \perp \alpha_P$;

– *Lq*-wave is a surface wave with an elliptical polarization in direction of propagation, so the calculated azimuth will coincide with actual azimuth to seismic source *($\alpha_{Lq}=\alpha_P$)*.

After determining types of seismic waves and estimating their arrival time, distance from OP to seismic event center is estimated using a difference hodograph. In Figure 4, hodograph – shows a graph of dependence of time of propagation from source to seismic wave receiver on epicentral distance for regional zone [6,10].

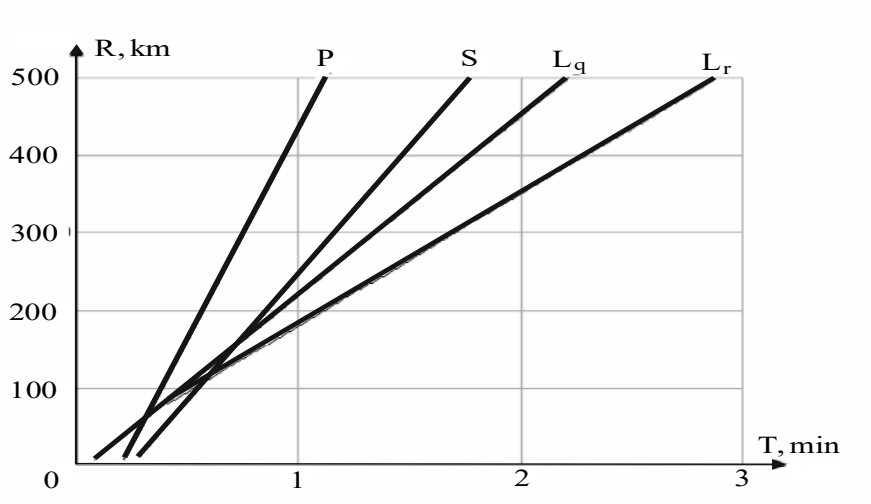


Figure 4: Hodograph of different types of seismic waves for regional zone

Slight differences occurring over short distances can cause each subsequent phase of a seismic signal to overlap the previous one, making it difficult for automatic identification of individual seismic signal components. Therefore, to determine a seismic event center in a regional area, it is sufficient to limit detection and identification to only the volume waves – P- and S-waves. At the same time, the time of obtaining measurement data necessary for detecting a seismic event and determining its parameters is limited by time of arrival second component a seismic signal (S-wave) for a separate TCSS, or time of arrival of one component (P-wave) to the observation points of SON.

Based on soil particle displacement features for bulk waves (Figure 3), S-wave detection can be realized as a result of using (taking into account) peculiarities of their angular characteristics, as a search for seismic signal recording areas for which the following conditions are met $(\alpha_P, \gamma_P) \perp (\alpha_S, \gamma_S)$.

On Figure 5 shows a seismic vertical component of recording from earthquake signal with center in Chernivtsi region (16.01.2020, M=2.4), registered by seismic station “Malyn” (Zhytomyr region), and results obtained from calculations on angular characteristics for recording sections corresponding to background, P- and S-waves. As shown previously, volume waves, in contrast to background waves, have grouping of their angular characteristics around a certain azimuth and angle of exit (direction).

As value of the angular characteristics of P- and S-waves, arithmetic mean seismic wave azimuth and exit angle to day surface $\bar{\gamma}$ is used, which is defined as [7,10].

$$\bar{\alpha} = \sum_{i=1}^{N} arctg \frac{A_{xi}}{A_{yi}} \cdot \quad (1)$$

$$\bar{\gamma} = \sum_{i=1}^{N} arctg \frac{A_{zi}}{\sqrt{A_{xi}^2 + A_{yi}^2}} . \quad (2)$$

where N – is a sample size, N = 40, which corresponds to 1 s, at a sampling rate of f_D = 40 Hz;

A_{zi}, A_{xi}, A_{yi} – a displacements of soil particles in the vertical and horizontal (north-south and east-west) channels, respectively.

An angle between position of P- and S-wave angular characteristics is defined as [8].

$$\cos\Omega = x_p \cdot x_s + y_p \cdot y_s + z_p \cdot z_s . \quad (3)$$

where $\{x_p, y_p, z_p\}$ and $\{x_s, y_s, z_s\}$ coordinates of single vectors for P- and S-wave angular characteristics.

Coordinates of single vectors of angular characteristics of P- and S-waves are related to arithmetic mean azimuth index $\bar{\alpha}$ and exit angle $\bar{\gamma}$ of corresponding seismic wave types as:

$$x_P = \cos(\bar{\gamma}_P)\cdot\cos(\bar{\alpha}_P),\ y_P = \cos(\bar{\gamma}_P)\cdot\sin(\bar{\alpha}_P),\ z_P = \sin(\bar{\gamma}_P); \quad (4)$$

$$x_S = \cos(\bar{\gamma}_S)\cdot\cos(\bar{\alpha}_S),\ y_S = \cos(\bar{\gamma}_S)\cdot\sin(\bar{\alpha}_S),\ z_S = \sin(\bar{\gamma}_S). \quad (5)$$

Values of determined angular characteristics for seismic recording of a signal from an earthquake with a center in Chernivtsi region (Figure 5c-b) are respectively for P-wave $\bar{\alpha}_P$ = 217º and $\bar{\gamma}_P$ = 37º, for S-wave $\bar{\alpha}_S$ = 293º and $\bar{\gamma}_S$ = 4º. Values of angular characteristics of seismic signal components were determined at levels of 76º.

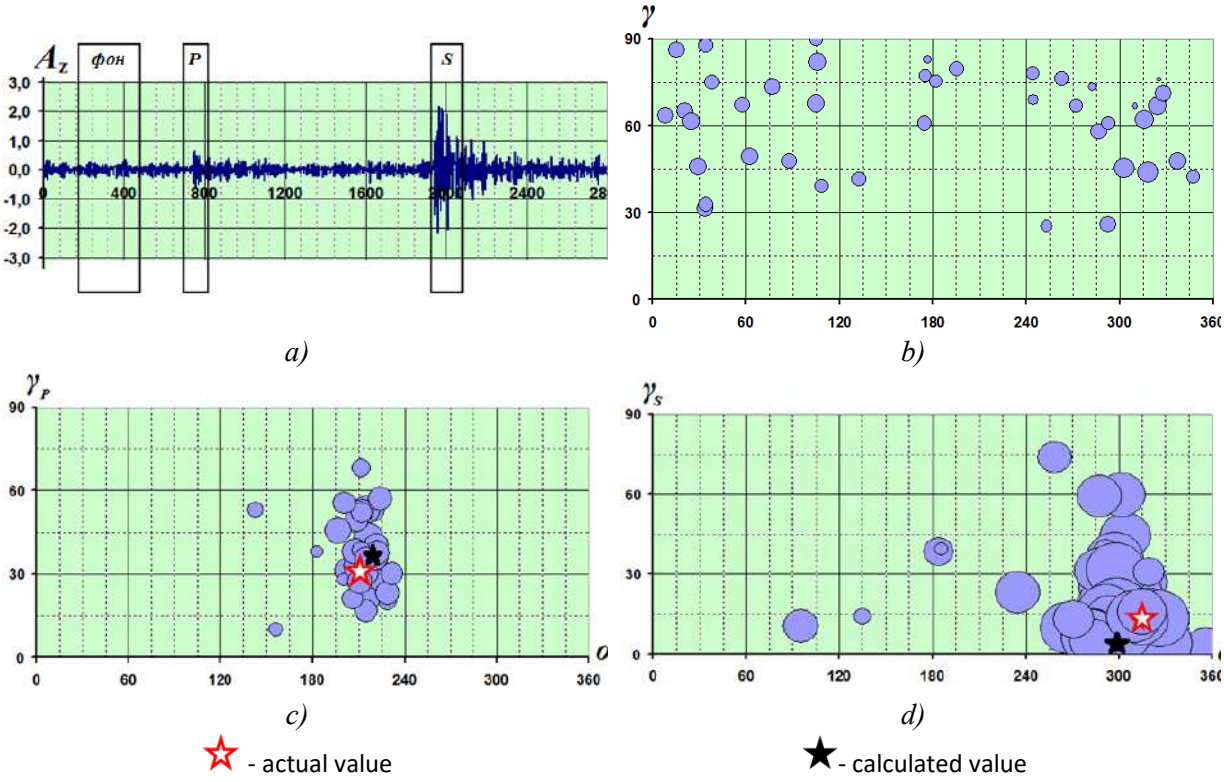


Figure 5: Seismic signal vertical component from an earthquake with center in Chernivtsi region (a) and angular characteristics for background (b), P-wave (c) and S-wave (d) recording sites

Difference obtained between the angles between the directions of the bulk wave exit from the normal (from 90º) is due to the fact that expressions (1,2) use only angular indicators, while not taking into account amplitude features.

3.2. Polarization analysis of a three-component seismic record

Another approach to determining angular characteristics is polarization analysis apparatus (PA) [8]. Trajectory of soil particles during seismic wave propagation has a shape of a strongly elongated ellipsoid, and for background it is close to a sphere. Main purpose of existing approaches to polarization analysis, regardless of implementation of solving function, is to assess degree of linearity in oscillations and determine angular position of large ellipsoidal axis.

In [6], a degree of linearity of a three-component seismic record $\{z_i, n_i, e_i\}$ is determined by a calculation of covariance matrix K

$$K = \begin{vmatrix} \mathrm{cov}(n,n) & \mathrm{cov}(n,e) & \mathrm{cov}(n,z) \\ \mathrm{cov}(e,n) & \mathrm{cov}(e,e) & \mathrm{cov}(e,z) \\ \mathrm{cov}(z,n) & \mathrm{cov}(z,e) & \mathrm{cov}(z,z) \end{vmatrix}. \quad (6)$$

A quadratic shape (ellipsoid) defined by this matrix is reduced to the major axes. Major axis of ellipsoid characterizes orientation in space of full seismic wave displacement vector by angles – azimuth α and angle of exit to bottom surface γ.

Linearity coefficient G (0< G <1) adopted implementation three-component recording is defined as:

$$G = 1 - \frac{b}{c}. \quad (7)$$

where, b and c are values of smallest and largest semi-axis of ellipsoid, respectively.

Other approaches to determining a degree of linearity of a three-component seismic record are approximation of trajectory for soil particles by an ellipsoid, sequential polarization filtering, etc. [12].

Despite different methodological foundations, main goal of known (existing) approaches to polarization analysis is to determine how close oscillations of current (recorded) sample in three-component seismic record are to a certain direction (direction is determined by azimuth and angle of seismic signal component output).

Values of determined angular characteristics for seismic recording of a signal from an earthquake with a center in Chernivtsi region (Figure 5a) using the APA (6) are, respectively, for the P-wave α_P = 210º and γ_P = 34º, for the S-wave α_S = 313º and γ_S = 14º. According to determined angular characteristics of seismic signal components, angle value is 92º, which corresponds to conditions on orthogonality of oscillation by soil particles for bulk waves, in contrast to case on using arithmetic mean azimuth and angle on seismic wave exit to the day surface. However, using polarization analysis apparatus requires significant computational costs and is usually used in post-processing of measurement data of three-component seismic recordings.

3.3. S-wave detection and focal point determination

For implementation of remote monitoring by seismic means in real time, it is necessary to use additional criteria that take into account particularities of seismic signal components from sources in regional area.

Monitoring potential sources of emergencies requires availability of a priori information - position of monitoring object relative to observation point where TCSS is located. Availability of such input data makes it possible to determine angles of seismic wave arrival at surface from an event at a monitored object (object for which continuous remote monitoring by seismic means is implemented) and difference of arrival time between seismic signal components.

Taking into account above-mentioned dynamic (polarization of soil particle oscillations) and kinematic (seismic wave propagation time) features of the main components in seismic signals from sources with foci in regional area, continuous remote monitoring of the PSE based on observations of a TCSS is proposed to be realized by applying coordinated polarization-time filtering, for which a polarization-time model of expected seismic signal from an event with a center in a controlled area (object) is used.

Input data for forming the polarization-time model of expected seismic signal are: expected angular characteristics of first arrival of seismic signal (P-wave) on day surface and distance between observation point and monitored object (area).

Polarization-time model of seismic signal from events with a center in a controlled potential source of emergencies for a three-component seismic station has following general form:

$$F(t) = \Omega(\alpha_P, \gamma_P, \tau_{PS}, t) = \Omega_P(\alpha_P, \gamma_P, t) \cdot \Omega_S(\alpha_S, \gamma_S, t + \tau_{PS}), \quad (8)$$

where, α_P, γ_P – expected azimuth and angle of first component seismic signal (P-wave) exit to daily surface, which are determined by position of the PSE relative to OP;

α_S, γ_S – expected azimuth and angle of S-wave exit to the daytime surface, which are determined by condition of orthogonality to expected direction of P-wave exit to daytime surface;

τ_{PS} – time difference between the arrivals of seismic signal components (volume waves), which is determined from information on distance between PSE and OP using a hodograph (Figure 4);

$\Omega_P(\alpha_P, \gamma_P, t)$ and $\Omega_S(\alpha_P, \gamma_P, t + \tau_{PS})$ – estimation of degree a linearity of sections in the three-component seismic record corresponding to P- and S-waves, respectively.

In order to assess degree of linearity and their compliance with controlled directions of seismic signal components exit to ground surface, it is proposed to define as the ratio between projection of ground vibrations to selected direction and total value of soil particles displacement for accepted realization.

To detect the first arrival of seismic signal (P-wave), decisive function has following form:

$$\Omega_P(\alpha_P,\gamma_P,i)=\frac{\sum_{i=1}^{N}|R_i|\cdot\cos\phi_i}{\sum_{i=1}^{N}|R_i|}, \tag{9}$$

where, R_i is modulus of soil particle displacement vector

$$R_i=\sqrt{z_i^2+n_i^2+e_i^2}\ ; \tag{10}$$

z_i, n_i, e_i – displacement of soil particles in vertical and horizontal (north-south and east-west) channels, respectively;

ϕ_i – angle between expected direction of seismic wave exit and current position of ground displacement vector.

Detection of a seismic record section corresponding to an S-wave is carried out under condition of orthogonal oscillations by soil particles relative a direction towards first arrival of seismic signal on day surface $(\alpha_P,\gamma_P)\perp(\alpha_S,\gamma_S)$. For S-wave detection, a decisive function is defined as:

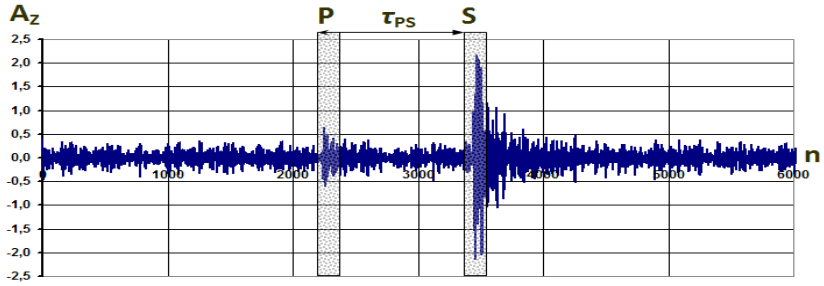
$$\Omega_S(\alpha_P,\gamma_P,\tau_{PS},i)=\frac{\sum_{i=1+\tau_{PS}}^{N}|R_i|\cdot\sin\phi_i}{\sum_{i=1+\tau_{PS}}^{N}|R_i|}\cdot \tag{11}$$

On Figure 6 shows results of applying proposed approach by detecting seismic signal from earthquake with center in Chernivtsi region (16.01.2020, M=2.4).

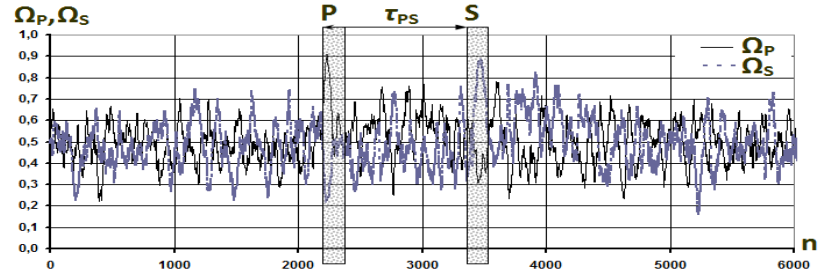
Input data for formation of polarization-time model a seismic signal are:

– distance to seismic event center 281 km, which, in accordance with seismic wave hodograph for regional zone (Figure 4), corresponds to 30 seconds difference in arrival time between P- and S-waves

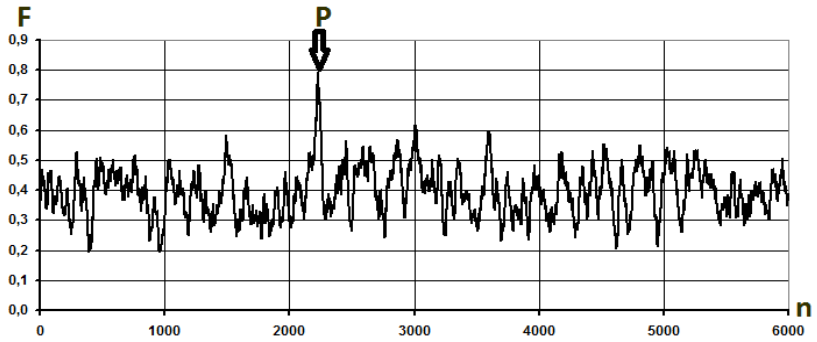
– angles of seismic wave exit on the daily surface are respectively α_P = 210º та γ_P = 34º.



a)



b)



c)

Figure 6: Results application of polarization-time model of the picked seismic signal from earthquake with a center in controlled area: a – seismic signal vertical component from an earthquake with center in Chernivtsi region; b – values of decisive functions for detecting volume waves; c – values of decisive function for detecting seismic event from controlled area

As can be seen from above results, maximums of solving functions for seismic signal components Ω_P and Ω_S correspond to arrivals of P- and S-waves, respectively. Maximum of solving function F(t) corresponds to first arrival of seismic signal P-wave.

On Figure 7 shows a histogram of distribution values for the solving functions proposed by the method p(F) and to the first arrival a seismic signal from the direction corresponding to a monitored object $p(\Omega_P)$. Due to small sampling of earthquake signals from monitoring area, it is proposed to assess signal presence at a probability of false signal detection error at level α=0.001:

$$\alpha_F = \int_{h_F}^{\infty} f(F / H_0)dF = 0{,}001\,,\ \alpha_{\Omega_P} = \int_{h_{\Omega P}}^{\infty} f(\Omega_P / H_0)d\Omega_P = 0{,}001\,. \qquad (12)$$

Under these conditions, threshold for detecting a signal from a seismic event with a center in monitoring area using polarization-time model of expected signal is h_F=0,64. At first arrival of seismic signal coming from direction corresponding to monitoring object $h_{\Omega P}$=0,81.

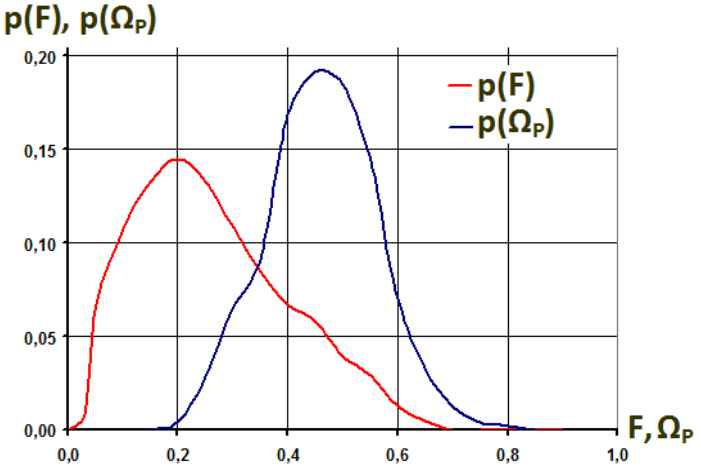


Figure 7: Histogram of distribution values using decision function for polarization-time model application of peaked seismic signal p(F), and for first seismic signal arrival corresponding to direction of monitoring object $p(\Omega_P)$

In Figure 8 shows results of applying polarization-time model with parameters for the earthquake with a center in Chernivtsi region relative to Malyn observation point for processing a three-component seismic signal record from earthquake with a center in Vrancea seismically active zone (Romanian part of the Carpathian Mountains).

A special feature of seismic signals from these areas is proximity in values for the first seismic signal arrival at the daytime surface. Figure 8a, shows arrival of S*-wave for signal from earthquake with a center in Vrancea zone. Values of solving functions for seismic signal components Ω_P and Ω_S correspond to arrival only for the P-wave arrival (Figure 8b). At the same time, values of decision function for the first arrival of ΩP in some areas exceed the threshold value (Figure 8b). A value of decision function F(t) at the time of P-wave arrival is 0.53 (Figure 8c), which doesn't exceed threshold value according to condition (12).

In summary, this approach to processing measured data from a separate three-component seismic station allows detecting seismic signals from events with centers in controlled potentially hazardous objects (areas) by implementing their continuous remote monitoring.

At the same time, time for establishing a fact of an event in the controlled area (at the controlled facility) based on TCSS observations is limited by the propagation time of the S*-wave (Figure 4). For MCSM observation network, using polarization-time model of the detected signal will reduce time for establishing a fact of an emergency event within Ukraine from 15 to 3 minutes.

In order to form a polarization-time model of expected seismic signal from a possible event with a center in a controlled area (object), information on azimuth on the monitoring object relative an observation point where three-component seismic station is deployed, distance between observation point and remote monitoring station is required.

Table 1 shows parameters of polarization-time model seismic signals from the most hazardous facilities (nuclear power plants and hydroelectric facilities of Dnipro cascade) for Malyn observation point (Vorsovka village, Malyn district, Zhytomyr region).

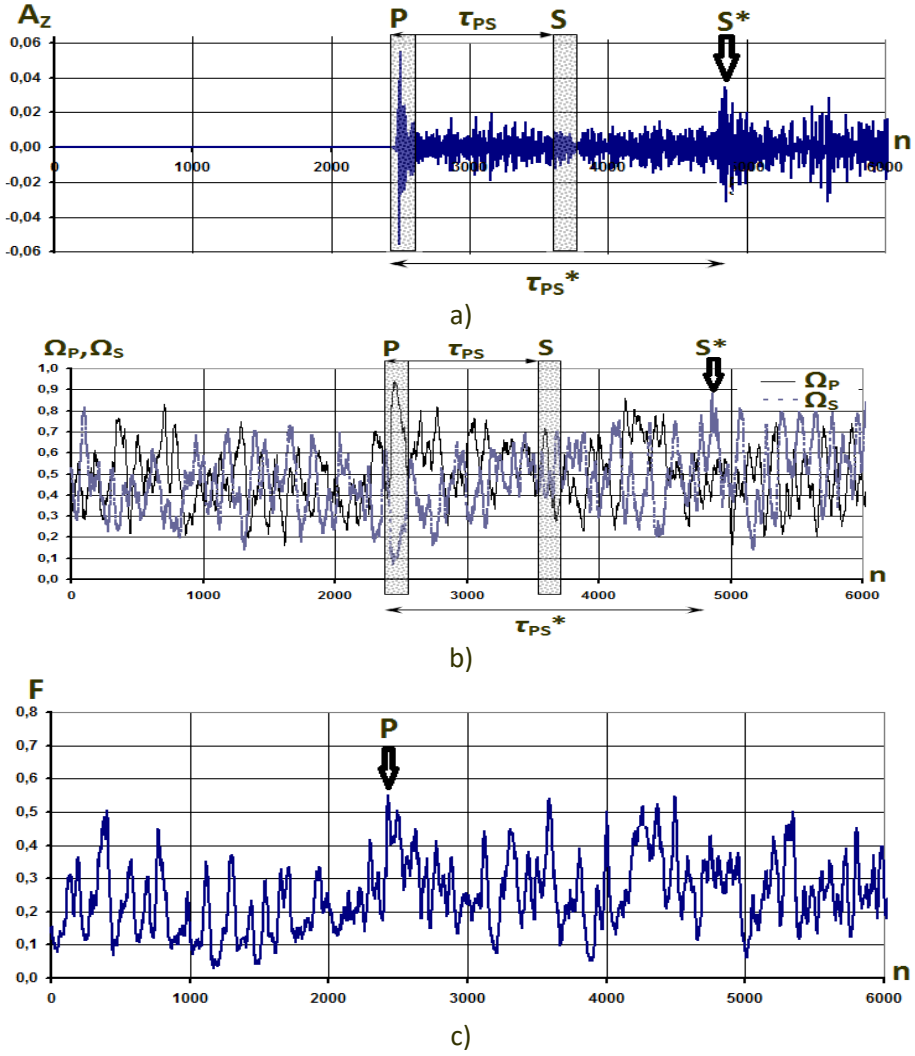


Figure 8: Results of applying polarization-time model of picked seismic signal from an earthquake with a source outside controlled area: a – seismic signal vertical component from an earthquake in the Wrench zone; b – values of decisive functions for detecting bulk waves; c – values of decisive function for detecting seismic

Table 1.

Parameters of polarization-time model seismic signals for nuclear power plants and hydroelectric facilities in Dnipro cascade for Malyn observation station

Monitoring object	Azimuth of the arrival α_P, °	Angle of access daylight surface γ_P, °	Difference in arrival time of bulk waves τ_{PS}, s

South Ukraine Nuclear Power Plant	153,5	37	42
Khmelnytsky Nuclear Power Plant	256,1	30	23
Rivne Nuclear Power Plant	286,2	32	28
Zaporizhzhya Nuclear Power Plant	130,4	44	47
Chernobyl Nuclear Power Plant	39,4	25	14
Kyivska Hydroelectric Power Plant	101,3	25	13
Kanivska Hydroelectric Power Plant	124,8	30	24
Kremenchuk Hydroelectric Power Plant	121,4	35	38
Kamianska Hydroelectric Power Plant	120,7	40	53
Dniprovska Hydroelectric Power Plant	123,9	46	60
Kakhovka Hydroelectric Power Plant	142,4	46	62

4. Conclusion

Prospects for seismic monitoring, as part of fulfilling tasks of information support within civil protection system (CPS) on seismic situation, are development and improvement of methodological and algorithmic approaches for complex processing seismic measurement data in order to promptly provide information on possibility, fact and consequences of a hazardous event.

One of approaches to solving this problem is implementation a continuous remote monitoring by seismic means potential sources of emergency events.

The present study analyzed components of seismic signals of events with centers in the regional area. A relationship is revealed between angular features of main components a seismic signal for events with centers in a regional zone and position of seismic source relative to observation point. Basics are formed and an approach is proposed for implementing remote monitoring using a three-component seismic station for potential sources by natural and man-made disasters. Monitoring potential sources of emergency events by a three-component seismic station is proposed to be realized by applying coordinated polarization-time filtering, for which a polarization-time model of the expected seismic signal from an event with a center in a controlled area (object) is proposed.

Results of applying proposed approach for detecting seismic signals from earthquakes with focal points in a regional area are presented. Relative simplicity of proposed approach allows to realize monitoring of PSE in real time. Application of this approach to the MCSM seismic observation network allows to reduce time to establish a fact of an emergency event within the territory of Ukraine from 15 to 3 minutes.

Parameters of polarization-time model of seismic signals for the most hazardous objects – nuclear power plants and hydroelectric facilities of the Dnipro cascade – are presented.

Main directions of further research are development and improvement a theoretical basis for processing seismic data, which will allow:

1. To identify the nature of a seismic source based on seismic data processing results in an automatic mode, taking into account amplitude-frequency differences in signals with different nature and seismic background features in a region where monitoring equipment is located;
2. Form databases of polarization-time models for potential sources of emergency events on the territory on Ukraine and neighboring countries;
3. Investigate variations of seismicity modes in seismically active zones (areas) for the purpose a timely detection a change in their state and prompt establishment a fact of an emergency.

Identified directions will allow solving the scientific and practical problem of detecting hazards factors related to man-made and natural emergencies by seismic means.

5. References

[1] The Code of Civil Legislation of Ukraine of October 2, 2012, No. 5403-VI. The Voice of Ukraine. 220 (5470), 4 – 20. [in Ukrainian].

[2] Civil Protection Code of Ukraine. No. 5403VI. (2012). Golos Ukrayiny. 220 (5470), 4 – 20.

[3] People's support for the state of technogenic and natural safety in Ukraine [Electronic resource]. Access mode: http://www.dsns.gov.ua/ [in Ukrainian].

[4] Resolution of the Cabinet of Ministers of Ukraine from 9.01.2014 № 11 “On approval of the Regulations on the Unified State System of Civil Protection”, 2014. URL:http://zakon.rada.gov.ua/laws/show/11- 2014-%D0%BF.
[5] Main Center of Special Monitoring. (2010). Seismic Network Main Center of Special Monitoring [20.12.2022]. International Federation of Digital Seismograph Networks. doi: 10.7914/SN/UD
[6] V. M. Vaschenko, I. V. Tolchonov, Y. O. Hordiienko, O. I. Solonets. Statement problem of detection danger factors of emergency situations by seismic means. Trudy “Information processing systems” 2/100 (2012) p. 280-284.
[7] Wang B and Huang L (2022) A Polarization Migration Velocity Model Building Method for Geological Prediction Ahead of the Tunnel Face. Front. Earth. doi: 10.3389/feart.2022.857984
[8] Lewis, W., D. Vigh, A. M. Popovici, and S. Fomel. "Deep learning prior models from seismic images for full-waveform inversion: 87th Annual International Meeting, SEG, Expanded Abstracts, 1512–1517. doi: 10.1190/segam2017-17627643.1.
[9] Anya & Mao, Weijian & Gubbins, David. (2001). Polarization filtering for automatic picking of seismic data and improved converted wave detection. Geophysical Journal International. 147. 227-234. Automatic picking of P and S phases using a neural tree. J Seismol 10, 39–63 (2006). doi:10.1007/s10950-006-2296-6
[10] Y. Hordiienko, V. Tiutiunyk, L. Chernogor and V. Kalugin, "Features of Creating an Automatically Controlled System of Detecting and Identifying the Seismic Signal Bulk Waves from High Potential Events of Technogenic and Natural Origin," *2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, Kharkiv, Ukraine, 2021, pp. 267-272, doi: 10.1109/PICST54195.2021.9772159.
[11] Gentili, S., Michelini, A. Automatic picking of P and S phases using a neural tree. J Seismol 10, 39–63 (2006). doi:10.1007/s10950-006-2296-6
[12] Vilajosana, I., Suriñach, E., Abellán, A., Khazaradze, G., Garcia, D., and Llosa, J.: Rockfall induced seismic signals: case study in Montserrat, Catalonia, Nat. Hazards Earth Syst. Sci., 8, 805–812. doi:10.5194/nhess-8-805-2008, 2008.

Case Driven TLC Model Checker Analysis in Energy Scenario

Vadym Shkarupylo [a,b], Ihor Blinov [c], Valentyna Dusheba [b], Jamil Abedalrahim Jamil Alsayaydeh [d]

[a] *National University of Life and Environmental Sciences of Ukraine, 15, Heroiv Oborony, Str., Kyiv, 03041, Ukraine*
[b] *G.E. Pukhov Institute for Modelling in Energy Engineering of the NAS of Ukraine, 15, General Naumov, Str., Kyiv, 03164, Ukraine*
[c] *Institute of Electrodynamics of the NAS of Ukraine, 56, Peremohy, Av., Kyiv, 03057, Ukraine*
[d] *Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, Melaka, 76100, Malaysia*

Abstract

Today, model checking techniques and corresponding tools are widely applied in diverse case driven scenarios, the safety critical ones in particular. Addressing current situation in Ukraine, an energy domain is among the topical spheres, where safety critical business processes take place. To foster the functional safety of corresponding program-algorithmic solutions, the model checking techniques and related tools are applied to the formal specifications of named solutions. Doing so is not a trivial task: it depends on a particular use case scenario determining the architecture (structure and couplings) of the resulting design artifact. Moreover, the outcomes of formal techniques and tools application directly depend on specification atomicity level chosen – as a tradeoff between the complexity of program-algorithmic constituent addressed to be represented in formal specification and available computational and spatial resources of the computing platform with model checking technique implementation – because of an exponential growth of transition system state space. To this end, to foster the effectiveness of model checking technique application, with respect to a particular case driven scenario, the analysis of broadly applied TLC model checker has been conducted on the basis of a role model from energy domain. Experimentation has been conducted by addressing two alternative implementations of the TLC method. Both – computational and spatial properties – have been covered. To estimate also the domain related spatial expenses on verification, with respect to the number of software threads utilized, the approximation task has been resolved.

Keywords 1

Artifact, formal specification, model checking, safety critical scenario, TLA, TLC, verification

1. Introduction

Nowadays, the complexity level of modern program-algorithmic solutions addressing the scenarios taking place in diverse safety-critical domains, e.g., energy scenarios (Finnish nuclear industry [1]), avionics (satellite operational mode management scenarios [2]), safety critical software as a part of railway control systems [3], etc., typically exceeds the limitations of computational and spatial resources required to successfully apply time-proven formal verification techniques and corresponding instruments in one-to-one manner – due to the exponential growth of transition system state space to be traversed through during an automated formal verification by way of model checking [4]. For instance, considering the avionics scenarios, an exponential growth of both spatial [5] and computational [6] expenses has been faced. To diminish named effect, different approaches have been

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: shkarupylo.vadym@nubip.edu.ua (V. Shkarupylo); blinovihor@gmail.com (I. Blinov); vdusheba@gmail.com (V. Dusheba); jamil@utem.edu.my (J. A. J. Alsayaydeh)
ORCID: 0000-0002-0523-8910 (V. Shkarupylo); 0000-0001-8010-5301 (I. Blinov); 0000-0002-8929-3625 (V. Dusheba); 0000-0002-9768-4925 (J. A. J. Alsayaydeh)

proposed previously. Among those, there is an attempt to vary the atomicity level to be applied in formal specification by coupling the specified properties into the groups, i.e., the "hyper-properties", expressing the relations between the constituents; to this end, the broadly applied Temporal Logic of Actions (TLA; by Leslie Lamport) and the corresponding TLA Checker (TLC) have been brought to the use [7]. Alternatively, the decomposition based approach can be applied: formal specification is decomposed into sub-specifications to be subsequently verified with the time proven Event-B method [8]. Yet another promising direction to proceed is to decrease the model checking related computational expenses via a mu-calculus based abstraction reduction technique implementation [2]. In addition, the bounded model checking approach can be applied, e.g., by bringing the mathematical apparatus of the discrete time Markov chains (DTMC), paired with the probabilistic computation tree logic (PCTL) and probabilistic symbolic model checker (PRISM) [9]. Moreover, to diminish the effect of transition system state space exponential growth, the Reduced Ordered Binary Decision Diagrams (ROBDDs) are shown to be the proven instrument – more than 10^{20} states can be encompassed [10].

Thus, it can be seen that the problem of an exponential growth of the transition system state space is multi-dimensional, and can be approached diversely. At the same time, it can be noted that implementation related aspects taking place during the model checking are still lacking the attention of the research community – in terms of the influence of a particular implementation on the related computational and spatial expenses on certain case driven model checking task resolving.

In our work, the TLC model checker has been chosen to be the instrument to be applied during the case study – because of its wide usage in diverse safety critical scenarios: the grounding is provided below (in the Section 3).

Rest of the paper is organized as follows: in the Section 2, the problem statement is made; Section 3 is devoted to the related work analysis; in the Section 4, the case study addressing the scenario from the energy domain is described; Section 5 contains the conclusion and thoughts on the future work; the acknowledgments are given in the Section 6; in the Section 7, the references are provided.

2. Problem statement

Taking into consideration the aforesaid, paper addresses the problem of an exponential growth of the transition system state space – in terms of the corresponding computational and spatial expenses. An attempt to estimate named expenses, with respect to a case driven scenario from the energy domain has been made. To this end, the broadly used TLC model checker (method) has been utilized as an instrument. Two alternative implementations of the TLC have been investigated in terms of the related computational and spatial expenses: the BFS (Breadth-first Search) one – implemented by the default; the alternative DFS (Depth-first Search) implementation. As it can be seen from the naming, these implementations differ by the method applied to traverse through the states of a transition system.

Let the model checking task is formalized as follows [4]:

$$M, b \models \psi, \tag{1}$$

where M – the transition system – Kripke structure over a set of the atomic prepositions AP; b – "behavior" – as a sequence of states to be traversed through during the model checking with the TLC method; ψ – temporal formula, in TLA+ syntax, prescribed in the formal specification. To positively state regarding specification consistency, ψ has to be "true" for each element of b.

In given paper, the model checking task has been resolved with respect to a case driven scenario taking place in energy domain, and specified with a role model of European identification codes registry update process.

The idea is to estimate and compare different implementations of the TLC method, by grounding on a particular case driven scenario – with respect to corresponding computational and spatial expenses that take place during the model checking. This approach is an attempt to discover and

assess the factors – e.g., the architecture (structure and couplings) specified in the formal model, number of state variables, etc. – affecting the computational and spatial costs of formal verification with a particular implementation of certain model checker (the TLC in our case). Moreover, an attempt to estimate the outcome of bringing the multithreading to the implementations of the TLC method has also been made.

3. Related work

When addressing specification atomicity concept, it first needs to be noted that, prior to being represented formally, the specified properties are commonly represented in certain textual or graphical form, e.g., with BPMN-notation (Business Process Model and Notation). To verify related spatial properties, the sBPMN Verification Framework has been proposed [11]: it is based on the TLC model checker application to formal specifications written in the TLA+ formalism of the TLA temporal logic [12]. The distinctive features of named formalism are the modularity and mathematical strictness. It can be addressed as a propositional logic coupled with the temporal operators: X (Next) and G (Globally). Because of the fact that model checkers (the implementations of model checking techniques) are the instruments to be applied to formal specifications, but not to the initial artifacts directly, there is also a necessity to control the adequacy of specifications. It can be accomplished, for instance, by comparing the properties of the transition system retrieved from the initial artifact – e.g., block-diagram, UML (Unified Modeling Language) activity diagram, etc. – and from the resulting formal specification [5]. In addition, to formally check the abstractions applied in specification, a deductive verification based technique can be utilized prior to the model checking [13]. Moreover, similarly to [11], from the architectural viewpoint, the TLC and corresponding tools have successfully been applied to discover the inconsistencies that may occur during the SDN (Software-defined Networking) topology and related policies changes [14].

As a scenario representing the necessity of chosen specification abstraction level adoption, i.e. atomicity level varying, an industrial distributed Taurus database formal specification synthesis and verification process can be considered: named TLC model checker has been successfully applied to consistency checking task resolving [15]. The obtained results have once again shown the effectiveness of model checking technique to be utilized for design flaws discovery at the design stage of engineering process, prior to the validation, e.g., testing in particular. In addition, in case there are no design flaws that have been discovered while model checking, verification outcome can be approached differently: either be treated as the design solution consistency approval, or as an indicator that specification atomicity level applied needs to be shifted. Moreover, dealing with the processes taking place in the large-scale distributed software systems, the task of eventual consistency over the data replicas maintenance arises: formal instruments (TLC, TLA, TLA+) have been utilized within the MET (Model Checking-driven Explorative Testing) framework – in an attempt to construct the "bridge" between the model checking being applied at the design stage of engineering process and the validation by way of testing: to address a tradeoff between the exhaustive nature of model checking facing the exponential growth of transition system state space and case-driven testing [16]. To diminish the effect of named exponential growth, the compositional model checking techniques can be applied, e.g., the Interaction-Preserving Abstraction (IPA) framework addressing the specifications written in TLA+ [17]. Moreover, novel TLA+ Debugger utility makes it possible not only debug the temporal formulas evaluations, but also inspect states and transitions of transition system [18].

4. Case study

As a demonstrative problem domain, where diverse safety-critical scenarios take place, modern electricity market of Ukraine [19], adjusted in accordance with the harmonized European electricity market model, has been considered [20]. European identification codes registry update process has been addressed as a case study (Figure 1).

Considering the UML (Unified Modeling Language) diagram, depicted in the Figure 1, the TLA+ specification atomicity level has been chosen to be applied in “one-to-one” manner. Similar step has been made in the previous case study, where the avionics scenario has been approached [5].

Among the distinctive features of current case study, there are, in particular, the different type of the initial artifact (UML activity diagram instead of block diagram), and also the different problem domain.

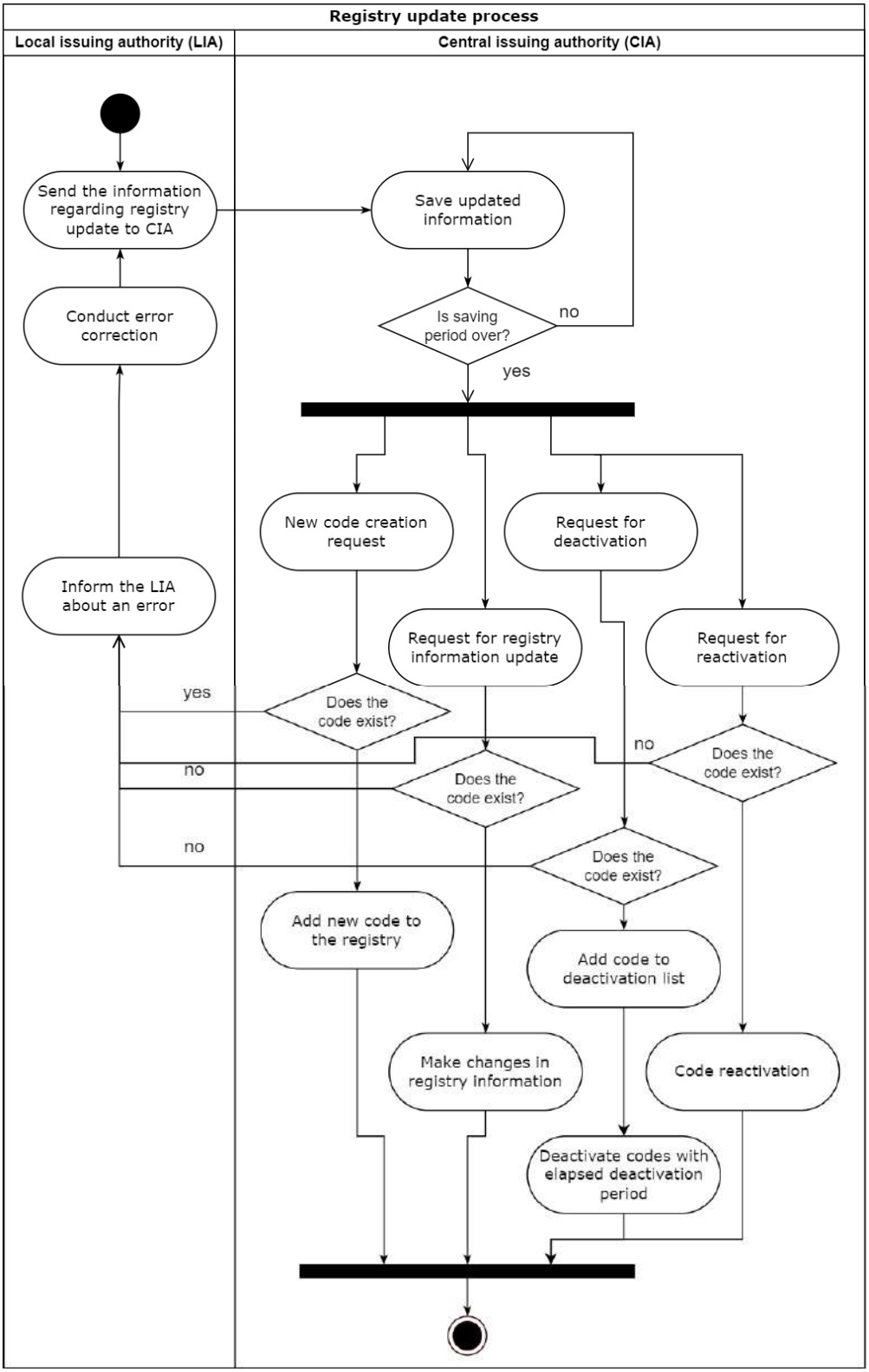


Figure 1: Fragment of a role model addressing the European identification codes registry update process

In the Figure 1, a fragment of the complete role model is depicted as the UML activity diagram. A complete role model encompasses 23 activities, plus 6 conditional operators. Named model is approached as an artifact – an entity with the architecture (structure and couplings) and the content [21]. These constituents are represented in the formal TLA+ specification with respect to the approach described below.

4.1. Formalization and experimentation

To synthesize the formal specification suitable for verification in an automated manner, the following two staged approach has been applied:

1. To create the architectural prototype (encompassing both algorithmic structure and the couplings; in a pseudo code-like manner) of yet to be synthesized TLA+ specification, the PlusCal algorithmic language has been used [22]. An outcome of this stage is an input data for the following one. The PlusCal specification is treated as the preliminary artifact.
2. To generate the resulting artifact – the TLA+ specification – from the PlusCal pseudo code, the TLA+ Toolbox IDE (Integrated Development Environment) has been utilized [12].

The TLA+ specification is addressed to be the input data for the TLC model checker intended to be applied in an automated manner.

4.1.1. Approach to the PlusCal specification synthesis

To obtain the preliminary artifact, the following approach has been applied:

1. Each of the aforementioned activities (Figure 1) has been represented in the PlusCal specification with a state variable. Named variables have been grouped within corresponding set V': $|V'| = 23$.
2. Moreover, additional three state variables have been created to represent conditional operators: initial conditional operator, prior the fork construct; group of 4 conditional operators after the fork construct (Figure 1); final conditional operator which is out of the scope of diagram fragment depicted. As an outcome, the complementary state variables have been introduced with the following set: V'': $|V''| = 3$.
3. Yet another state variable has been used to initiate/terminate the computational process formalized in the PlusCal specification.

Thus, with respect to the approach applied, the resulting state variables set has been obtained as follows: $V = V' \cup V'' \cup \{v^*\}$, where $v^* \in V$ is the initiate/terminate state variable.

4.1.2. TLA+ specification synthesis and checking

It has taken 348 rows of pseudo code to represent the role model in PlusCal. After that, the instruments of TLA+ Toolbox IDE have been applied to generate the resulting TLA+ specification from the PlusCal representation. As an outcome, the TLA+ specification has been synthesized containing 508 rows of code. Thus, it can be seen that the initial PlusCal artifact is significantly more concise comparing to the resulting TLA+ specification, while preserving the same architectural constituent.

The consistency of synthesized TLA+ specification has been proven in an automated manner – with the TLC applied. To state about the consistency of the initial role model (Figure 1), the aforementioned adequacy checking technique has been used [5].

Spatial properties of a transition system traversed through during an automated TLC checking can be represented with the elements of Kripke structure M (1) [4], over a set of atomic prepositions

$AP = V \times D$, where, in our case, $D = \{0,1\}$ – set of allowable state variable values, i.e., set of Boolean values: $M = \langle S_0, S, R, L \rangle$, where $S_0 \subset S$ – set of initial states; S – finite set of states; $R \subseteq S^2$ – total set of transitions: $\forall s \in S \; \exists s' \in S : (s, s') \in R$; $L : S \to 2^{AP}$ – states labeling function.

4.1.3. Transition system analysis

In our case, $S_0 = \{s_0, s_1\}$, where $s : V \to D$: $L(s_0) \Delta L(s_1) = \{(v'', 0), (v'', 1)\}$, where $v'' \in V'' \subset V$ state variable represents the group of four identical conditions after the fork construct in the Figure 1. Depending on whether the identification code exists initially, we can have either $s_0 \in S_0$ or $s_1 \in S_0$ initial state.

To form the AP set, a dichotomy principle has been applied: $AP = AP' \cup AP''$, where $AP' = V \times \{0\}$, $AP'' = V \times \{1\}$. Elements of these subsets have been approached as follows: $(v_i, 0) \in AP'$ – i-th activity has not been accomplished yet ($i = 1,2,...,27$); $(v_i, 1) \in AP''$ – i-th activity has already been accomplished.

Transition system (TS) spatial properties discovered during an automated formal verification with the TLC model checker are as follows: "TS depth" – 28 – number of sequential transitions from certain initial state ($s_0 \in S_0$ or $s_1 \in S_0$) to the final one; total number of distinct states found – $|S| = 290$. These properties have been treated as the indexes of model checking task complexity.

4.1.4. Experimentation and obtained results

Experimentation has been conducted on the following platform: CPU – AMD Ryzen R5 2400g; RAM – 16 GB; Java Runtime Environment, version "1.8.0_241"; TLC version: 2.14 of 10 July 2019.

During this, two alternative implementations of the TLC model checker have been encompassed: the BFS- and the DFS-based ones. It needs to be noted, though, that the DFS-implementation of the TLC is coupled with a significant drawback in terms of the automation – "TS depth" has to be discovered first, and then be specified manually. On the contrary, default BFS-implementation does not need to be accompanied with the "TS depth" parameter.

Results of previous experimentations have shown, that, depending on the architectural plane of specification, there is a bound, in terms of the number of state variables, when certain TLC implementation is more efficient in terms of the corresponding computational expenses [6, 23].

Adequacy of the resulting TLA+ specification has been proven with the aforementioned technique [5]. Two alternative implementations (BFS and DFS) of the TLC model checker have been applied, in both single and multithreaded manner.

Let $tn = 2^0, 2^1, ..., 2^3$ be the number of software threads utilized for the TLC model checking. Let $f(tn)$ be the related computational expenses. Let the speedup from multithreading implementation is calculated as follows: $\alpha(tn) = (f(1) / f(tn))$. Obtained experimental results are provided in the Table 1.

Table 1

Computational expenses on the automated formal verification with the TLC model checker

tn	$f_{BFS}(tn)$, ms	$\alpha_{BFS}(tn)$	$\lvert S^*_{BFS} \rvert$	$f_{DFS}(tn)$, ms	$\alpha_{DFS}(tn)$	$\lvert \overline{S^*_{DFS}} \rvert$	ξ_{DFS}
2^0	1214.000	1.000	336	638.400	1.000	4194.000	1.000
2^1	1157.800	1.049	336	632.900	1.009	5382.890	1.283
2^2	1149.100	1.056	336	665.300	0.960	7267.050	1.733
2^3	1143.800	1.061	336	682.700	0.935	8209.850	1.958

In the Table 1, each numerical value is an arithmetical average of 10^2 measures; $f_{BFS}(tn)$ – computational expenses related with the BFS implementation of the TLC method; $f_{DFS}(tn)$ – computational cost of the alternative DFS implementation; $\alpha_{BFS}(tn)$ – speedup from bringing multithreading to the BFS implementation; $\alpha_{DFS}(tn)$ – speedup taking place for the DFS implementation of the TLC; $\left|S^*_{BFS}\right| = const$ – total number of states generated while the BFS model checking to construct the resulting transition system with $|S| = 290$ states; $\left|\overline{S^*_{DFS}}\right|$ – average total number of states generated with the alternative DFS implementation (an average value has been calculated because of the non-deterministic nature of the DFS implementation when the multithreading is applied); ξ_{DFS} – model checking task spatial complexity relative estimation, depending on the number of threads utilized; is calculated conceptually similarly to $\alpha(tn)$: $\xi_{DFS} = \left(\overline{S^*_{DFS_1}} \Big/ \overline{S^*_{DFS_{tn}}}\right)$.

In the Table 1, it can be seen that the DFS implementation of the TLC method has appeared to be significantly more effective in terms of the related computational expenses, when comparing to the default BFS alternative: from about 1.902 times (for $tn = 2^0$) to about 1.675 times (for $tn = 2^3$). It does correspond to the results obtained previously [6, 23], and it can be considered as a significantly viable argument in favor of the DFS implementation when the iterative approach to verification takes place [24]. On the contrary, when addressing the spatial properties of the model checking task resolved with a particular TLC implementation, the picture changes drastically – in terms of the ratio between the numbers of transition system states generated ($\left|S^*_{BFS}\right|$ and $\left|\overline{S^*_{DFS}}\right|$ values): $\left(\left|\overline{S^*_{DFS}}\right| \Big/ \left|S^*_{BFS}\right|\right) \in [12.482; 24.434]$. The BFS implementation becomes to be significantly more preferable.

With respect to the multithreading, the BFS implementation once again looks to be having a significant advantage in terms of the spatial aspect – because of the constant value of $\left|S^*_{BFS}\right|$ index, regarding the number of software threads utilized. At the same time, when dealing with the alternative DFS implementation, the value of $\left|\overline{S^*_{DFS}}\right|$ index rises rapidly with the increase of the number of concurrently acting threads.

In an attempt to estimate the growth of the DFS-related spatial expenses (from the number of software threads utilized), an approximation task has been resolved on the basis of tn and ξ_{DFS} indexes (Figure 2). As an outcome, the following estimation function $\xi'_{DFS}(tn)$ has been obtained:

$$\xi'_{DFS}(tn)^{-1} = (a + b/tn), \tag{2}$$

where $a = 0.435$, $b = 0.603$; determination coefficient $R^2 = 0.986$.

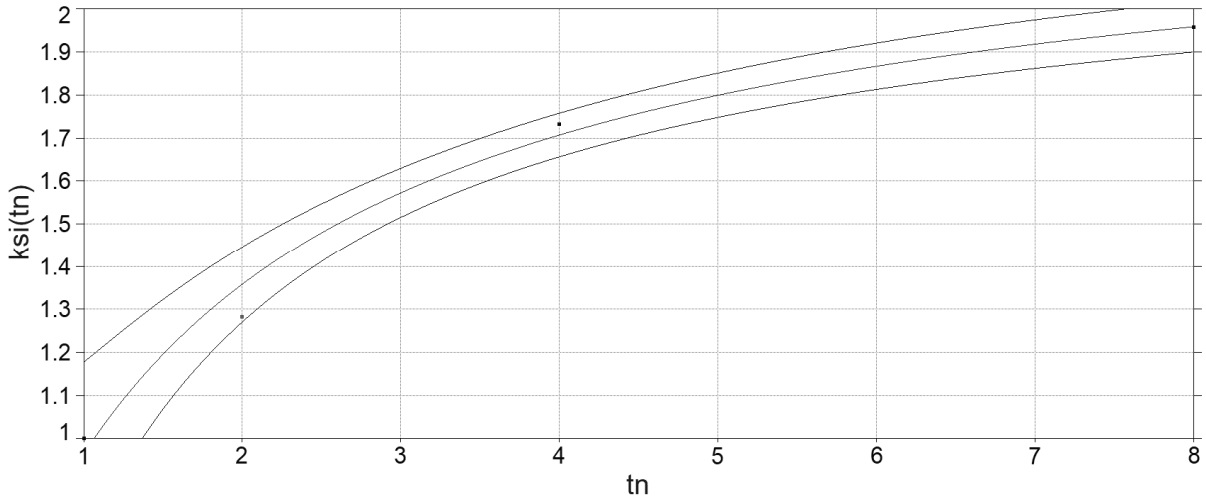


Figure 2: DFS related spatial expenses growth estimation, with respect the number of threads utilized

In the Figure 2, confidence intervals have been built for the confidence probability 0.95. Expression (2) can be used as an estimation of spatial expenses growth, with the increase of the number of concurrently acting threads.

In terms of the computational expenses, it can be seen, in the Table 1, that bringing multithreading to the default BFS implementation does not lead to a significant speedup. Moreover, the major "leap ahead" has been faced while shifting from $tn = 2^0$ to $tn = 2^1$ – about 5% improvement. These results and concluding remarks conform to the ones obtained previously, when an avionics safety critical scenario has been addressed [25]. On the contrary, in accordance with the data from the Table 1, when dealing with multithreaded DFS implementation, the outcome is contradictory: in case of $tn = 2^1$, just a minor speedup (about 1%) has been faced; at the same time, by further increasing the tn value, the results have appeared to be even worse, an opposite trend has been revealed. On the other hand, previously obtained results from the avionics domain have shown a significant positive trend (more than two times speedup for the case of $tn = 2^2$), with a slight decrease for the case of $tn = 2^3$ [25]. Such contradictory outcomes for different case scenarios prompt an assumption that DFS implementation is vastly case sensitive, depending on the number of state variables and the architectural plane of formal specification.

To summarize the distinctive features of both implementations of the TLC method, with respect to the energy domain scenario considered, the following conclusions can be formulated:

1. Default BFS implementation of the TLC model checker can be considered to be the more preferable solution in terms of the following aspects: no need to specify the depth of search space – a definitive argument in terms of the automation, when comparing to the DFS alternative, where named parameter has to be specified manually; constant spatial expenses on the model checking tasks resolving, in terms of the multithreaded implementation.
2. Under proper circumstances, the alternative DFS implementation of the TLC method provides a significant verification related time costs reduction. By encompassing also the results of previous experimentations (both synthetic and domain related – avionics) [5, 6, 23, 25], these circumstances (factors) have been discovered to be the number of state variables and the architectural plane of formal specification.

These concluding thoughts provide the background to further increase the set of case driven scenarios with the TLC model checker implementations utilized, in an attempt to work out and generalize the rules (recommendations) for a particular TLC implementation practical usage, in terms of the related computational and/or spatial expenses decrease.

5. Conclusion

Thus, broadly used TLC model checker has been investigated in given paper on the basis of the case driven scenario taking place in modern energy domain: a role model describing the European identification codes registry update process has been addressed as a design artifact. The consistency of corresponding program-algorithmic constituent has been proven with the named method applied. To prove the credibility of the results obtained, the adequacy of the resulting formal specification has been checked with respect to previously introduced technique.

Both alternative implementations of the TLC method have been considered, including the multithreading aspect: the BFS and the DFS implementations. It has been found out that while the BFS-related outcomes conform to the results and assumptions that have been made previously (on the basis of synthetic and domain related – avionics – scenarios), the DFS-related ones, on contrary, have appeared to be demonstrating just a minor speedup (about 5%) in the case of two threads applied; with further increase of the number of threads utilized, even the negative speedup factor has been faced. These results have led us to a conclusion that the DFS implementation of the TLC method significantly depends on both the number of state variables taking place in formal specification and the architectural plane of the latter. At the same time, in general, on the basis of the case study conducted, the DFS implementation has appeared to be about 1.675 – 1.902 times more efficient in terms of the related time costs, when comparing to the default BFS implementation of the TLC. On the contrary, when addressing the spatial aspects (the number of transition system states generated), the DFS implementation has appeared to be significantly worse comparing to the BFS alternative: from about 12.482 to about 24.434 times.

With an intention to assess the character of the DFS-related spatial expenses growth, with respect to the number of concurrently acting software threads, the approximation task has been resolved, and corresponding estimating function has been obtained.

Future work is focused on an attempt to formulate the recommendations to both TLC implementations (the BFS and the DFS) practical usage, depending on the number of state variables taking place in the formal specification and on the peculiarities of the architectural plane of the latter.

6. Acknowledgements

Paper has been prepared on the basis of the results obtained in accordance with the tasks of the following research works: 0120U102683 “Development of specialized computer technologies for modeling and processing of operational information in energy problems”; 0121U110615 “Development of methods and means for safety-critical systems designing process artifacts verification”, carried out by the Department of Mathematical and Computer Modeling of the G.E. Pukhov Institute for Modeling in Energy Engineering of the National Academy of Sciences of Ukraine, with a contribution of the Institute of Electrodynamics of the National Academy of Sciences of Ukraine. Paper also addresses the tasks of W911NF-22-2-0153 research work carried out by the G.E. Pukhov Institute for Modeling in Energy Engineering of the National Academy of Sciences of Ukraine and funded by the US Army Engineer Research and Development Center (ERDC).

7. References

[1] A. Pakonen, T. Tahvonen, M. Hartikainen, M. Pihlanko, Practical applications of model checking in the Finnish nuclear industry, in: Proc. 10th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies, San Francisco, CA, USA, June 11–15, 2017, pp. 1342–1352.

[2] V. Nardone, A. Santone, M. Tipaldi, D. Liuzza, L. Glielmo, Model checking techniques applied to satellite operational mode management, IEEE Systems Journal, vol. 13(1) (2019) 1018–1029. doi: https://doi.org/10.1109/JSYST.2018.2793665.

[3] S. Resch, M. Paulitsch, Using TLA+ in the development of a safety-critical fault-tolerant middleware, in: 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Toulouse, France, 23-26 October 2017, pp. 146–152. doi: https://doi.org/10.1109/ISSREW.2017.43.

[4] E. M. Clarke, O. Grumberg, D. Kroening, D. Peled, H. Veith, Model checking: 2nd ed. Massachusetts: The MIT Press, 2018.
[5] V. Shkarupylo, J.A.J. Alsayaydeh, I. Tomičić, A. Chemeris, V. Dusheba, A technique for checking the adequacy of formal model, ARPN Journal of Engineering and Applied Sciences, 16 (2021) 1707–1719. URL: http://www.arpnjournals.org/jeas/research_papers/rp_2021/jeas_0821_8670.pdf.
[6] V. Shkarupylo, I. Blinov, A. Chemeris, V. Dusheba, J. Alsayaydeh, A. Oliinyk, Iterative approach to TLC model checker application, in: Proc. 2021 IEEE KhPI Week on Advanced Technology, Kharkiv, Ukraine, September 13–17, 2021, pp. 283–287. doi: https://doi.org/10.1109/KhPIWeek53812.2021.9570055.
[7] L. Lamport, F. B. Schneider, Verifying hyperproperties with TLA, IEEE Computer Security Foundations Symposium, CSF 2021, IEEE (2021). doi: https://doi.org/10.1109/CSF51468.2021.00012.
[8] K. Kraibi, R.B. Ayed, J. Rehm, S. Collart-Dutilleul, P. Bon, D. Petit, Event-B Decomposition analysis for systems behavior modeling, in: Proc. 14th International Conference on Software Technologies, ICSOFT 2019, Prague, Czech Republic, July 26–28, 2019, vol. 1, pp. 278–286. doi: https://doi.org/10.5220/0007929602780286.
[9] H. Gao, W. Huang, X. Yang, Applying probabilistic model checking to path planning in an intelligent transportation system using mobility trajectories and their statistical data, Intelligent Automation and Soft Computing, vol. 25(3) (2019) 547–559.
[10] A. Biere, A. Cimatti, E.M. Clarke, O. Strichman, Y. Zhu, Bounded model checking, Advances in computers, vol. 58(11) (2003) 117–148.
[11] R. Saddem-Yagoubi, P. Poizat, S. Houhou, Business processes meet spatial concerns: the sBPMN verification framework, in: M. Huisman, C. Păsăreanu, N. Zhan (Ed.), Formal Methods, FM 2021, Lecture Notes in Computer Science, vol. 13047, Springer, Cham, 2021, pp. 218–234. doi: https://doi.org/10.1007/978-3-030-90870-6_12.
[12] M. A. Kuppe, L. Lamport, D. Ricketts, The TLA+ toolbox, in: 5th Workshop on Formal Integrated Development Environment, F-IDE 2019, Porto, Portugal, October 7, 2019, EPTCS 310, 2019, pp. 50–62. doi: http://doi.org/10.4204/EPTCS.310.6.
[13] J. Amilon, C. Lidström, D. Gurov, Deductive verification based abstraction for software model checking, in: T. Margaria, B. Steffen (Eds.), Leveraging Applications of Formal Methods, Verification and Validation. Verification Principles, ISoLA 2022, Lecture Notes in Computer Science, vol. 13701, Springer, Cham, 2022. doi: https://doi.org/10.1007/978-3-031-19849-6_2.
[14] Y.-M. Kim, M. Kang, Formal verification of SDN-based firewalls by using TLA+, IEEE Access, 8 (2020) 52100–52112. doi: https://doi.org/10.1109/ACCESS.2020.2979894.
[15] S. Gao et al., Formal verification of consensus in the Taurus distributed database, in: M. Huisman, C. Păsăreanu, N. Zhan (Eds.), Formal Methods, FM 2021, Lecture Notes in Computer Science, vol. 13047, Springer, Cham, 2021. doi: https://doi.org/10.1007/978-3-030-90870-6_42.
[16] Y. Zhang, Y. Huang, H. Wei, X. Ma, MET: Model checking-driven explorative testing of CRDT Designs and Implementations: report, 2022. doi: https://doi.org/10.48550/arXiv.2204.14129.
[17] X. Gu et al., Compositional model checking of consensus protocols via interaction-preserving abstraction, in: 2022 41st International Symposium on Reliable Distributed Systems (SRDS), Vienna, Austria, September 19–22, 2022. doi: https://doi.org/10.1109/SRDS55811.2022.00018.
[18] M. A. Kuppe, The TLA+ debugger, in: P. Masci, C. Bernardeschi, P. Graziani, M. Koddenbrock, M. Palmieri (Eds.), Software Engineering and Formal Methods, SEFM 2022 Collocated Workshops, SEFM 2022, Lecture Notes in Computer Science, vol. 13765, Springer, Cham, 2023. doi: https://doi.org/10.1007/978-3-031-26236-4_15.
[19] I.V. Blinov, Ye.V. Parus, H.A. Ivanov, Imitation modeling of the balancing electricity market functioning taking into account system constraints on the parameters of the IPS of Ukraine mode, Tekhnichna elektrodynamika, 6 (2017) 72–79. doi: https://doi.org/10.15407/techned2017.06.072.
[20] I. Blinov, S. Tankevych, The harmonized role model of electricity market in Ukraine, in: 2016 2nd International Conference on Intelligent Energy and Power Systems, IEPS 2016 Conference Proceeding, Kyiv, Ukraine, 07–11 June 2016. doi: https://doi.org/10.1109/IEPS.2016.7521861.
[21] M. Broy, A logical approach to systems engineering artifacts and traceability: from requirements to functional and architectural views, in: M. Broy, D. Peled, G. Kalus (Eds.), vol. 34:

Engineering Dependable Software Systems, NATO Science for Peace and Security Series – D: Information and Communication Security, IOS Press, 2013, pp. 1–48. doi: https://doi.org/10.3233/978-1-61499-207-3-1.
[22] H. Alkayed, H. Cirstea, S. Merz, An extension of PlusCal for modeling distributed algorithms, TLA+ Community Event 2020, Oct. 2020, Freiburg (online), Germany. URL: https://hal.inria.fr/hal-03143502/.
[23] V.V. Shkarupylo, I. Tomičić, K.M. Kasian, The investigation of TLC model checker properties, Journal of Information and Organizational Sciences, 1 (2016) 145–152. doi: https://doi.org/10.31341/jios.40.1.7.
[24] E. Mercer, K. Slind, I. Amundson et al., Synthesizing verified components for cyber assured systems engineering, Software and Systems Modeling, 2023. doi: https://doi.org/10.1007/s10270-023-01096-3.
[25] V.V. Shkarupylo, I.V. Blinov, A.A. Chemeris et al., On applicability of model checking technique in power systems and electric power industry, in: A. Zaporozhets (Ed.), Systems, Decision and Control in Energy III. Studies in Systems, Decision and Control, vol. 399. Springer, Cham, 2022. doi: https://doi.org/10.1007/978-3-030-87675-3_1.

Similarity Metric of Categorical Distributions for Topic Modeling Problems with Akin Categories

Serhiy Shtovba[a], Mykola Petrychko[b] and Olena Shtovba[b]

[a] *Vasyl' Stus Donetsk National University, 600-richchia str., 21, Vinnytsia, 21021, Ukraine*
[b] *Vinnytsia National Technical University, Khmelnytske Shose, 95, Vinnytsia, 21021, Ukraine*

Abstract

Estimating a level of similarity of two objects is a common problem in pattern recognition, clustering, and classification. Among these problems can be reviewer recommendation, similar text documents analysis, human pose detection in video, species distribution clustering, recommendation in internet-shops etc. In case of categorical attributes an object is described as a distribution of membership degrees over categories. Similarity metrics of such distributions are usually defined as a superposition of objects' similarities for each category. Most often it is a sum of similarities in separate categories. In addition to that each category is considered independently and in isolation from the others. Some practical problems have categories that are akin. Therefore, it is expedient to consider objects' similarity not only directly, as a similarity between equivalent categories, but it is also necessary to consider an indirect similarity, cross-similarity through akin categories. It is such similarity metric of two categorical distributions that accounts for the kinship of different categories is proposed in this paper. The metric has two components. The first component is defined as Czekanowski metric. It defines a direct similarity of categorical distributions as a sum of intersection of distributions' membership degrees of two objects. After the intersection the remaining residuals are accounted for in the second component of the metric. The second metric's component is defined as element-wise product of two matrices: matrix of residuals composition from memberships of two categorical distributions and matrix of categories' paired kinship. It is assumed that kinship indices for each pair of categories are known. As a result, with a large number of categories the overall noisy contribution from weakly akin categories is prominent. Therefore, it is proposed to filter the noise and account only for contribution from strongly akin categories.

Keywords 1

Categorical distribution, akin categories, kinship coefficient, similarity metric, Czekanowski metric, topic modeling, reviewer recommendation, generalized Pareto distribution

1. Introduction

Topic modeling is a machine learning technology for summarization of huge volumes of information [1]. A popularity of topic modeling is increasing drastically over the last decade. Annual number of research papers is doubling every 3-4 years.

Estimating the level of similarity between two objects is a common task in topic modeling. For estimating the level of similarity it is necessary to describe each object as a vector of attributes. If objects are defined in a metric space, then each attribute is defined in a numerical scale. For example, in Fisher's Iris dataset, each flower is described with four attributes, namely the petal width, the petal length, the sepal width, and the sepal length. Object's attributes can be categorical as well, in this case they are defined as a distribution of membership degrees over categories. Such a categorical

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: s.shtovba@donnu.edu.ua (S. Shtovba); petrychko.myckola@gmail.com (M. Petrychko); olenashtovba@vntu.edu.ua (O. Shtovba)
ORCID: 0000-0003-1302-4899 (S. Shtovba); 0000-0001-6836-7843 (M. Petrychko); 0000-0003-1418-4907 (O. Shtovba)


representation of objects is often used in topic modeling problems. For the Fisher's Iris dataset, the result of flower classification can be represented as a categorical distribution, for example, some flower is classified as *Iris Setosa* with a membership degree of 0.7, as *Iris Virginica* with a membership degree of 0.1, and as *Iris Versicolor* with a membership degree of 0.2.

Depending on the type of object representation different metrics of similarity are used. For objects defined in a metric space, similarity is defined as a quantity that is inversed to the distance between two points. Coordinates of each point are numerical values of the object's attributes. The smaller the distance between the analyzed objects the more similar they are. In paper [2] around 50 different metrics are analyzed, the most popular among them are particular cases of Minkowski's metric: Euclidean distance, City Block distance, and Chebyshev metric. The cosine similarity metric is often used as well. It calculates the cosine angle between two vectors that start at the origin and end at the analyzed objects.

In categorical space, the similarity between two objects is usually defined as similarity superposition of objects over each category. Most often it is the sum of similarities over each separated category. For this approach, each category is considered independently and in isolation from others. There is also an inversed approach when the object divergence is first found for each category and then aggregates to find the overall similarity. One of the popular variants of such a metric is proposed in [3] to find the similarity of fuzzy sets. The authors define the divergence of objects as a module of membership degree difference. All metrics from the survey paper [2] and other relevant publications, for example, [4, 5, 6, 7], do not consider the kinship between categories. But for some practical problems, the categories are akin. This leads to the fact that it is better calculate the similarity between objects not only directly as the similarity between equivalent categories, but also consider indirect cross-similarity through akin categories. Developing of such a metric that additionally considers the similarity of objects through akin categories is, therefore, the purpose of this paper.

2. Objects representation in the space of akin categories

We present here an example of problems in which objects are described in the space of akin categories.

Let us consider task of finding similar researchers, for example, for reviewer recommendation system. Based on research papers, each researcher can be categorized into a few research specialties (research fields) according to some research classification system. For example, researcher *A* is categorized to *System Analysis* with a membership degree of 0.4 and to *Information Systems and Information Technologies* with a membership degree of 0.6. Researcher *B* is categorized to *System Analysis* with a membership degree of 0.7 and to *Computer science* with a membership degree of 0.2. Researcher *C* is categorized to *System Analysis* with a membership degree of 0.4 and to *Marketing* with a membership degree of 0.6.

Finding the similarity between each pair of these researchers using known metrics will only take into account their membership degrees to *System Analysis*. Membership degrees in other categories are not taken into account because they are different for each researcher. The similarity between researchers *A* and *B* is defined only by their membership degrees to *System Analysis* category, which equals 0.4 and 0.7, respectively. If the similarity is defined as the common part of membership degrees using *min* operation, then between researchers *A* and *B*, it equals to $Fit(A,B)=\min(0.4,0.7)=0.4$. By the same reasoning, the similarity between researchers *A* and *C* equals to $Fit(A,C)=\min(0.4,0.4)=0.4$, and between researchers *B* and *C* equals to $Fit(B,C)=\min(0.7,0.4)=0.4$. This shows that the similarity of all pairs of researchers is the same. But the research domains are such that *Information Systems and Information Technologies* is significantly closer to *Computer Science* than to *Marketing*. Also, *Computer Science* is significantly closer to *System Analysis* than to *Marketing*. Therefore, the similarity between researchers *A* and *B* has to be stronger than the similarity between researchers *A* and *C*, or between researchers *B* and *C*. But the well-known similarity metrics do not take into account the kinship of categories; therefore, it is impossible to take into account such peculiarities using them.

3. The proposed metric for akin categories

Let us denote the number of categories as m. Then, objects X and Y, the similarity of which is to be estimated, we represent with the following membership distributions to categories: $(\mu_1(X), \mu_2(X), ..., \mu_m(X))$ and $(\mu_1(Y), \mu_2(Y), ..., \mu_m(Y))$. The distributions are considered to satisfy the following conditions:

$\mu_i(X) \in [0;1],\ i = \overline{1,m};$

$\mu_i(Y) \in [0;1],\ i = \overline{1,m};$

$\sum_{i=\overline{1,m}} \mu_i(X) = 1;$

$\sum_{i=\overline{1,m}} \mu_i(Y) = 1.$

The problem is that the level of similarity is to be calculated for objects X and Y. The domain research's peculiarity lies in the fact that some categories are akin. Therefore, it is necessary to consider not only the similarity of identical categories but also the similarity of akin categories. Below we propose a metric that accounts for the semantic kinship of categories.

The similarity between two objects X and Y is proposed to be calculated in the following way:

$$Fit(X,Y) = F(X,Y) + \Delta F(X,Y), \tag{1}$$

where $F(X,Y)$ denotes addend that assesses the direct similarity between the objects X and Y over identical categories;

$\Delta F(X,Y)$ denotes addend that considers the similarity of objects X and Y over akin categories.

We calculate the first addend of the formula (1) by the simplified version of Chekanowski metric for the case when membership degrees are in the $[0;1]$ and their sum equals 1. The resulting form of the first addend in (1) is as follows:

$$F(X,Y) = \sum_{i=\overline{1,m}} \min(\mu_i(X), \mu_i(Y)) \tag{2}$$

where $\mu_i(X)$ denotes membership degree of object X to the i-th category, $i = \overline{1,m}$;

$\mu_i(Y)$ denotes membership degree of object Y to the i-th category, $i = \overline{1,m}$.

The formula (2) can be interpreted as a sum of membership degrees of intersection of fuzzy sets X and Y. In formula (2), it is implied that the overall similarity of two objects is a sum of their similarities by each category. The similarity by category is defined as both objects' minimum memberships to the category. Thereby, one of the objects contributes the entire value of the membership degree to a category, and the other one – only a part of it.

After applying the formula (2) we get the following residuals of membership degrees:

$r_i(X) = \max(0, \mu_i(X) - \mu_i(Y));$

$r_i(Y) = \max(0, \mu_i(Y) - \mu_i(X)), i = \overline{1,m}$.

Let us consider the contribution of residuals to the similarity of two objects using kinship of categories. We assume that the information about categories' pair kinship is known, and denote it with the following binary relation:

$$\mathbf{K} = \|k_{ij}\|,$$

where $k_{ij} \in [0;1]$ denotes the kinship coefficient of the i-th and j-th categories, $i = \overline{1,m}$, $j = \overline{1,m}$.

The categories are more akin, the higher the kinship coefficient. Kinship relation is symmetric and reflexive, therefore, $k_{ij} = k_{ji}$ and $k_{ii} = 1$.

We denote the composition of residuals in the form of a matrix as follows:

$$\mathbf{E} = \left\| e_{ij} \right\|,$$

where $e_{ij} = \min\left(r_i(X), r_j(Y)\right)$, $i = \overline{1, m}$, $j = \overline{1, m}$.

The contribution of residuals to metric (1) using paired kinship of categories is calculated as follows:

$$\Delta F(X, Y) = \sum_{i=\overline{1, m}} \sum_{j=\overline{1, m}} e_{ij} \cdot k_{ij}. \quad (3)$$

Example. Two objects are defined with the following membership degrees to categories $\{A, B, C, D\}$: $X = (0.4 \quad 0.3 \quad 0.2 \quad 0.1)$ and $Y = (0.7 \quad 0.1 \quad 0.1 \quad 0.1)$. Kinship of the categories are described by the following matrix: $\mathbf{K} = \begin{Vmatrix} 1.0 & 0.5 & 0.0 & 0.0 \\ 0.5 & 1.0 & 0.1 & 0.0 \\ 0.0 & 0.1 & 1.0 & 0.3 \\ 0.0 & 0.0 & 0.3 & 1.0 \end{Vmatrix}$.

Let us calculate the similarity between objects X and Y using the proposed metric (1).

To calculate the first addend of the similarity metric (1) let us find the intersection of two distributions. The intersection is shown in Figure 1 with a hatch.

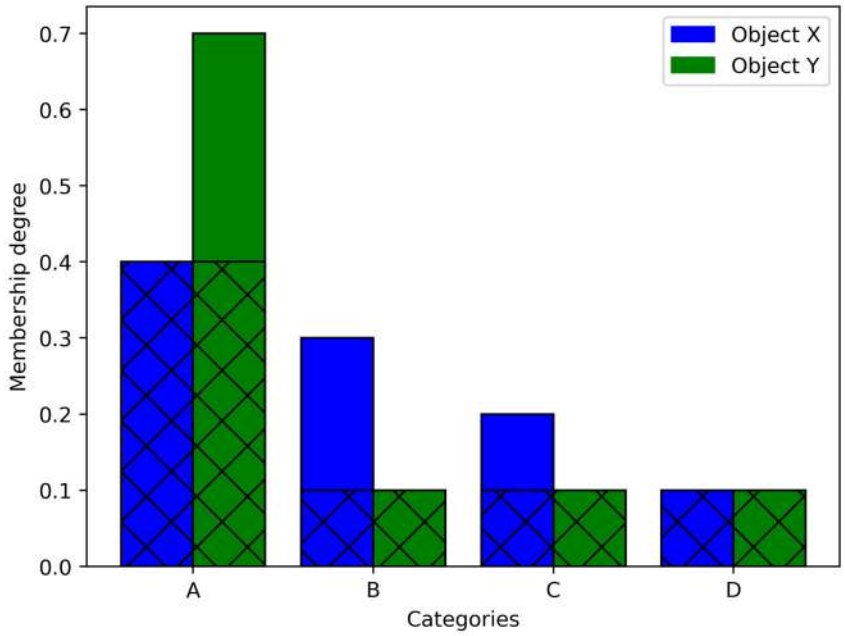


Figure 1: Intersection of two categorical distributions for calculating $F(X, Y)$

The numerical value of the first addend is as follows:

$$F(X, Y) = \min(0.4, 0.7) + \min(0.3, 0.1) + \min(0.2, 0.1) + \min(0.1, 0.1) = 0.4 + 0.1 + 0.1 + 0.1 = 0.7.$$

Residuals after the intersection are:

$e(X) = (0 \quad 0.2 \quad 0.1 \quad 0)$;

$e(Y) = (0.3 \quad 0 \quad 0 \quad 0)$.

The residuals composition is calculated as $\mathbf{E} = \begin{Vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.2 & 0.0 & 0.0 & 0.0 \\ 0.1 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{Vmatrix}$.

By having performed element-wise product of matrices $\mathbf{E}$ and $\mathbf{K}$, we get the following matrix of contributions through akin categories: $\begin{Vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.1 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{Vmatrix}$. From this matrix it is observed, that the contribution of kinship of the second and first category equals to 0.1. The kinship contribution of other categories is zero.

The summed contribution of akin categories equals to: $\Delta F(X,Y) = 0.1$. The resulting similarity value of objects X and Y using formula (1) equals $F(X,Y) = 0.7 + 0.1 = 0.8$.

4. Computational Experiments

In the above example, taking into account the kinship of categories increased the similarity metric by 0.1, which is 14% of the initial value obtained by the Chekanowski metric. Let us conduct computational experiments to establish how sensitive the proposed metric is to taking into account the kinship of categories.

We perform 18 series of experiments. All the experiments within a series is carried out with the same number of categories. The number of categories (m) from series to series increases from 2 to 70 with a step of 4. In each series, the similarity of 5000 pairs of objects X and Y is calculated. Attributes of objects X and Y as well as the kinship matrix of categories are randomly generated using the generalized Pareto law [8]:

$$f(x) = \frac{1}{\sigma}\left(1 + \frac{k(x-\theta)}{\sigma}\right)^{\left(-\frac{1}{k}-1\right)}.$$

The choice of this law is due to the fact that in topic modeling, the categories' similarity distribution often is similar to the Pareto distribution (see, for example, [9, 10]). For each experiment, we firstly randomly generate parameters of the distributions. The shape parameter k is generated from the range $[0.15; 0.5]$, the scale parameter σ is generated from the range [0.001; 0.01]; we set the bias as zero: $\theta = 0$. In each experiment, using synthesized distributions, we randomly generate the coordinates of the vectors X and Y, as well as kinship matrix $\mathbf{K}$. In matrix $\mathbf{K}$ the maximum value of the elements is limited to 0.4; vectors X and Y are normalized by 1. Then, we calculate the similarity between X and Y using the proposed metric (1).

The results of the experiments in the form of box-plot diagrams are shown in Figure 2. It can be seen from the figure that the similarity values are in the range $[0;1]$. As the number of categories increase, the spread of results decreases. Median of distributions for $m > 6$ does not depend on the number of categories.

On Figure 3 box-plot diagrams of the distribution of the term $\Delta F(X,Y)$ are shown, which takes into account the similarity of objects X and Y over akin categories. The median of this value increases from 0.001 to 0.066 with an increase in the number of categories. The spread of the distributions also increases, while the number of outliers decreases from 550 to 7. The increase in the median probably indicates that as the number of categories increases, a large number of weak ties between akin categories make a significant contribution to $\Delta F(X,Y)$. As the number of categories increase, the number of pairs of akin categories increases quadratically. The contribution of many pairs of akin categories is tiny. It looks as a noise. But the sum of huge number of noise contributions turns out to be large. The decrease of the number of outliers is also a negative factor. The need for a new metric is primarily due to the need to identify specific cases with strong cross-over effects due to akin categories. Also, the decrease in the number of outliers indicates that noise kinships make it difficult to detect such pairs of objects.

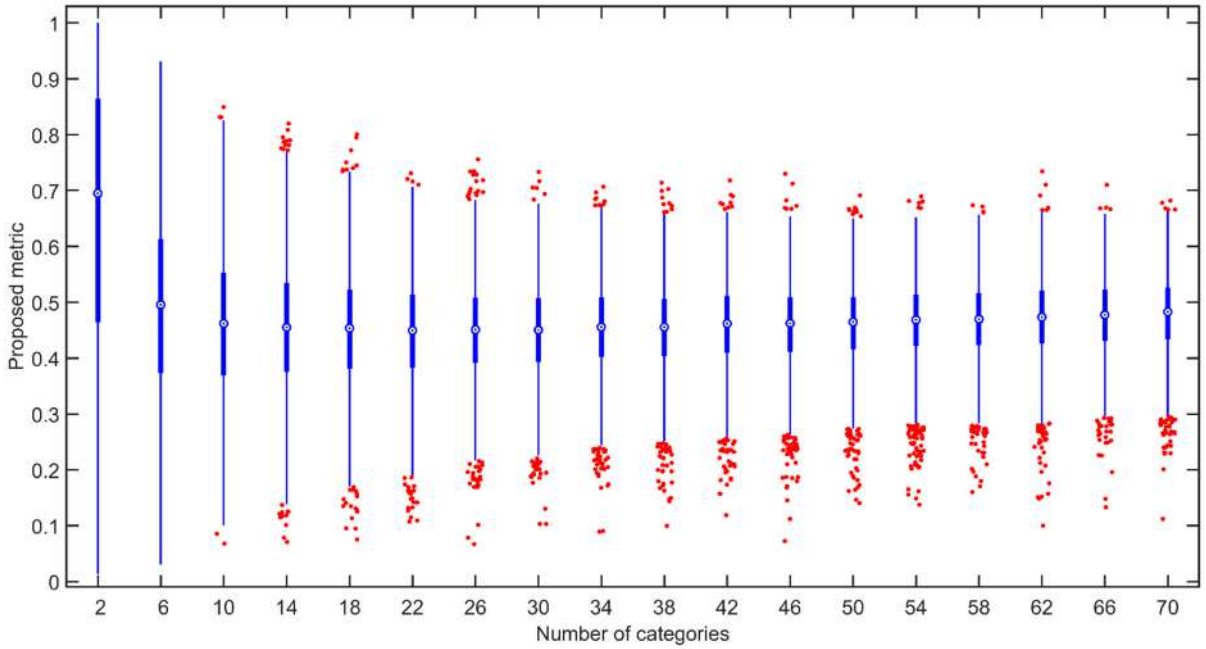


Figure 2: Similarity distributions according to the proposed metric

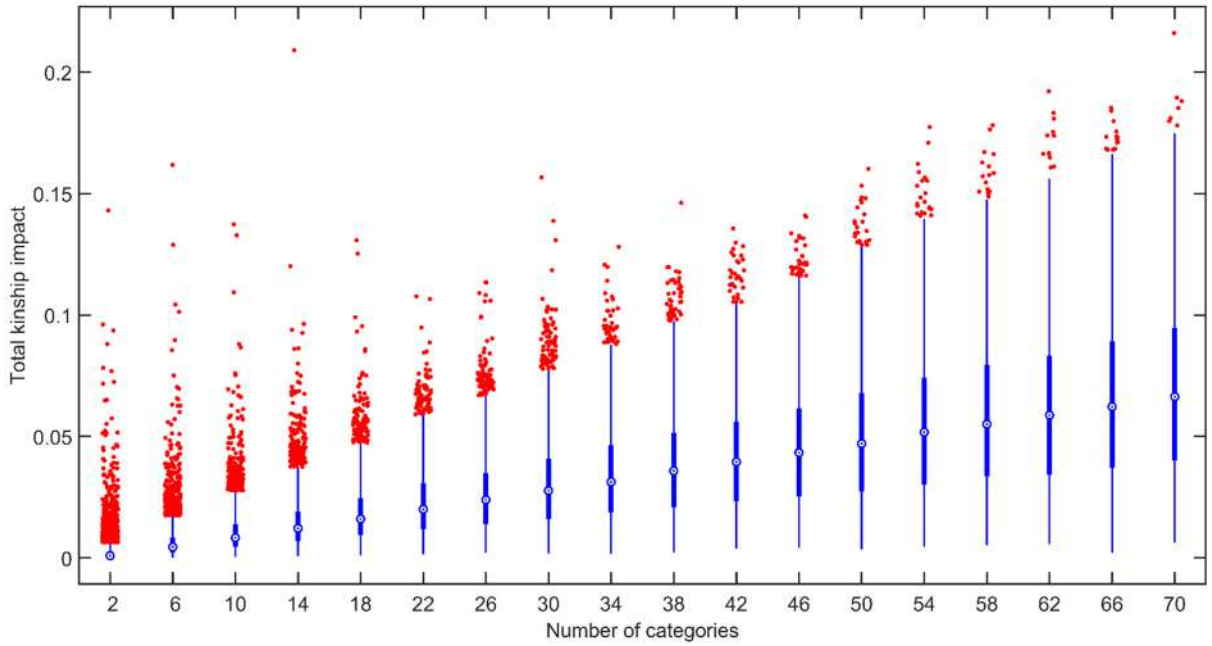


Figure 3: Distribution of the akin categories' contribution

Let us reduce the noise contribution of categories with weak kinship. We assume the kinship to be noisy if the kinship coefficient is less than 0.05. Box-plot diagrams of the noise kinship distribution's contribution are shown in Figure 4. It shows that the median noise contribution increases linearly and reaches a value of 0.056 for 70 categories. The maximum value of the contribution from noise kinship is fixed at the level of 0.122.

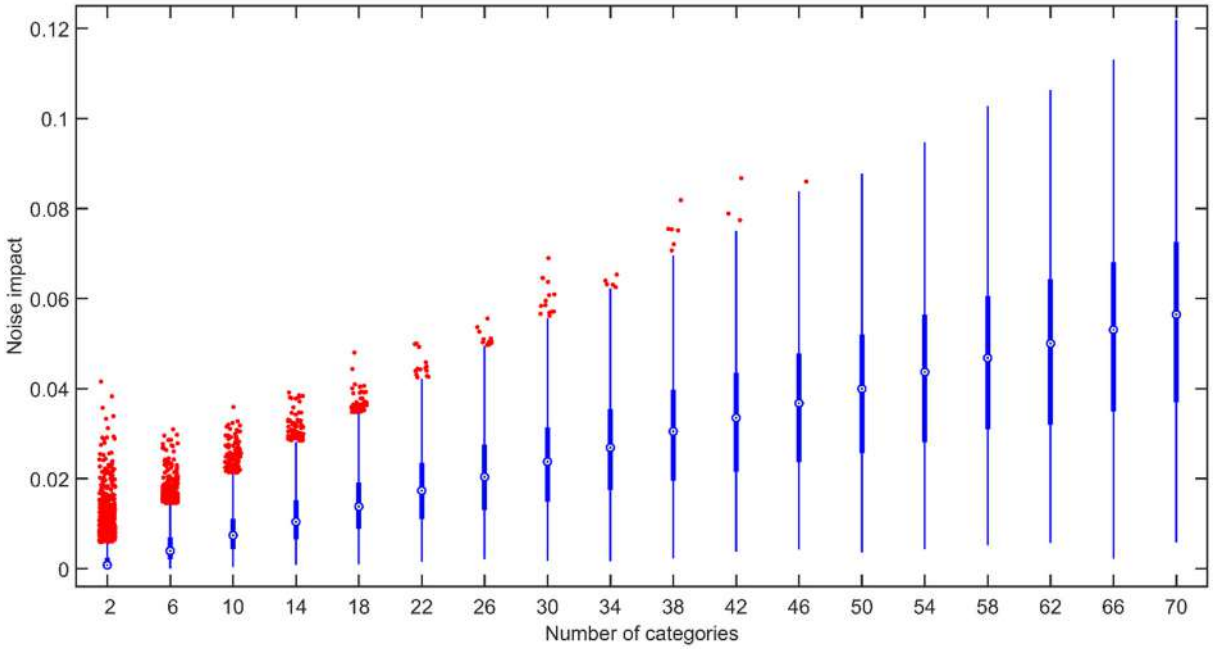


Figure 4: Distributions of noise contribution from consideration of akin categories

As for the relative noise contribution (shown on Figure 5), its median exceeds 10% in the series of experiments for a large number of categories. In each series of experiments, there are numerous cases when the noise contribution exceeds 15%. The noise contribution exceeds 25% in 194 cases out of 90,000 (Figure 6).

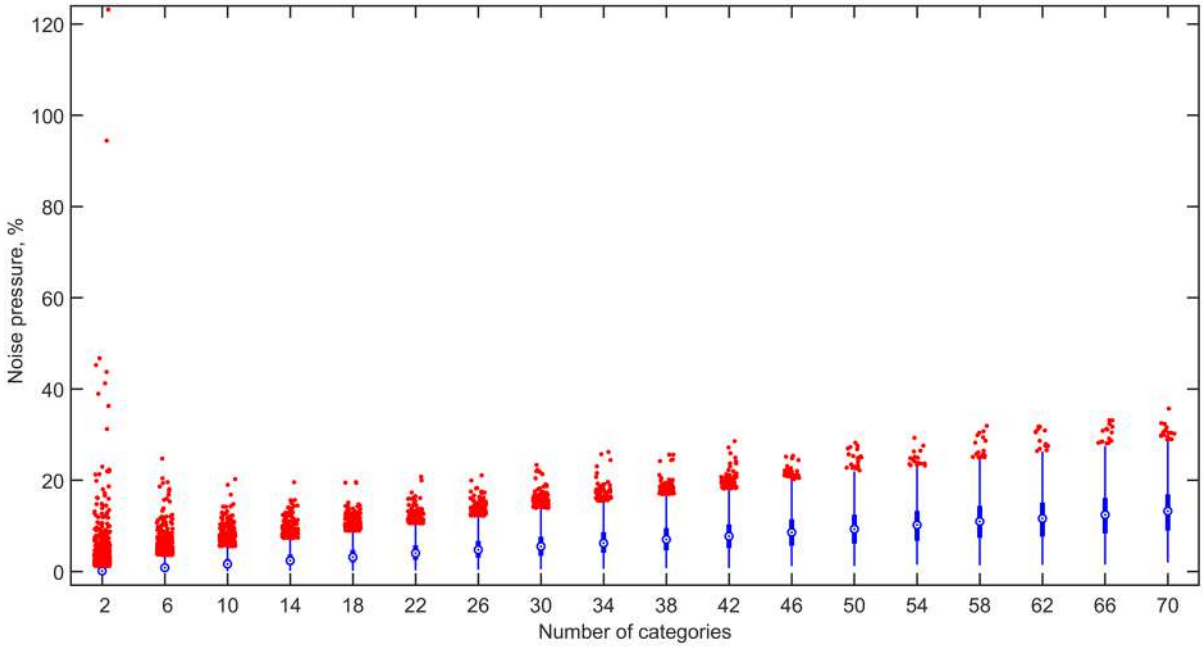


Figure 5: Distributions of the relative value of the noise contribution from akin categories

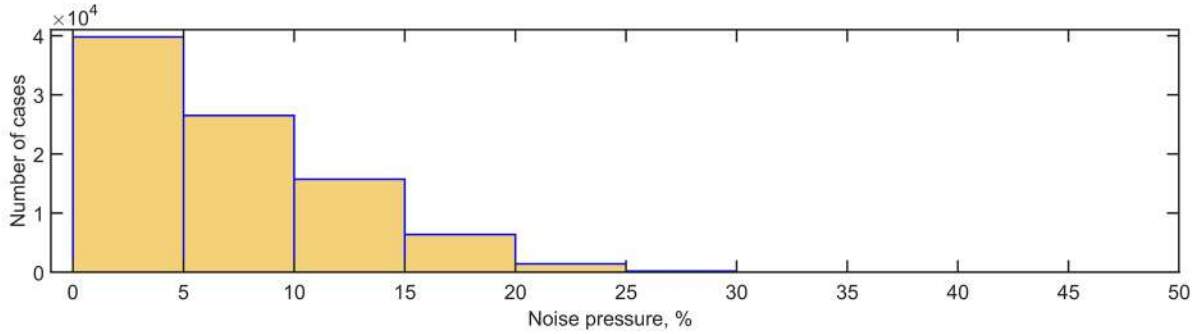


Figure 6: Distribution of noise contribution

After noised component elimination, the box-plot diagrams of the akin categories contribution's distribution are shown in Figure 7. The median of this value increases only from 0 to 0.005 as the number of categories increases. This is 13 times less than without noise elimination. At the same time, a large number of outliers are observed – their number is from 5-20%. This indicates that the metric is able to identify cases of strong interaction across akin categories. The box-plots of the proposed metric after removing the contribution from the noisy kinship of the categories are shown in Figure 8. Median of the refined metric lies in range [0.43; 0.69]. It is still high, especially for cases with large number of categories. Probably, it is caused by the noised memberships to many categories in source distributions in for X and Y. Hence, it is better to eliminate the noised memberships before assessing the similarity of two categorical distributions. It is reasonable to do so for such topical modeling problems when it is known that any object may belong just to a small number of categories.

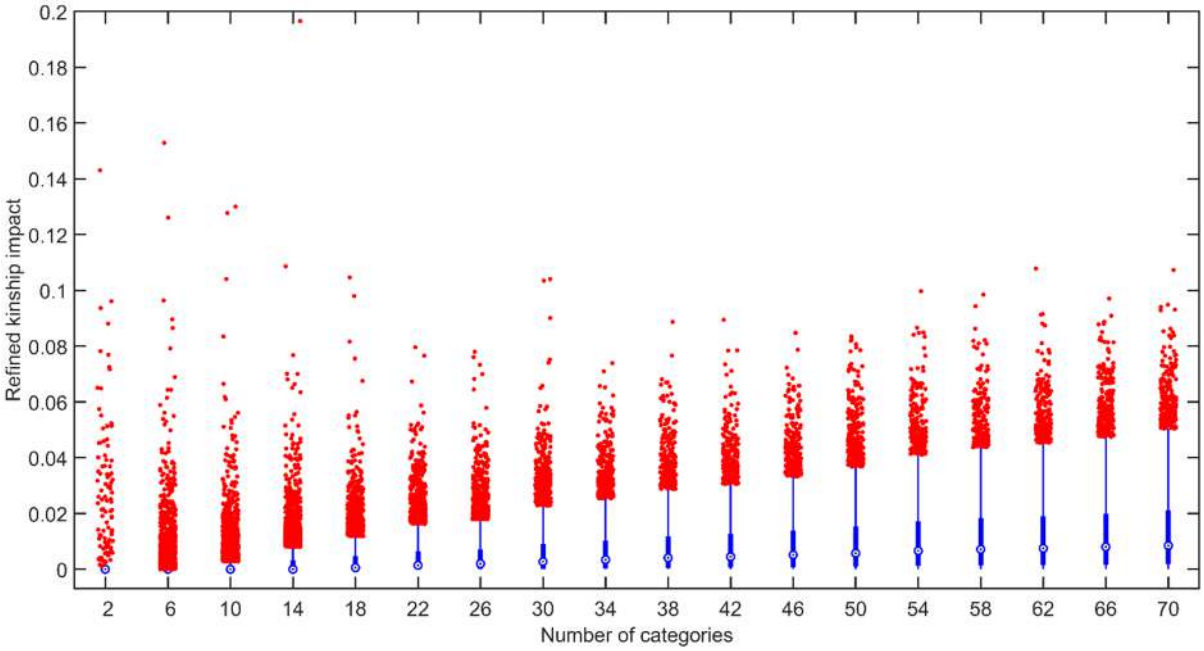


Figure 7: Distributions of the akin categories contribution after noise filtering

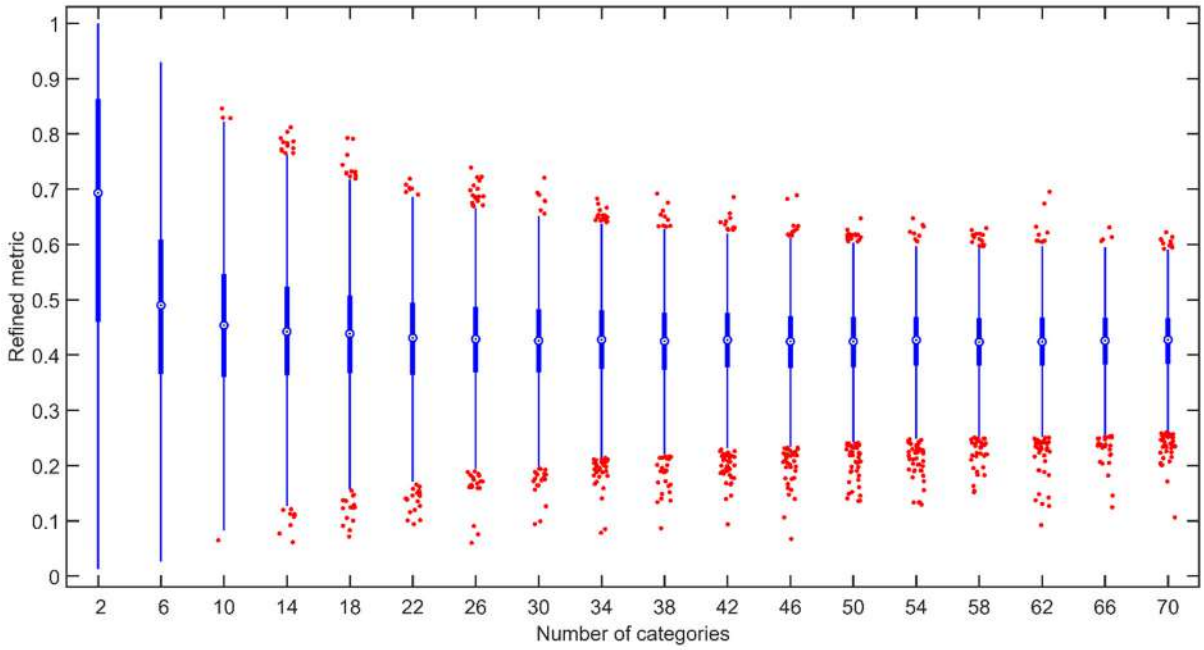


Figure 8: Similarity distributions according to the proposed metric with noise filtering

5. Conclusions

A new similarity metric of categorical distributions is proposed. A feature of the proposed metric consists of taking into account the kinship of categories. The proposed metric has two components. The first component is defined as Chekanowski metric. It calculates the direct similarity of distributions by categories as the sum of the intersection of distributions' membership degrees of two objects. The second component of the metric takes into account the similarity of objects through akin categories. It is assumed that the kinship coefficients of each pair of categories are known.

Computational experiments have shown that in the case of Pareto distributions of objects memberships to categories and Pareto distributions of kinship coefficients of categories, the proposed metric takes values from the interval [0; 1]. It was established that with an increase in the number of categories, the contribution to the proposed metric of the term, which takes into account the kinship of the categories, strongly increases. This is due to the fact that the number of weak ties between akin categories increases quadratically, each of which adds some contribution to the value of the metric. And although the contribution from many pairs of akin categories is tiny, roughly speaking - noisy, but the sum of the contributions turns out to be large.

To eliminate the noise impact, we suggest ignoring the noise kinship of the categories. A simple filter with kinship coefficient threshold at the level of 0.05 eliminated this drawback. After such noise filtering, the distributions of the second component of the metric, which takes into account the kinship of the categories, have a significant number of outliers. A significant number of outliers is observed in all the series of experiments, both with a small number of categories and with a large one. This fact indicates that the proposed metric makes it easy to identify pairs of objects whose similarity is largely determined by membership to akin categories.

The proposed metric can be used for topic modeling tasks, in which, when evaluating the similarity of two objects, it is necessary to take into account their membership to akin categories. Such tasks can be the selection of reviewers of research papers and theses, the analysis of the similarity of text documents, the identification of poses of people in a video stream, the clustering of species distribution, the formation of recommendations in online shops, etc.

6. References

[1] A. Abdelrazek, E. Yomna, G. Ema, M. Walaa, H. Ahmed, Topic modeling algorithms and applications: A survey, Information Systems 112 (2023) 102131. doi: 10.1016/j.is.2022.102131.

[2] N. Sebe, J. Yu, Q. Tian, J. Amores, A new study on distance metrics as similarity measurement, in: Proceedings of IEEE International Conference on Multimedia and Expo, Toronto, 2006, pp. 533–536. doi: 10.1109/ICME.2006.262443.

[3] W.-J. Wang, New similarity measures on fuzzy sets and on elements, Fuzzy Sets and Systems 85 (1997) 305-309. doi: 10.1016/0165-0114(95)00365-7.

[4] S.-H. Cha, Comprehensive survey on distance/similarity measures between probability density functions, International Journal of Mathematical Models and Methods in Applied Sciences 1 (2007) 300-307.

[5] J. Yu, Q. Tian, J. Amores, N. Sebe, Toward robust distance metric analysis for similarity estimation, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, 2006, pp. 316–322, doi: 10.1109/CVPR.2006.310.

[6] K. Louisa, A. Colubi, New metrics and tests for subject prevalence in documents based on topic modeling, International Journal of Approximate Reasoning 157 (2023) 49-69. doi: 10.1016/j.ijar.2023.02.009.

[7] D. J. Weller-Fahy, B. J. Borghetti, A. A. Sodemann, A survey of distance and similarity measures used within network intrusion anomaly detection, IEEE Communications Surveys & Tutorials, 17 (2015) 70-91. doi: 10.1109/COMST.2014.2336610.

[8] H. T. Davis, M. L. Feldstein, The generalized Pareto law as a model for progressively censored survival data, Biometrika, 66 (1979) 299-306. doi: 10.1093/biomet/66.2.299.

[9] S. Shtovba, M. Petrychko, An algorithm for topic modeling of researchers taking into account their interests in Google Scholar profiles, in: Proceedings of the Fourth International Workshop on Computer Modeling and Intelligent Systems, Zaporizhzhia (2021), CEUR Workshop Proceedings, 2864 (2021) 299-311. URL: https://ceur-ws.org/Vol-2864/paper26.pdf.

[10] S. Shtovba, M. Petrychko, Jaccard index-based assessing the similarity of research fields in Dimensions, in: Proceedings of the First International Workshop on Digital Content & Smart Multimedia, Lviv (2019), CEUR Workshop Proceedings, 2533 (2019) 117-128. URL: https://ceur-ws.org/Vol-2533/paper11.pdf.

Neural Network Method for Detecting and Diagnostics Helicopters Turboshaft Engines Surge at Flight Modes

Serhii Vladov[1], Yurii Shmelov[1], Ruslan Yakovliev[1] and Maryna Petchenko[2]

[1] *Kremenchuk Flight College of Kharkiv National University of Internal Affairs, vul. Peremohy, 17/6, Kremenchuk, Poltavska Oblast, Ukraine, 39605*
[2] *Kharkiv National University of Internal Affairs, L. Landau Avenue, 27, Kharkiv, Ukraine, 61080*

Abstract

The work is devoted to the development of a neural network method for diagnostics (monitoring) helicopters turboshaft engines pre-surge status in real time (at helicopter flight mode). The developed method for helicopters turboshaft engines is based on a mathematical model of the time distribution of air pressure in the compressor during the transient process, which is construct on the basis of the classical theory of the movement of liquids and gases, taking into account the features of the thermogas-dynamic flow in the compressor of helicopters turboshaft engines. Diagnostics (monitoring) helicopters turboshaft engines in real time is carried out using a linear neural network with the optimal number of neurons – 10 or more. It is proposed to train a linear neural network on dynamic neurons, which, due to the adjustable smoothing parameter from the range from zero to one, made it possible to obtain an accuracy of 99.99 % of the task being solved. The developed method can serve as one of the blocks of the onboard neural network expert system for the integrated monitoring and operation of helicopters turboshaft engines, which automatically decides on helicopters turboshaft engines operational status at helicopter flight mode and provides the crew with information about the possibility of helicopter further movement.

Keywords

Helicopters turboshaft engines, compressor, neural network, training, thermogas-dynamic parameters, diagnostics (monitoring), pre-surge status, signal, pressure, error

1. Introduction

At present, a modern aircraft gas turbine engine (GTE) and its control systems are a complex dynamic system. The correctness and safety of the operation of aviation gas turbine engines require constant and continuous monitoring of its parameters, the effectiveness of which significantly depends on the probability of correct recognition of its technical condition, including defects, which directly affects the quality of gas turbine operation control systems, which ultimately determines the efficiency and safety of flights.

One of aircraft GTE leading defects is the stall mode of its operation – surge, which is characterized by various non-stationary phenomena resulting from the loss of stability of the air flow in the compressor. In this case, strong pulsations of the air flow appear, a drop in its pressure, which leads to vibrations of the compressor blades and can cause its destruction. Thus, surge is not allowed during engine operation [1], which indicates the need for its monitoring (diagnostics).

The development of approaches to diagnostics aircraft GTE operational status, including helicopters turboshaft engines (TE), is proceeding in several directions. Much attention is paid to the improvement of algorithmic support, which expands the capabilities of diagnostic models and increases the reliability

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: ser26101968@gmail.com (S. Vladov); nviddil.klk@gmail.com (Yu. Shmelov); ateu.nv.klk@gmail.com (R. Yakovliev); marinapetcenko@gmail.com (M. Petchenko)
ORCID: 0000-0001-8009-5254 (S. Vladov); 0000-0002-3942-2003 (Yu. Shmelov); 0000-0002-3788-2583 (R. Yakovliev); 0000-0003-1104-5717 (M. Petchenko)

of diagnostics. In [2, 3], the advantages of using artificial intelligence methods over classical diagnostic methods for troubleshooting are shown. It is noted that neural networks are the most effective, since they have high adaptive characteristics and can solve complex problems of classification and pattern recognition. Existing neural network diagnostic methods are limited by the specificity of the tasks being solved, the insufficient development of the theory of their application for TE diagnostics, the lack of universal and formalized approaches, and the imperfection of the methods themselves. In this regard, an urgent scientific and practical task is the development of a neural network method for monitoring (diagnosing) surge (pre-surging state) of TE in helicopter flight conditions, which will significantly improve the safety of helicopter flights.

2. Related Works

There are known surge diagnostic methods [4, 5], in which the measured parameters are: gas temperature in front of the compressor turbine, pressure at the inlet and outlet of the compressor, gas generator rotor r.p.m. As is known [6], when a surge occurs, gas temperature in front of the compressor turbine increases, gas generator rotor r.p.m. decreases, and the air pressure behind the compressor sharply decreases relative to the pressure at the inlet to air inlet section. The conclusion about the development of surge in the compressor is made in case of exceeding a predetermined threshold value of the ratio of gas temperature in front of the compressor turbine to gas generator rotor r.p.m.

The disadvantage of the known methods is that the ratio of gas temperature in front of the compressor turbine to gas generator rotor r.p.m. can exceed a predetermined threshold value when the engine operation mode changes, for example, when throttling, on the basis of which a false conclusion can be made about the presence of surge.

An analysis of published works devoted to the use of neural networks for diagnostics the aircraft GTE operational status shows that in [7, 8] the main trends are outlined and the characteristic features of solving the problems of diagnostics aircraft GTE based on neural networks are highlighted.

At the same time, they are mainly devoted to solving particular problems (for example, GTE turbine blades operational status diagnostics [9], forming a space of diagnostics signs of aircraft GTE operational status to build a neural network classifier [10], indirectly measuring the temperature of gases behind the combustion chamber based on neural networks for diagnostics the thermal state of the engine [11]. However, they do not contain instructions on the choice of architecture, structure and training algorithms for the neural network, there is no engineering methodology for designing neural networks in relation to the problems of aircraft GTE operational status diagnostics) of helicopters TE are also missing.

3. Methods and Materials

Let $\rho(x, t)$ be the density of the liquid, then applying the law of conservation of mass, we obtain that the rate of change of mass in the volume V must be equal to the mass flow crossing the surface S of the volume V (fig. 1, a). The mass flow through the surface element dS is equal to $-\rho v dS$ [12, 13]:

$$\frac{\partial}{\partial t}\iiint_V \rho d\tau = -\iint_S \rho V dS. \quad (1)$$

Applying the Gauss's-Ostrogradsky's theorem, we arrive at the differential equations for conservation of mass [12, 13]:

$$\frac{\partial \rho}{\partial t} + \nabla \rho V = 0. \quad (2)$$

Similarly, the equation of motion for the medium can be derived from the condition of conservation of pulse. Consider the preservation of the projection of the pulse in the X-direction (fig. 1, b). The X-component of the total pulse in the volume V is $\iiint_V \rho v_x d\tau$. Due to pulse convection and the influence of pressure p in the X-direction, the X-component of the pulse of the medium in the volume V increases with time (e_x – unit vector in the X-direction), i.e., $-\iint_S \left(\rho v_x V + p e_x\right) dS$. Then, from the law of conservation of pulse (X-components), we get the equation:

$$\frac{\partial}{\partial t}\iiint_V \rho v_x d\tau = -\iint_S \left(\rho v_x V + p e_x\right) dS. \quad (3)$$

Using the Gauss's-Ostrogradsky's theorem, we obtain:

$$\frac{\partial}{\partial t}\iiint_V \rho v_x d\tau = \iiint_V \nabla\left(\rho v_x V + p e_x\right) d\tau; \quad (4)$$

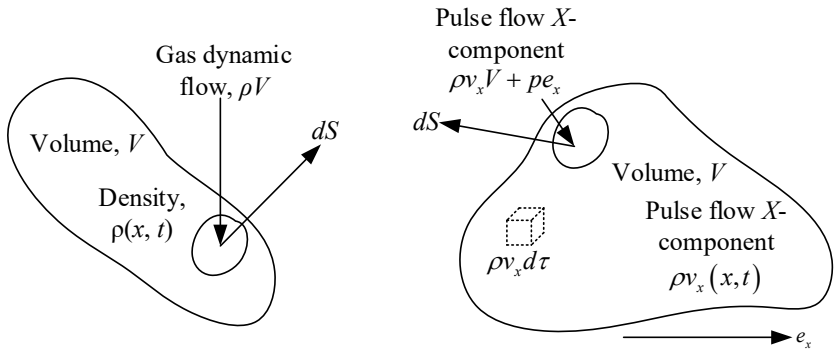
$$\frac{\partial \rho v_x}{\partial t} + \nabla\left(\rho v_x V + p e_x\right) = 0. \quad (5)$$

Similarly, the equations of motion for the *Y*- and *Z*-directions are obtained. Let's combine these equations and get:

$$\frac{\partial \rho v_x}{\partial t} + \nabla\left(\rho V + pI\right) = 0; \quad (6)$$

where I – unit tensor.

Let's draw two cross-sections at any place of the flow in the gas pipeline, let the distance between them be *dx*.



a b

Figure 1: Diagram of conservation in the environment: a – mass; b – pulse [12]

Using the pulse theorem, we obtain (fig. 1, *b*):

$$\frac{\partial}{\partial t}\left(\int_f \rho v df dx\right) + \frac{\partial}{\partial x}\left(\int_f \rho v^2 df\right) = -\frac{\partial}{\partial x} fp dx - \tau\chi dx - \gamma f dx \sin\alpha + p\frac{\partial f}{\partial x} = -f\frac{\partial p}{\partial x}dx - \tau\chi dx - \gamma f dx \sin\alpha; \quad (7)$$

where ρ – density; p – average pressure in the section; f – cross-sectional area; v – longitudinal velocity in the cross-sectional element; t – time; τ – projection of the tangential stress on the pipe wall onto the X axis – flow direction – average over the wetted perimeter; χ – wetted perimeter; γ – gas volume unit weight; α – elevation angle of the dx element axis above the horizon.

We reduce this equation to *dx*, we get:

$$\frac{\partial M}{\partial t} + \frac{\partial f}{\partial x} = -f\frac{\partial p}{\partial x} - \tau\chi - \gamma f dx; \quad (8)$$

where M – flow rate; I – projection on the X-axis of the amount of movement of the mass M:

$$M = \int_f \rho v df; \quad (9)$$

$$I = \int_f \rho v^2 df. \quad (10)$$

Equation (8) is a general equation that is valid for any gas-dynamic flow in a pipe.

Consider the mass balance entering and leaving the *dx* element, using the continuity theorem, we obtain the equation:

$$\frac{\partial\left(fp\right)}{\partial t} + \frac{\partial M}{\partial x} = 0. \quad (11)$$

In the general case, the quantity I can be given in the form:

$$I = \int_f \rho v^2 df = (1+\beta) f \rho \omega^2 = (1+\beta) M \omega; \tag{12}$$

where ω – average velocity in the section, β – Coriolis correction for the uneven distribution of velocities in the expression of the amount of flow movement due to the average velocity and the average density in the section. As is known, in steady motion for a normal distribution of velocities in a turbulent flow $\beta \approx 0$, in a parabolic distribution $\beta = \frac{1}{3}$. In case of unsteady motion, β will naturally be a variable value that depends on the nature of the distribution of velocities in the pipe sections.

Next, we will use the following formula from gas dynamics for the tangential stress τ:

$$\tau = \frac{\lambda}{8} \rho \omega^2; \tag{13}$$

where λ – resistance coefficient in the Darcy-Weisbach formula for frictional head loss in the pipe. Its λ can always be set knowing the roughness of the pipe and the flow rate.

It is known that λ depends on the roughness of the pipe and the mode of movement (Reynolds number). We will accept the following assumption that the resistance characteristics established for stationary movements are preserved for non-stationary ones as well.

Rigorous substantiation of this assumption is quite difficult, although it is confirmed by a rather satisfactory agreement between theory and experience. Using the above remarks, equation (8) will be rewritten in the form:

$$\frac{\partial M}{\partial t} = -S \frac{\partial p}{\partial x} - \frac{\lambda}{8} \rho \omega^2 \chi S \sin\alpha - \frac{\partial}{\partial x}\left((1+\beta) M \omega\right). \tag{14}$$

Since $M = \rho S \omega$, the equation can be simplified:

$$\frac{\partial M}{\partial t} = -S \frac{\partial p}{\partial x} - M \frac{\lambda \omega}{8S} \chi - \gamma S \sin\alpha - \frac{\partial}{\partial x}\left((1+\beta) M \omega\right) = -S \frac{\partial p}{\partial x} - M \frac{\lambda \omega}{8\delta} - \gamma S \sin\alpha - \frac{\partial}{\partial x}\left((1+\beta) M \omega\right); \tag{15}$$

where δ – hydraulic radius of the section, which is equal to:

$$\delta = \frac{f}{X} = \frac{\pi \frac{d^2}{4}}{\pi d} = \frac{d}{4}; \tag{16}$$

where d is the diameter of the gas pipeline.

Next, let's return to the inseparability equation (11). For gas compression, let's take S = const and use the following formula:

$$c^2 = \frac{\partial p}{\partial \rho}; \tag{17}$$

where c – gas sound speed, from which we obtain, opening the full differentials dp and $d\rho$:

$$\frac{\partial \rho}{\partial t} \partial t + \frac{\partial \rho}{\partial x} \partial x = \frac{1}{c^2} \left(\frac{\partial p}{\partial t} \partial t + \frac{\partial \rho}{\partial x} \partial x \right). \tag{18}$$

Since the increments of dt and dx are arbitrary, it is necessary that $\frac{\partial \rho}{\partial t} = \frac{1}{c^2} \frac{\partial p}{\partial t}$. From here we get the following equation:

$$\frac{\partial (\rho S)}{\partial t} = S \frac{\partial \rho}{\partial t} = \frac{S}{c^2} \frac{\partial p}{\partial t}. \tag{19}$$

As a result, the equation of motion (8) and continuity (11) can be written in the form of the following system:

$$\begin{cases} -S \frac{\partial p}{\partial x} = \frac{\partial M}{\partial t} + M \frac{\lambda \omega}{8\delta} + \gamma S \sin\alpha + \frac{\partial}{\partial x}\left((1+\beta) M \omega\right); \\ -S \frac{\partial p}{\partial t} = c^2 \frac{\partial M}{\partial x}. \end{cases} \tag{20}$$

The system of equations (20) is a system of two differential equations of the first order in partial derivatives of the hyperbolic type, in the general case nonlinear. Dividing both parts of the system by S, we get:

$$\begin{cases} -\dfrac{\partial p}{\partial x} = \dfrac{\partial(\rho\omega)}{\partial t} + \dfrac{\lambda}{8\delta} + \gamma\sin\alpha + \dfrac{\partial}{\partial x}\left((1+\beta)M\omega\right); \\ \dfrac{\partial p}{\partial t} = \dfrac{1}{c^2}\dfrac{\partial p}{\partial x} - \dfrac{\partial(\rho\omega)}{\partial t}. \end{cases} \tag{21}$$

Let's simplify the system of equations (20). We will show that in equations (8) and (21) it is possible to neglect the term $\dfrac{\partial I}{\partial x} = \dfrac{\partial}{\partial x}\left((1+\beta)M\omega\right)$.

Let us denote $\sin\alpha = \dfrac{dz}{dx}$, where z – excess of the center of gravity of the pipe section above an arbitrary horizontal plane. It is possible to integrate the first of equations (21) over x from x_1 to x_2 at a fixed t and present the result in the following form:

$$\left(p_1 + \gamma z_1\right) - \left(p_2 + \gamma z_2\right) = \int_{x_1}^{x_2} \frac{\partial(\rho\omega)}{\partial t}\partial x + \rho\int_{x_1}^{x_2} \frac{\lambda\omega^2}{8\delta}\partial x + \left((1+\beta)\rho\omega^2\right)_2 - \left((1+\beta)\rho\omega^2\right)_1. \tag{22}$$

During the movement of gas with subsonic speed, it is always possible to neglect the dynamic pressure corresponding to the high-speed pressure, as well as to neglect the hydrostatic pressure of the gas due to its low density. It follows that in (21) the last difference $\left((1+\beta)\rho\omega^2\right)_2 - \left((1+\beta)\rho\omega^2\right)_1$ can be omitted, which means $\dfrac{\partial I}{\partial x} = \dfrac{\partial}{\partial x}\left((1+\beta)M\omega\right)$ that the term can be discarded in the first equation of system (20). In the future, under pressure p we mean the sum $p + \gamma z$ and omit the term $\gamma\sin\alpha$. As a result, we will get the following system:

$$\begin{cases} -\dfrac{\partial p}{\partial x} = \dfrac{\partial(\rho\omega)}{\partial t} + \dfrac{\lambda\rho\omega^2}{8\delta}; \\ -\dfrac{\partial p}{\partial t} = c^2\dfrac{\partial(\rho\omega)}{\partial x}. \end{cases} \tag{23}$$

Let's transform system (23) into the form:

$$\begin{cases} -\dfrac{\partial p}{\partial x} = \dfrac{\partial(\rho\omega)}{\partial t} + \dfrac{\lambda\omega}{8\delta}(\rho\omega); \\ -\dfrac{\partial(\rho\omega)}{\partial x} = \dfrac{1}{c^2}\dfrac{\partial p}{\partial t}. \end{cases} \tag{24}$$

Equations known in electrical engineering that describe the change in electric potential along an electric circuit ($d\varphi/dx$) and over time ($d\varphi/dt$) have a similar form, if this electric circuit is composed of elements that have, per unit length, an ohmic resistance of R_0, capacity C_0 and inductance L_0. These equations are known as telegraph equations of a long line and have the form [14, 15]:

$$\begin{cases} -\dfrac{\partial u}{\partial x} = L\dfrac{\partial i}{\partial t} + Ri; \\ -\dfrac{\partial i}{\partial x} = C\dfrac{\partial u}{\partial t}; \end{cases} \tag{25}$$

where $p \to u$, $\rho\omega \to i$, $R = \dfrac{\lambda\omega}{8\delta}$, $L = 1$, $C = \dfrac{1}{c^2}$.

In order to detect surge in the helicopter aircraft TE compressor, an elementary section of its model is considered in the form of a long line, which is a sequential *RCL*-oscillating circuit with a voltage source U_0 acting on it. The analysis of free oscillations in a sequential oscillating circuit leads to the solution of a system of two linear differential equations of the first order with constant coefficients of the relative variable state – current in the inductance (blood flow) and the voltage (pressure) on the capacity of the circuit. One of the equations is formed as a result of applying Kirchhoff's second law to the contour:

$$U_L + U_R + U_C - U_0 = 0. \tag{26}$$

The second equation relates the current in the circuit to the voltage on one of the elements:

$$i_L = C\frac{dU_C}{dt}. \tag{27}$$

It is also worth adding two more auxiliary equations describing the voltages on the inductance and resistance:

$$U_R = i_L R; \tag{28}$$

$$U_L = L\frac{di_L}{dt}. \tag{29}$$

Taking into account the above, the system of differential equations according to Kirchhoff's laws has the form:

$$\begin{cases} i_L = C\dfrac{dU_C}{dt}; \\ L\dfrac{di_L}{dt} + i_L R + U_C = U_0. \end{cases} \tag{30}$$

Having expressed from system (30) the derivatives of the voltage on the capacitor and the current in the inductance, we obtain the equation of state:

$$\begin{cases} \dfrac{dU_C}{dt} = i_L \dfrac{1}{C}; \\ \dfrac{di_L}{dt} = -U_C \dfrac{1}{L} - i_L \dfrac{R}{L} + U_0 \dfrac{1}{L} = 0. \end{cases} \tag{31}$$

The unknown state variables in the obtained system of equations (31) are the voltage on the capacitors and the current in the inductance. The matrix of system coefficients (31) in this case has the form $A = \begin{Vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{Vmatrix}$, where $a_{11} = 0$; $a_{12} = \dfrac{1}{C}$, $a_{21} = -\dfrac{1}{L}$, $a_{22} = -\dfrac{R}{L}$. Therefore

$$A = \begin{Vmatrix} 0 & \dfrac{1}{C} \\ -\dfrac{1}{L} & -\dfrac{R}{L} \end{Vmatrix}. \tag{32}$$

The matrix of free members is determined by the parameters of the active sources in the circuit and has the form:

$$BF = \begin{Vmatrix} 0 \\ U_0 \cdot \dfrac{1}{L} \end{Vmatrix}. \tag{33}$$

The matrix of initial conditions has the form:

$$X(0) = \begin{Vmatrix} U_C(0) \\ i_L(0) \end{Vmatrix} = \begin{Vmatrix} U_0 \\ 0 \end{Vmatrix}. \tag{34}$$

We determine the eigenvalues of matrix A:

$$\begin{vmatrix} -\lambda & \dfrac{1}{C} \\ -\dfrac{1}{L} & -\dfrac{R}{L} - \lambda \end{vmatrix} = 0; \tag{35}$$

$$\lambda^2 + \frac{R}{L}\lambda + \frac{1}{LC} = 0 \Rightarrow \lambda_{1,2} = \frac{-\frac{R}{L} \pm \sqrt{\left(\frac{R}{L}\right)^2 - 4\frac{1}{LC}}}{2} = -\frac{R}{2L} \pm \frac{1}{2}\sqrt{\left(\frac{R}{L}\right)^2 - 4\frac{1}{LC}};$$

$\lambda_{12} = a \pm jb$, because when substituting the model *RCL*-parameters $\left(\dfrac{R}{L}\right)^2 < \dfrac{4}{LC}$, where $a = -\dfrac{R}{2L}$; $b = \dfrac{1}{2}\sqrt{\left|\left(\dfrac{R}{L}\right)^2 - 4 \cdot \dfrac{1}{LC}\right|}$.

$$\lambda_1 - \lambda_2 = (a + jb) - (a - jb) = 2jb.$$

Sylvester's formula is used to calculate e^{At}, which has the form:

$$e^{At}=\sum_{k=1}^{n}\frac{\prod_{i=1,i\neq k}^{n}\left(A-\lambda_i\cdot 1\right)}{\prod_{i=1,i\neq k}^{n}\left(\lambda_k-\lambda_i\right)}e^{\lambda_k t};\tag{36}$$

where $1=\begin{Vmatrix}1 & 0\\ 0 & 1\end{Vmatrix}$ – diagonal identity matrix of order n for two states i_L and U_C. Then Sylvester's formula takes the form:

$$e^{At}=\frac{A-\lambda_2\cdot 1}{\lambda_1-\lambda_2}e^{\lambda_1 t}-\frac{A-\lambda_1\cdot 1}{\lambda_1-\lambda_2}e^{\lambda_2 t};\tag{37}$$

$$\lambda_2\cdot 1=(a-jb)\cdot\begin{Vmatrix}1 & 0\\ 0 & 1\end{Vmatrix}=\begin{Vmatrix}a-jb & 0\\ 0 & a-jb\end{Vmatrix};$$

$$A-\lambda_2\cdot 1=\begin{Vmatrix}0 & \frac{1}{C}\\ -\frac{1}{L} & -\frac{R}{L}\end{Vmatrix}-\begin{Vmatrix}a-jb & 0\\ 0 & a-jb\end{Vmatrix}=\begin{Vmatrix}-a+jb & \frac{1}{C}\\ -\frac{1}{L} & -\frac{R}{L}-a+jb\end{Vmatrix};$$

$$\lambda_1\cdot 1=(a+jb)\begin{Vmatrix}1 & 0\\ 0 & 1\end{Vmatrix}=\begin{Vmatrix}a+jb & 0\\ 0 & a+jb\end{Vmatrix};$$

$$A-\lambda_1\cdot 1=\begin{Vmatrix}0 & \frac{1}{C}\\ -\frac{1}{L} & -\frac{R}{L}\end{Vmatrix}-\begin{Vmatrix}a+jb & 0\\ 0 & a+jb\end{Vmatrix}=\begin{Vmatrix}-a-jb & \frac{1}{C}\\ -\frac{1}{L} & -\frac{R}{L}-a-jb\end{Vmatrix};$$

$$\frac{A-\lambda_2\cdot 1}{\lambda_1-\lambda_2}=\frac{1}{2jb}\cdot\begin{Vmatrix}-a+jb & \frac{1}{C}\\ -\frac{1}{L} & -\frac{R}{L}-a+jb\end{Vmatrix}=\begin{Vmatrix}-\frac{a}{2jb}+\frac{1}{2} & \frac{1}{2jbC}\\ -\frac{1}{2jbL} & -\frac{R}{2jbL}-\frac{a}{2jb}+\frac{1}{2}\end{Vmatrix};$$

$$\frac{A-\lambda_1\cdot 1}{\lambda_1-\lambda_2}=\frac{1}{2jb}\cdot\begin{Vmatrix}-a-jb & \frac{1}{C}\\ -\frac{1}{L} & -\frac{R}{L}-a-jb\end{Vmatrix}=\begin{Vmatrix}-\frac{a}{2jb}-\frac{1}{2} & \frac{1}{2jbC}\\ -\frac{1}{2jbL} & -\frac{R}{2jbL}-\frac{a}{2jb}-\frac{1}{2}\end{Vmatrix};$$

$$e^{At}=e^{(a+jb)t}\cdot\begin{Vmatrix}-\frac{a}{2jb}+\frac{1}{2} & \frac{1}{2jbC}\\ -\frac{1}{2jbL} & -\frac{R}{2jbL}-\frac{a}{2jb}+\frac{1}{2}\end{Vmatrix}-e^{(a-jb)\cdot}\cdot\begin{Vmatrix}-\frac{a}{2jb}-\frac{1}{2} & \frac{1}{2jbC}\\ -\frac{1}{2jbL} & -\frac{R}{2j\cdot bL}-\frac{a}{2jb}-\frac{1}{2}\end{Vmatrix}.\tag{38}$$

To calculate the voltage on the capacitor and the current in the inductance, the state equations are solved in matrix form:

$$X(t)=e^{At}\cdot X(0)+\int_0^t e^{A(t-\tau)}\cdot BFd\tau=e^{(a+jb)t}\cdot\begin{Vmatrix}-\frac{a}{2jb}+\frac{1}{2} & \frac{1}{2jbC}\\ -\frac{1}{2jbL} & -\frac{R}{2jbL}-\frac{a}{2jb}+\frac{1}{2}\end{Vmatrix}\cdot\begin{Vmatrix}U_0\\ 0\end{Vmatrix}-e^{(a-jb)t}\times$$

$$\times\begin{Vmatrix}-\frac{a}{2jb}-\frac{1}{2} & \frac{1}{2jbC}\\ -\frac{1}{2jbL} & -\frac{R}{2jbL}-\frac{a}{2jb}-\frac{1}{2}\end{Vmatrix}\cdot\begin{Vmatrix}U_0\\ 0\end{Vmatrix}+\int_0^t\left(e^{(a+jb)(t-\tau)}\cdot\begin{Vmatrix}-\frac{a}{2jb}+\frac{1}{2} & \frac{1}{2jbC}\\ -\frac{1}{2jbL} & -\frac{R}{2jbL}-\frac{a}{2jb}+\frac{1}{2}\end{Vmatrix}\cdot\begin{Vmatrix}0\\ U_0\frac{1}{L}\end{Vmatrix}-\right.$$

$$-e^{(a-jb)(t-\tau)}\cdot\begin{Vmatrix}-\frac{a}{2jb}-\frac{1}{2} & \frac{1}{2jbC}\\ -\frac{1}{2jbL} & -\frac{R}{2jbL}-\frac{a}{2jb}-\frac{1}{2}\end{Vmatrix}\cdot\begin{Vmatrix}0\\ U_0\frac{1}{L}\end{Vmatrix}\Bigg)d\tau=e^{(a+jb)t}\cdot\begin{Vmatrix}-U_0\left(\frac{a}{2jb}-\frac{1}{2}\right)\\ -\frac{U_0}{2jbL}\end{Vmatrix}-e^{(a-jb)t}\times$$

$$\times\begin{Vmatrix}-U_0\left(\frac{a}{2jb}+\frac{1}{2}\right)\\ -\frac{U_0}{2jbL}\end{Vmatrix}+\int_0^t\left(e^{(a+jb)(t-\tau)}\cdot\begin{Vmatrix}\frac{U_0}{2jbLC}\\ -\frac{U_0\left(\frac{a}{2jb}+\frac{R}{2jbL}-\frac{1}{2}\right)}{L}\end{Vmatrix}-e^{(a-jb)(t-\tau)}\cdot\begin{Vmatrix}\frac{U_0}{2jbLC}\\ -\frac{U_0\left(\frac{a}{2jb}+\frac{R}{2jbL}+\frac{1}{2}\right)}{L}\end{Vmatrix}\right)d\tau.$$

Taking into account $L = 1$, the voltage on the capacitor is calculated as follows:

$$U_C(t)=-e^{(a+jb)t}U_0\left(\frac{a}{2jb}-\frac{1}{2}\right)+e^{(a-jb)t}U_0\left(\frac{a}{2jb}+\frac{1}{2}\right)+\int_0^t\left(\frac{U_0}{2jbC}\cdot e^{(a+jb)(t-\tau)}-\frac{U_0}{2jbC}e^{(a-jb)(t-\tau)}\right)d\tau=$$

$$-e^{(a+jb)t}U_0\left(\frac{a}{2jb}-\frac{1}{2}\right)+e^{(a-jb)t}U_0\left(\frac{a}{2jb}+\frac{1}{2}\right)+\frac{U_0}{2jbC}\left(\frac{e^{at+jbt}-1}{a+jb}-\frac{e^{at-jbt}-1}{a-jb}\right)=-e^{\left(-\frac{R}{2}+j\frac{1}{2}\sqrt{\left|R^2-\frac{4}{C}\right|}\right)t}\times$$

$$\times U_0\left(\frac{-\frac{R}{2}}{j\sqrt{\left|R^2-\frac{4}{C}\right|}}-\frac{1}{2}\right)+e^{\left(-\frac{R}{2}-j\frac{1}{2}\sqrt{\left|R^2-\frac{4}{C}\right|}\right)t}\cdot U_0\left(\frac{-\frac{R}{2}}{j\sqrt{\left|R^2-\frac{4}{C}\right|}}+\frac{1}{2}\right)+\frac{U_0}{jC\sqrt{\left|R^2-\frac{4}{C}\right|}}\left(\frac{e^{\left(-\frac{R}{L}+j\frac{1}{2}\sqrt{\left|R^2-\frac{4}{C}\right|}\right)\cdot t}-1}{-\frac{R}{2}+j\frac{1}{2}\sqrt{\left|R^2-\frac{4}{C}\right|}}-\right.$$

$$\left.-\frac{e^{\left(-\frac{R}{2}-j\cdot\frac{1}{2}\cdot\sqrt{\left|R^2-\frac{4}{C}\right|}\right)\cdot t}-1}{-\frac{R}{2}-j\cdot\frac{1}{2}\cdot\sqrt{\left|R^2-\frac{4}{C}\right|}}\right). \tag{39}$$

Expression (39) describes the pressure distribution in the compressor over time. Since surge is characterized by a sharp change in pressure at the inlet and outlet of the compressor, an increase in the gas temperature in front of the compressor turbine and a decrease in gas generator rotor r.p.m., therefore, it is advisable to use expression (39) to monitoring the pre-surge engine status.

Since the air flow turbulence phenomenon takes place in the compressor of helicopters TE [16, 17], we assume that the useful signal $U_C(t)$ is mixed with noise δ_0 that is not correlated with it. The signal δ is set, it is not correlated with U_C, but correlates in an unknown way with the interference signal δ_0. It is assumed that U_C, δ_0 and δ are statistically stationary, and their average values are equal to zero. The task of the neural network is to process the signal δ in such a way that the signal y at the output of the network is as close as possible to the noise signal δ_0. The error signal ε is determined according to the expression:

$$\varepsilon = U_C + \delta_0 - y. \tag{40}$$

The objective function is presented in the form of the expected value E of the quadratic error:

$$Z=\frac{1}{2}E\left[\Delta^2\right]=\frac{1}{2}E\left[U_C^2+\left(\delta_0-y\right)^2+2\delta\left(\delta_0-y\right)\right]. \tag{41}$$

If we take into account that the signal U_C does not correlate with the noise signal, then the expected value $E\left[\delta\left(\delta_0-y\right)\right]=0$, and the objective function is simplified to the expression

$$Z=\frac{1}{2}\left(E\left[U_C^2\right]+E\left[\left(\delta_0-y\right)^2\right]\right). \tag{42}$$

Since the filter does not change the signal U_C, minimization of the objective function of the error Z is ensured by selecting its parameters in such a way that the value of $E\left[\left(\delta_0-y\right)^2\right]\to\min$. Thus, reaching the minimum of the objective function Z means the best adaptation of the value of y to the

noise δ_0. The minimum possible value $x = E\left[U_C^2\right]$ for which $y = \delta_0$. In this case, the output signal ε corresponds to the useful signal U_C completely denoised.

In the task of monitoring helicopters TE pre-surge status, the input signal and the objective function coincide, that is, the training error of the neural network, which must be minimized, is determined according to the expression:

$$E(w) = \frac{1}{N}\sum_{n=1}^{N} e(n)^2 = \frac{1}{N}\sum_{n=1}^{N}\left(d(n) - y(n)\right)^2. \tag{43}$$

To solve the task of monitoring helicopters TE pre-surge status, we consider the use of a linear neural network (fig. 2), which consists of K neurons located in one layer and connected to **R** inputs through the weight matrix **W**.

For a given network and the corresponding set of input and target vectors, it is possible to calculate the network output vector and form the difference between the output vector and the target vector, which will determine some error [18]. In the training process, it is required to find such values of weights and biases so that the sum of the squares of the corresponding errors is minimal. A supervised training procedure is proposed that uses a training set of the form $\{p_1,t_1\},\{p_2,t_2\},...,\{p_K,t_K\}$, where p_1, p_2, …, p_K – neural network inputs, t_1, t_2, …, t_K – corresponding target outputs. It is required to minimize the following mean square error function:

$$E(k) = \frac{1}{K}\sum_{k=1}^{K}\left(e(k)\right)^2 = \frac{1}{K}\sum_{k=1}^{M}\left(t(k) - a(k)\right)^2. \tag{44}$$

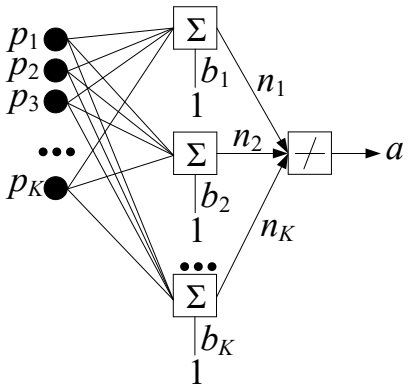


Figure 2: Neural network structure

If signal values at different time intervals are used as an input vector for a neural network, then the linear network is a linear adaptive filter with a finite impulse response. The difference equation describing the communication between the input and output signals of such a filter has the form [20]:

$$y(n) = b_0 x(n) + b_1 x(n-1) + ... + b_P x(n-P); \tag{45}$$

where P – filter order; $x(n)$ – input signal; $y(n)$ – output signal; b_i – filter coefficients.

For a linear neural network, a recurrent training least squares rule is used; it minimizes the mean value of the sum of squares of training errors [19]. You can estimate the total standard error using the standard error of one iteration.

It is known that a standard static neuron implements a non-linear mapping [21, 22]:

$$x_j^{[k+1]} = \psi_j^{[1]} u_j^{[k+1]} = \psi_j^{[1]} \sum_{i=0}^{n^{[1]}} u_{j,i}^{[k+1]} = \psi_j^{[1]} \sum_{i=0}^{n^{[1]}} w_{j,i}^{[1]} x_i^{[1]}; \tag{46}$$

the synaptic weights $w_{j,i}^{[1]}$ of which are subject to refinement in the process of training the neural network.

A nonlinear mapping implemented by a dynamic neuron can be written as:

$$x_j^{[k+1]}(k) = \psi_j^{[1]} u_j^{[k+1]}(k) = \psi_j^{[1]} \sum_{i=0}^{n^{[1]}} u_{j,i}^{[k+1]}(k) = \psi_j^{[1]} \sum_{i=0}^{n^{[1]}} w_{j,i}^{[1]} x_i^{[1]}(k). \tag{47}$$

To train neural networks on dynamic neurons, according to [23], a gradient procedure is used – the back propagation of errors in time.

According to [23], the one-step training criterion is defined as:

$$J(k)=\frac{\left\|e(k)\right\|^2}{2}=\frac{\left\|d(k)-N\left(x^{[1]},w^{[1]},...,w^{[K]}\right)\right\|^2}{2}=\frac{\left\|d(k)-\hat{x}(k)\right\|^2}{2}; \quad (48)$$

where $d(k)$ is the training signal, which in the problem being solved is taken as the current value $x(k)$, that is, the signal $U_C(k)$.

According to [23], the procedure for minimizing the training criterion is:

$$w_{j,i}^{[1]}(k+1)=w_{j,i}^{[1]}(k)-\gamma^{[1]}(k)\frac{\partial J(k)}{\partial u_j^{[1]}(k+1)}\nabla_{w_{j,i}^{[1]}}u_j^{[1]}(k); \quad (49)$$

where $\gamma^{[1]}(k)$ – parameter that determines the training convergence rate.

According to [23], local training error is defined as:

$$\frac{\partial J(k)}{\partial u_j^{[1]}(k+1)}=\delta_j^{[1]}(k). \quad (50)$$

The procedure for setting the neurons of the output layer is [23]:

$$w_{j,i}^{[K]}(k+1)=w_{j,i}^{[K]}(k)+\gamma^{[K]}(k)e_j(k)\psi_j'^{[K]}u_j^{[K]}(k)x_i^{[K]}(k)=w_{j,i}^{[K]}(k)+\gamma^{[K]}(k)e_j(k)J_i^{[K]}(k). \quad (51)$$

The local training error for the hidden layers of the neural network is [23]:

$$\delta_j^{[1]}(k)=\frac{\partial J(k)}{\partial u_j^{[1]}(k+1)}=\sum_{q=1}^{n_{k+1}}\sum_{t=k}^{n_A^{[1]}+k}\delta_q^{[k+1]}(t)\frac{\partial u_q^{[k+1]}(t)}{\partial u_j^{[1]}(k)}=\psi_j'^{[k-1]}u_j^{[1]}(k)\sum_{q=1}^{n_{k+1}}\sum_{t=k}^{n_A^{[1]}+k}\delta_q^{[k+1]}(t)\frac{\partial u_{j,q}^{[k+1]}(t)}{\partial x_j^{[1]}(k)}. \quad (52)$$

An adaptive procedure is written as [23]:

$$w_{j,i}^{[1]}(k+1)=w_{j,i}^{[1]}(k)+\frac{e_j^{[1]}(k)J_i^{[1]}(k)}{\beta+\left\|J_i^{[1]}(k)\right\|^2}; \quad (53)$$

where $\beta > 0$ – regularizing parameter.

The procedure for setting up the training process in hidden layers is optimized for speed. Thus, taking into account [23], the modified neural network training method has the form:

$$\begin{cases} w_{j,i}^{[1]}(k+1)=w_{j,i}^{[1]}(k)+\dfrac{e_j^{[1]}(k)J_i^{[1]}(k)}{\beta_i^{[1]}(k)}; \\ \beta_i^{[1]}(k+1)=\alpha\beta_i^{[1]}(k)+\left\|J_i^{[1]}(k)\right\|^2; \end{cases} \quad (54)$$

where $0 \le \alpha \le 1$ – smoothing parameter, which is selected within a given interval for the most efficient value of the neural network.

A distinctive feature of the modified neural network training method from the method proposed in [23] is the use of a custom smoothing parameter α. The proper choice of the value of the smoothing parameter α is critical to the performance of the neural network. Note that the value of α affects the quality of restoring the output signal of the neural network. The method has been modified in order to impart smoothing properties necessary for processing "noisy" signals, such as $U_C(t)$.

However, a significant limitation [23] is the lack of a description of the criterion for choosing the smoothing parameter. In this paper, we propose to select the smoothing parameter for the task diagnostics (monitoring) helicopters TE pre-surge status in real time.

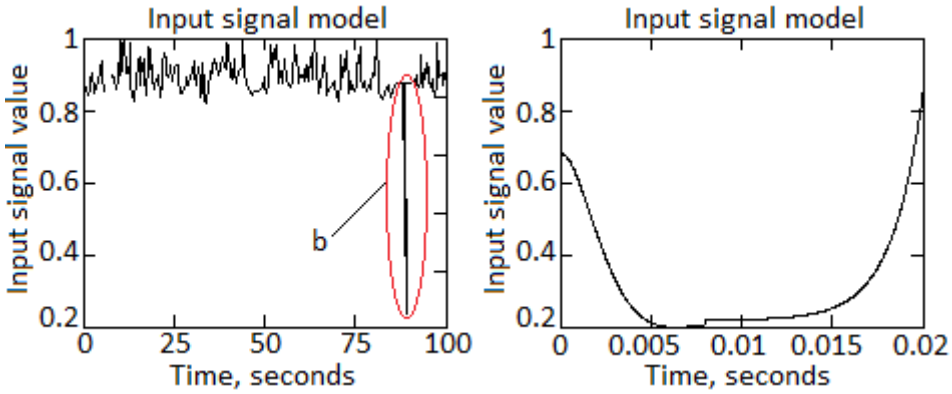
4. Experiment

In the experimental work, the signal $U_C(t)$ obtained according to (39) is used and analyzed for the TV3-117 TE, which is part of the power plant of the Mi-8MTV helicopter, according to the data obtained on board the helicopter during the flight [24] (table 1).

Table 1
Transfer function coefficient values

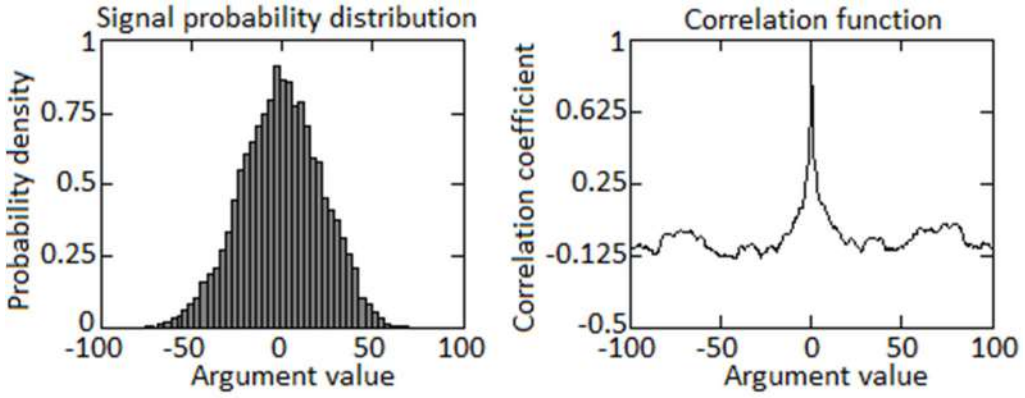
Number	λ	ω	δ	c
1	0.985	0.992	0.973	0.999
2	0.989	0.991	0.976	0.999
3	0.987	0.996	0.977	0.999
4	0.974	0.983	0.971	0.999
5	0.977	0.982	0.968	0.999
6	0.972	0.987	0.978	0.999
7	0.981	0.988	0.973	0.999
8	0.983	0.993	0.975	0.999
9	0.992	0.990	0.977	0.999
...	...	...	...	...
256	0.985	0.991	0.974	0.999

To conduct experimental researches (test example), the Matlab application package was used. The diagram of the dependence of the signal amplitude on time for the signal under study is shown in fig. 3, where a – signal with noise taken into account, b – simulated area of a sharp pressure drops [12]. On fig. 3 input signal values are given in absolute units. Fig. 4 shows a diagram of the probability distribution for a noise signal, where a – signal probability distribution (Gaussian form), b – signal correlation function.



a b

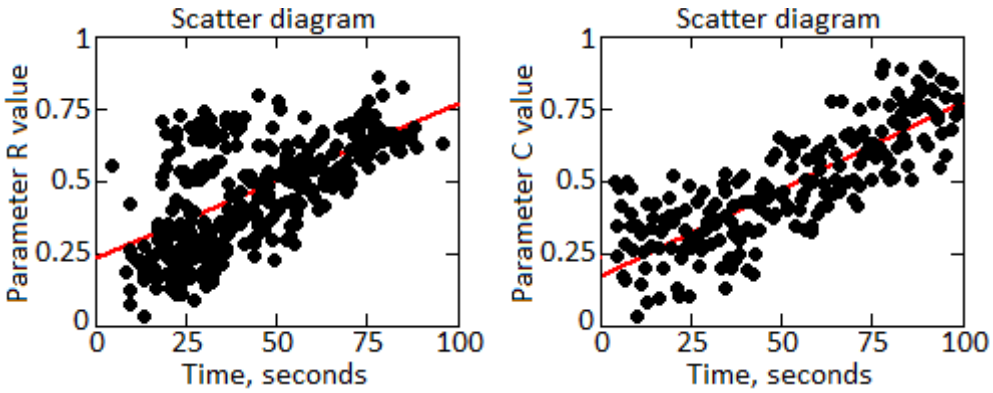
Figure 3: Input signal diagram: a – signal with noise taken into account; b – sudden pressure drop area [12]



a b

Figure 4: Input signal diagram: a – signal probability distribution; b – signal correlation function

To form the training and test subsets, cross-validation [25] was used to estimate the values of TV3-117 TE parameters, the results of which are shown in fig. 5.



a b

Figure 5: Scatter diagram of input parameters: a – parameter *R*; b – parameter *C*

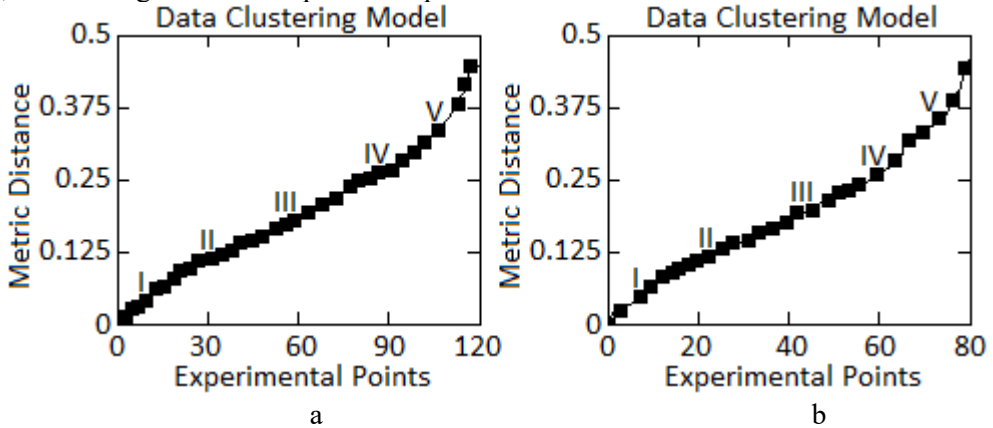
As the input signal U_C for the neural network (fig. 2), a sequential sampling of the values of the fragment of the noise signal was used. The delay line consists of 10 blocks, that is, a sequence of $U_C/10$ values is fed to the input of each neuron. The output signal of the network was the sum of the output signals of each of the neurons. The network had one layer of neurons with a linear activation function. The number of neuron inputs was equal to the sample length of the studied signal. The initial weights of neurons were initiated by random values, and the bias was chosen to be the same and equal to $b = 0.027$.

A supervised training algorithm was used to train the network, and a sequence of input signal values was used as the target vector. The absolute network training error is calculated according to (43): $E(n)=\frac{1}{N}\sum_{n=1}^{N}\left(e(n)\right)^2=\frac{1}{N}\sum_{k=1}^{M}\left(Y(n)-U_C(n)\right)^2$, where $Y(n)$ – output signal of the network, $U_C(n)$ – neural network input signal values, N – input vector dimension. The relative error of the output signal is determined according to the expression:

$$\delta E=\frac{E(n)}{\sigma^2}; \tag{55}$$

where $\sigma^2 = 0.00028$ – neural network input signal variance.

The neural network was trained on a sample of 100 input values of the U_C signal, which is $3T_{cor}$. The signal was applied to the input of each of 10 neurons with 10 signal values on the segment [0; 100], the neural network training rate was chosen as 0.1 (this value was chosen according to the same principles as the number of neurons). Training was carried out at different values of training cycles (depending on the experiment, values from 1 to 100 were used). In order to establish the representativeness of the training and test samples, a cluster analysis [26] of the initial data was carried out (table 1), during which eight classes were identified (fig. 6, a). After the randomization procedure, the actual training (control) and test samples were selected (in a ratio of 2:1, that is, 67 % and 33 %). The process of clustering the training (fig. 6, b) and test samples shows that they, like the original sample, contain eight classes each. The distances between the clusters practically coincide in each of the considered samples, therefore, the training and test samples are representative.



a b

Figure 6: Clustering results: a – initial experimental sample (I…V – classes); b – training sample

The assessment of the homogeneity of the training and test samples is carried out using the Fisher-Pearson χ^2 criterion [27] with $r-k-1$ degrees of freedom:

$$\chi^2 = \min_{\theta} \sum_{i=1}^{r} \left(\frac{m_i - np_i(\theta)}{np_i(\theta)} \right); \tag{56}$$

where θ – maximum likelihood estimate found from the frequencies $m_1, \ldots, m_r$; n – number of elements in the sample; $p_i(\theta)$ – probabilities of elementary outcomes up to some indeterminate k-dimensional parameter θ.

The above-mentioned statistics $\chi 2$ permits, under the above assumptions, to check the hypothesis about the representability of sample variances and covariance of factors contained in the statistical model. The field of hypothesis acceptance is $\chi^2 \le \chi_{n-m,\alpha}$, where α – significance level of the criterion.

The results of calculations in accordance with (56) are in table 2.

Table 2

Part of the training sample during the helicopters TE operation (on the example of TV3-117 TE)

Number	$P(\lambda)$	$P(\omega)$	$P(\delta)$	$P(c)$
1	0.739	0.784	0.798	0.989
2	0.742	0.783	0.800	0.989
3	0.740	0.787	0.801	0.989
4	0.731	0.776	0.796	0.989
5	0.733	0.779	0.794	0.989
6	0.729	0.780	0.802	0.989
7	0.736	0.781	0.798	0.989
8	0.738	0.784	0.800	0.989
9	0.744	0.782	0.801	0.989
...	...	...	...	...
256	0.739	0.783	0.799	0.989

Calculation of the $\chi 2$ value based on the observed frequencies $m_1, \ldots, m_r$ (summing line by line the probabilities of the outcomes of each measured value) and comparing it with the critical values of the distribution $\chi 2$ with the number of degrees of freedom $r - k - 1$. In this article, with the number of degrees of freedom $r - k - 1 = 13$ and $\alpha = 0.05$, the random variable $\chi 2 = 3.588$ did not exceed the critical value from table 4 is 22.362, which means that the hypothesis of the normal distribution law can be accepted and the samples are homogeneous.

5. Results

Table 3 shows the results of the neural network operation for various values of the smoothing parameter. The network teacher is the value 0.989, which is the average value of the input sample (table 1).

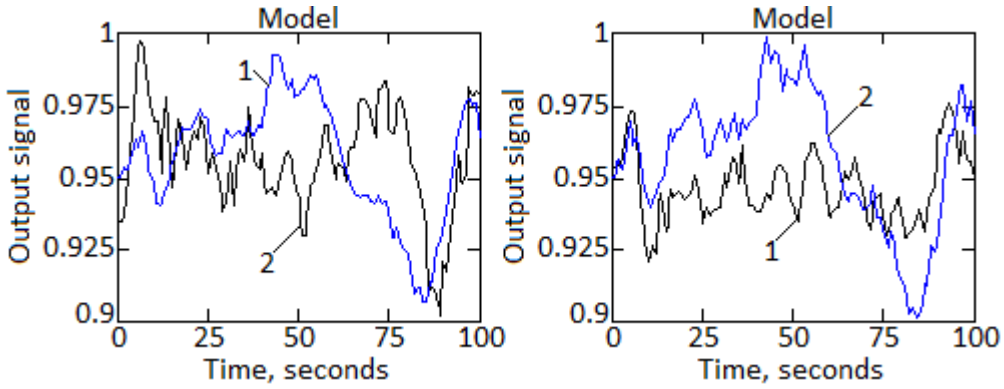
Table 3

The results of determining the smoothing parameter

$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1.0$
0.9951	0.9945	0.9934	0.9926	0.9921	0.9915	0.9912	0.9889	0.9873	0.9862

The results show that at $\alpha = 0.8$, the predicted value is closest to the teacher by 99.99 %.

The neural network training error depending on the number of training cycles is illustrated by the diagram in fig. 7 (a – diagram corresponds to 5 training cycles, b – diagram corresponds to training cycles), which show the input and output signals of the neural network resulting from training, where 1 – initial signal U_C, 2 – neural network output signal.



a b

Figure 7: Diagram of neural network output and signal versus time

Fig. 7, a corresponds to an absolute error of 0.004717 and a neural network training time of 1.5 seconds. Fig. 7, b corresponds to an absolute error of 0.000000638 and a neural network training time of 11.8 seconds. A diagram of the dependence of the neural network training error on the number of training cycles is shown in fig. 8, from which it can be seen that the error decreases with an increase in the number of training cycles approximately according to a linear law. The training quality of the neural network was also assessed using regression analysis of neural network input and output signals.

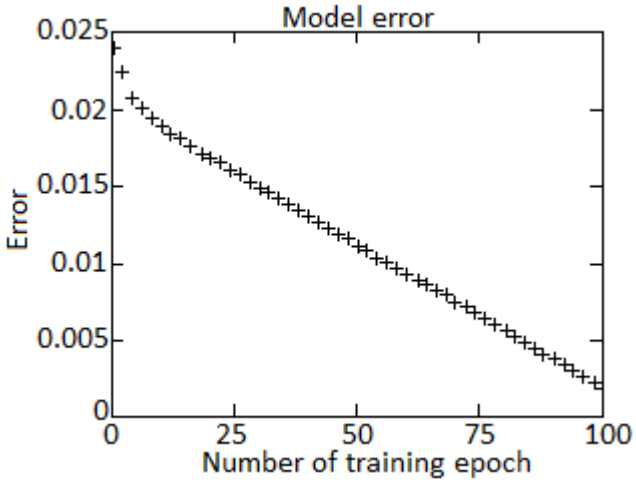


Figure 8: Diagram of neural network training error dependence on the number of cycles

The dependence of neural network training error on the number of neurons is shown in fig. 9, from which it can be seen that the training error decreases with an increase in the number of neurons.

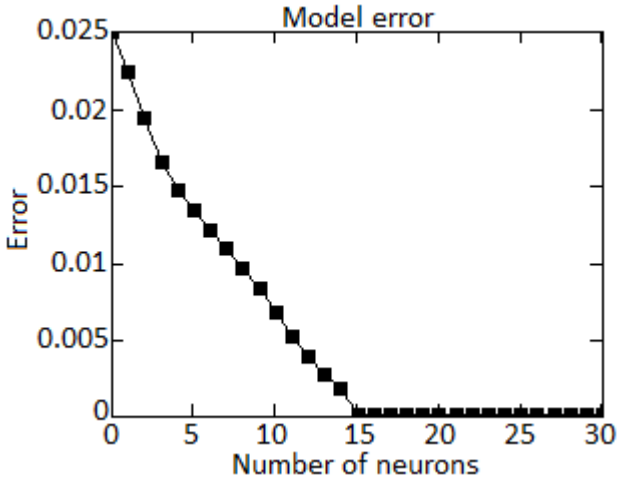


Figure 9: Neural network training error versus number of neurons diagram

When the number of neurons is more than 15, it decreases to a minimum and ceases to change. At the same time, the training time increases from 3 seconds with 3 neurons to 150 seconds with 15 neurons. In further experiments, to increase the training rate, the number of neurons was chosen to be 10.

To check the independence of the residuals, the Durbin-Watson test, the autocorrelation function, etc. are usually used. [28, 29]. To check the normality of the distribution of residuals, a plot of the normal distribution of residuals was constructed, that is, the correlation field between the target vector U_C (input signal values at each point) and the output of the neural network Z after 50 training cycles (fig. 9). It can be seen that the distribution is normal – almost all observations follow the line, the points are grouped near the straight line in fig. 10. A high value of the correlation coefficient R = 0.999 indicates that the neural network training algorithm was chosen correctly.

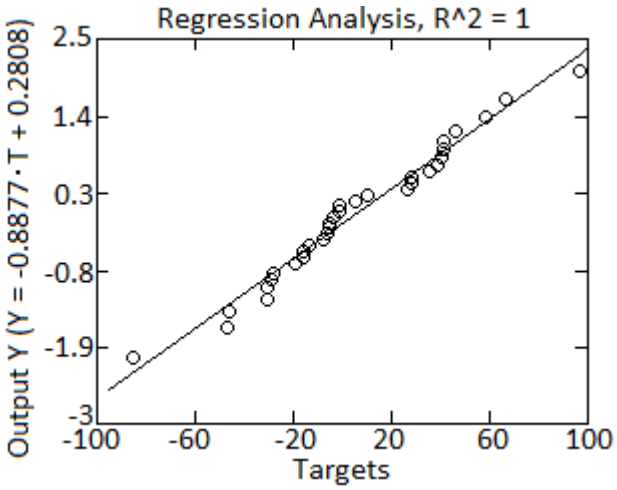
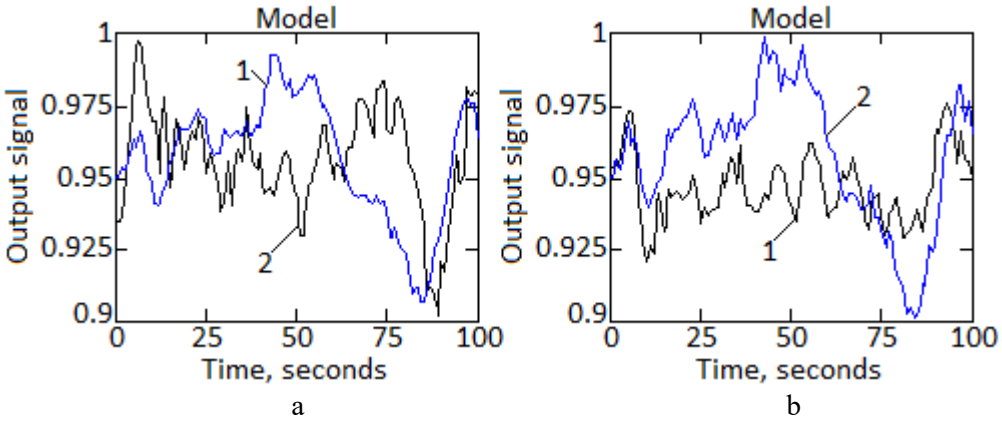


Figure 10: Normal distribution diagram of residuals (Regression Analysis)

6. Discussions

The neural network input signal and neural network simulated output signal after training on a sample with a length of 100 reports ($3T_{cor}$) are shown in fig. 11, where 1 – initial signal, 2 neural network – output signal. Fig. 11, a show the signals immediately after training the neural network, fig. 11, b – after a time interval corresponding to 30 input noise correlation times. From fig. 11 it can be seen that, despite the increase in the time interval for diagnosing (monitoring) the neural network, the developed method makes it possible to determine the helicopters TE pre-surge status.



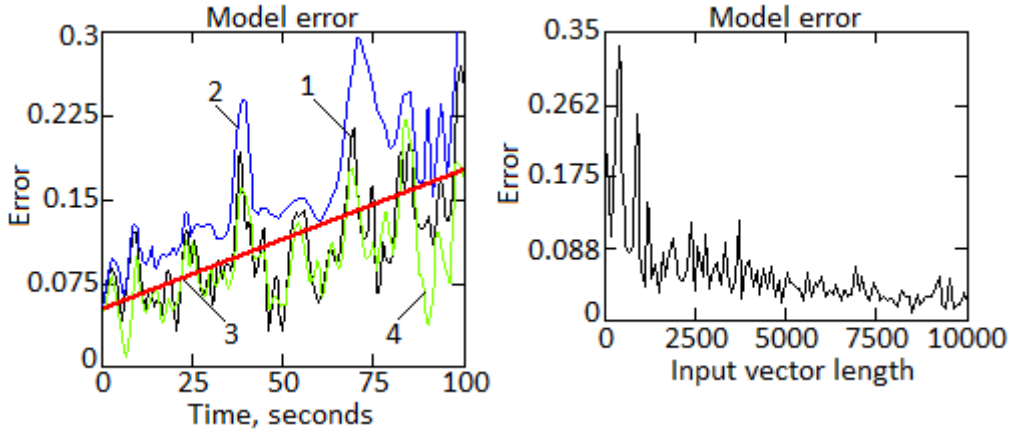
a b

Figure 11: Diagram of research of TV3-117 TE pre-surge status

Fig. 12 shows the results of determining the error in diagnostics (monitoring) of helicopters TE pre-surge status using the developed neural network, where 1 – initial signal U_C error, 2 – upper error limit, 3 – lower error limit, 4 – trend line.

As can be seen from fig. 12, a, the maximum error in diagnostics (monitoring) of helicopters TE pre-surge status does not exceed 0.3 %, which indicates an accuracy of > 99 % of the developed method.

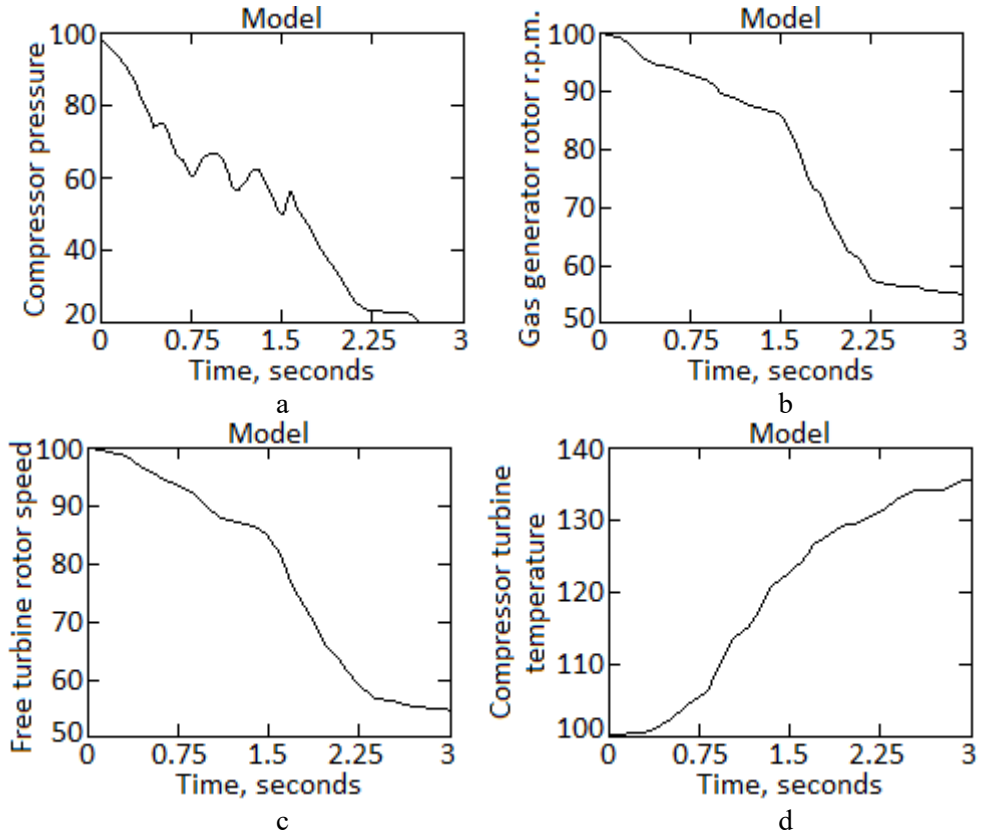
To reduce the training time, the neural network went through 3 training cycles, and the error was measured at a prediction time of $300T_{cor}$, while the diagnostics (monitoring) error remained unchanged. From fig. 12, b it can be seen that with an increase in the length of the input vector, the relative network training error decreases significantly. The training time increases with the length of the input vector: with a length equal to $3T_{co}$, the training time is 3.5 seconds, and with a length of 100 correlation times it increases to 60 seconds.



a b

Figure 12: Diagram for determining the diagnostics (monitoring) error: a – diagram of the dependence of the diagnostics (monitoring) error on time; b – diagram of diagnostics (monitoring) error versus input vector length

Fig. 13 shows the results of modeling helicopters TE surge status.



a b c d

Figure 13: The resulting diagram of monitoring (diagnostics) of helicopters TE surge by a neural network

The research results displayed in fig. 13 are based on the physical processes occurring in the engine path – using thermogas-dynamic indicators obtained using a universal mathematical model of the engine [30, 31], while taking into account that the surge, as a rule, is accompanied by the following phenomena:

– fluctuations in pressure, velocities and gas flow rates along the path with a pronounced pressure drop downstream of the compressor relative to the pressure at its inlet (fig. 13, a);

– gas generator rotor r.p.m. decrease (n_{TC}) (fig. 13, b);

– free turbine rotor rotation frequency decrease (n_{FT}) (fig. 13, c);

– gases temperature in front of the compressor turbine increase (T_G) (fig. 13, d).

Fig. 13 corresponds to: a – diagram of pressure changes behind the compressor during surge; b – diagram of the dynamics of gas generator rotor r.p.m. during surge; c – diagram of the dynamics of free turbine rotor rotation frequency during surge; d – diagram of changes in gases temperature before the compressor turbine during surging.

According to the results of modeling helicopters TE surge status by a neural network (fig. 13), a division into classes of statuses was carried out [25, 32] (I (red) – surge is present; II (green) – there is no surge; III (blue) – there is a risk of surge), according to the dynamics of changes in the values of engine thermogas-dynamic parameters (Fig. 14).

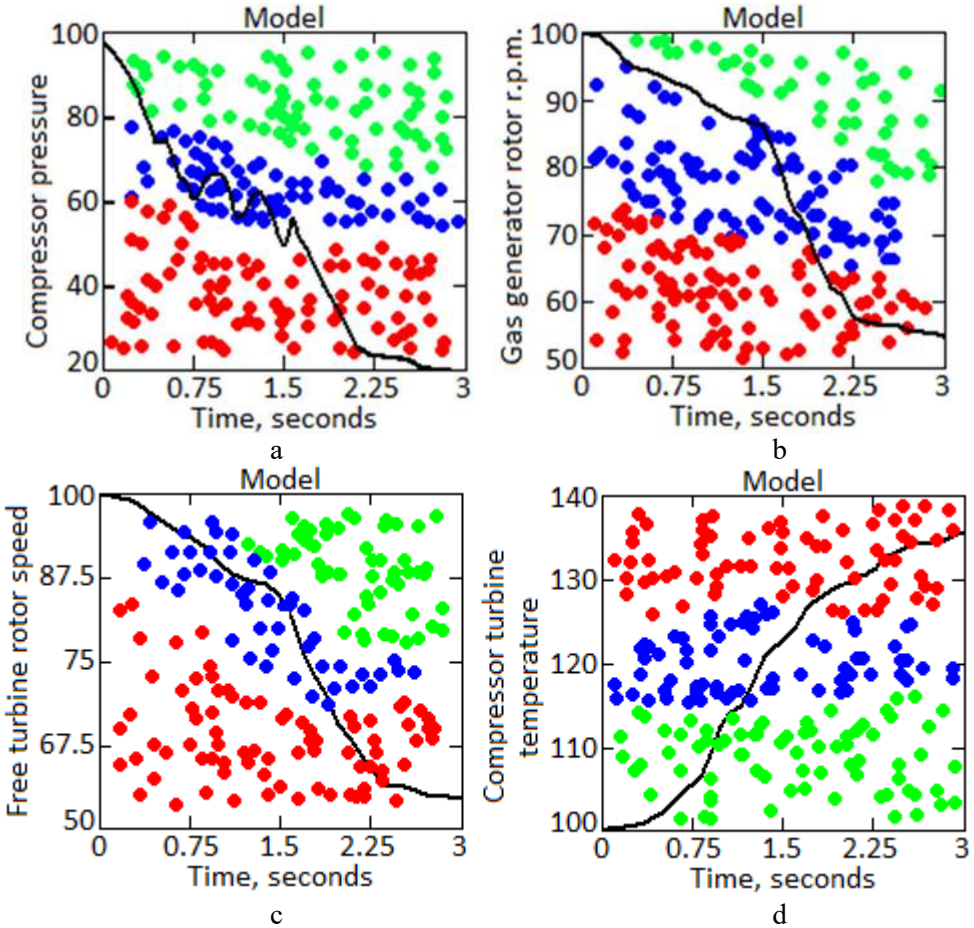


Figure 14: The resulting diagram of helicopters TE classification operational status by a neural network

Fig. 14 corresponds to: a – indicator of pressure changes behind the compressor; b – indicator of gas generator rotor r.p.m.; c – indicator of free turbine rotor rotation frequency; d – indicator of gases temperature before the compressor turbine.

A comparative analysis of the accuracy of the classical and neural network methods for diagnostics (monitoring) of helicopters TE (using the TV3-117 engine as an example) pre-surge status is given in table 4 which displays the probabilities of errors of the 1st and 2nd kind [33, 34] when determining the dynamics of changes in engine thermogas-dynamic parameters (according to fig. 13).

Table 4
Comparative characteristics of methods

Method of determination	Probability of error in determining							
	By parameter U_C		By parameter n_{TC}		By parameter n_{FT}		By parameter T_G	
	Type 1st error	Type 2nd error	Type 1st error	Type 2nd error	Type 1st error	Type 2nd error	Type 1st error	Type 2nd error
Classic	1.78	1.61	1.81	1.74	2.05	1.86	1.74	1.55
Neural Network	0.69	0.62	0.70	0.65	0.80	0.72	0.68	0.60

7. Conclusions

1. A mathematical model was further developed that describes the time distribution of pressure in helicopters aircraft turboshaft engines compressor, which, by obtaining a universal expression for determining the distribution of pressure values in the engine compressor, made it possible to develop a neural network method for diagnostics (monitoring) helicopters turboshaft engines pre-surge status in real time.

2. For the first time, a neural network method for diagnostics (monitoring) helicopters turboshaft engines pre-surge status in real time was developed, based on a linear neural network trained using a modified method with direct transmission of information on dynamic neurons, which makes it possible to classify helicopters turboshaft engines statuses in the helicopter flight mode for the presence, absence or risk of surge.

3. The method of training neural networks with direct transmission of information on dynamic neurons was further developed, which, due to the adjustable smoothing parameter from the range from zero to one, made it possible to obtain the predicted output signal value that is closest to the teacher by 99.99 %.

4. It is shown that the errors of the 1st and 2nd kind of the method for diagnostics (monitoring) helicopters turboshaft engines pre-surge status using a linear neural network did not exceed 0.80 % and 0.72 %, respectively, while for the classical method they amounted to 2.05 % and 1.86 %, respectively. The obtained results prove that the application of the developed neural network method will make it 38.77 % more accurate to determine helicopters turboshaft engines current status at helicopter flight mode for the presence or development of surge.

5. The prospect for further research is the implementation of the results of the obtained studies, as well as the developed method for diagnostics (monitoring) helicopters turboshaft engines pre-surge status, into the onboard neural network expert system for integrated monitoring and operation control of helicopters turboshaft engines at helicopter flight mode [35].

The prospects for further research are the integration of this method into self-organizing Kohonen maps, which will improve the method for detecting engine surge using Kohonen self-organizing maps, developed by Viktor Dubrovin and Tetiana Kiprych.

8. References

[1] S.-B. Yang, X. Wang, H.-N. Wang, Y.-G. Li, Sliding mode control with system constraints for aircraft engines, ISA Transactions, vol. 98 (2020) 1–10. doi: 10.1016/j.isatra.2019.08.020

[2] W. Jiang, Y. Xu, Z. Chen, N. Zhang, X. Xue, J. Zhou, Measurement of health evolution tendency for aircraft engine using a data-driven method based on multi-scale series reconstruction and adaptive hybrid model, Measurement, vol. 199 (2022) 111502. doi: 10.1016/j.measurement.2022.111502

[3] H. Yu, S. Zhensheng, C. Lijia, Z. Yin, P. Pengfei, Optimization configuration of gas path sensors using a hybrid method based on tabu search artificial bee colony and improved genetic algorithm in turbofan engine, Aerospace Science and Technology, vol. 112 (2021) 106642. doi: 10.1016/j.ast.2021.106642

[4] U. Ahmed, F. Ali, I. Jennions, A review of aircraft auxiliary power unit faults, diagnostics and acoustic measurements, Progress in Aerospace Sciences, vol. 124, no. 1 (2021) 100721. doi: 10.1016/j.paerosci.2021.100721
[5] Y.-Z. Chen, E. Tsoutsanis, C. Wang, L.-F. Gou, A time-series turbofan engine successive fault diagnosis under both steady-state and dynamic conditions, Energy, vol. 263, part D (2023) 125848. doi: 10.1016/j.energy.2022.125848
[6] M. Hruz, P. Pecho, V. Socha, M. Bugaj, Use of the principal component analysis for classification of aircraft components failure conditions using vibrodiagnostics, Transportation Research Procedia, vol. 59 (2021) 166–173. doi: 10.1016/j.trpro.2021.11.108
[7] 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
[8] M. Lungu, R. Lungu, Automatic control of aircraft lateral-directional motion during landing using neural networks and radio-technical subsystems, Neurocomputing, vol. 171 (2016) 471–481 doi: 10.1016/j.neucom.2015.06.084
[9] S. Kiakojoori, K. Khorasani, Dynamic neural networks for gas turbine engine degradation prediction, health monitoring and prognosis, Neural Computing & Applications, vol. 27, no. 8 (2016) 2151–2192. doi: 10.1007/s00521-015-1990-0
[10] S. Pang, Q. Li, H. Zhang, A new online modelling method for aircraft engine state space model, Chinese Journal of Aeronautics, vol. 33, issue 6 (2020) 1756–1773. doi: 10.1016/j.cja.2020.01.011
[11] R. Andoga, L. Fozo, M. Schrotter, M. Ceskovic, Intelligent thermal imaging-based diagnostics of turbojet engines, Applied Sciences, vol. 9, no. 11 (2019) 2253. doi: 10.3390/app9112253
[12] Y. Shmelov, S. Vladov, A. Kryshan, S. Gvozdik, Simulation of transients of gas-parameters flow in compressor of aviation engine TV3-117, Transactions of Kremenchuk Mykhailo Ostrohradskyi National University, issue 4/2018 (111) (2018) 36–42. doi: 10.30929/1995-0519.2018.4.36-42.
[13] G. Coppola, A. Veldman, Global and local conservation of mass, pulse and kinetic energy in the simulation of compressible flow, Journal of Computational Physics, vol. 475 (2023) 111879. doi: 10.1016/j.jcp.2022.111879
[14] J. C. Pozo, V. Vergara, A non-local in time telegraph equation, Nonlinear Analysis, vol. 193 (2020) 111411. doi: 10.1016/j.na.2019.01.001
[15] P. Veigend, G. Necasova, V. Satek, Model of the telegraph line and its numerical solution, Open Computer Science, vol. 8 (2018) 10–17. doi: 10.1515/comp-2018-0002
[16] Y. Shmelov, S. Vladov, Y. Klimova, Simulation gas-dynamic processes occurring in the helicopter engine MI-8MTV, Scientific notes of V.I. Vernadsky Taurida National University. Series: Technical science, vol. 29 (68), no. 2 (2018) 29–34.
[17] M. Adamowicz, G. Zywica, Advanced gas turbines health monitoring systems, Diagnostyka, vol. 19, issue 2 (2018) 77–87. doi: 10.29354/diag/89730
[18] E. Hedrick, K. Hedrick, D. Bhattacharyya, S. E. Zitney, B. Omell, Reinforcement learning for online adaptation of model predictive controllers: Application to a selective catalytic reduction unit, Computers & Chemical Engineering, vol. 160 (2022) 107727 doi: 10.1016/j.compchemeng.2022.107727
[19] R. Vang-Mata, Multilayer Perceptrons: Theory and Applications, New York, Nova Science Publishers (2020) 143.
[20] A. Sarraf, S. Khalili, An upper bound on the variance of scalar multilayer perceptrons for log-concave distributions, Neurocomputing, vol. 488, no. 1 (2022) 540–546. doi: 10.1016/j.neucom.2021.11.062
[21] W. Mlynarski, M. Hledik, T. R. Sokolowski, G. Tkacik, Statistical analysis and optimality of neural systems, Neuron, vol. 109, issue 7 (2021) 1227–1241.e5 doi: doi.org/10.1016/j.neuron.2021.01.020
[22] C. Qin, J. Wang, H. Zhu, J. Zhang, S. Hu, D. Zhang, Neural network-based safe optimal robust control for affine nonlinear systems with unmatched disturbances, Neurocomputing, vol. 506 (2022) 228–239. doi: doi.org/10.1016/j.neucom.2022.07.072
[23] T. Chepenko, Time series predicting methods based on artificial neural networks with time delay elements, Kharkiv National University of Radio Electronics. URL: https://openarchive.nure.ua/server/api/core/bitstreams/dd87fbf2-2b7c-49b8-bf6e-ad2beee80361/content
[24] S. Vladov, Y. Shmelov, R. Yakovliev, Optimization of Helicopters Aircraft Engine Working Process Using Neural Networks Technologies. COLINS-2022: 6th International Conference on

Computational Linguistics and Intelligent Systems, May, 12–13, 2022, Gliwice, Poland. CEUR Workshop Proceedings (ISSN 1613-0073), vol. 3171 (2022) 1639–1656.
[25] S. Vladov, Y. Shmelov, R. Yakovliev, Methodology for Control of Helicopters Aircraft Engines Technical State in Flight Modes Using Neural Networks, The Fifth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2022), May, 12, 2022, Zaporizhzhia, Ukraine, CEUR Workshop Proceedings (ISSN 1613-0073) vol. 3137 (2022) 108–125. doi: 10.32782/cmis/3137-10
[26] S. Vladov, Y. Shmelov, R. Yakovliev, Modified Helicopters Turboshaft Engines Neural Network On-board Automatic Control System Using the Adaptive Control Method. ITTAP'2022: 2nd International Workshop on Information Technologies: Theoretical and Applied Problems, November 22–24, 2022, Ternopil, Ukraine. CEUR Workshop Proceedings (ISSN 1613-0073), vol. 3309 (2022) 205–229.
[27] H.-Y. Kim, Statistical notes for clinical researchers: Chi-squared test and Fisher's exact test, Restor Dent Endod, vol. 42, no. 2 (2017) 152–155. doi: 10.5395/rde.2017.42.2.152
[28] H. Mark, J. Workman Jr., Chapter 65 – Linearity in Calibration, Act III Scene II: A Discussion of the Durbin-Watson Statistic, a Step in the Right Direction, Chemometrics in Spectroscopy (Second Edition), (2018) 433–440. doi: 10.1016/B978-0-12-805309-6.00065-9
[29] P. Carpena, A. V. Coronado, On the Autocorrelation Function of 1/f Noises, Mathematics, vol. 10 (2022) 1416. doi: 10.3390/math10091416
[30] O. Avrunin, S. Vladov, M. Petchenko, V. Semenets, V. Tatarinov, H. Telnova, V. Filatov, Y. Shmelov and N. Shushlyapina. Intelligent automation systems, Kremenchuk, Novabook (2021) 322. doi: 10.30837/978-617-639-347-4
[31] S. Pang, Q. Li, B. Ni, Improved nonlinear MPC for aircraft gas turbine engine based on semi-alternative optimization strategy, Aerospace Science and Technology, vol. 118 (2021) 106983. doi: 10.1016/j.ast.2021.106983
[32] L. Zhao, S. H. Goh, Y. H. Chan, B. L. Yeoh, H. Hu, M. H. Thor, A. Tan, J. Lam, Optimization of an Artificial Neural Network System for the Prediction of Failure Analysis Success, Microelectronics Reliability, vol. 92 (2019) 136–142. doi: 10.1016/j.microrel.2018.11.014
[33] J. Park, H. E. Kim, I. Jang, Empirical estimation of human error probabilities based on the complexity of proceduralized tasks in an analog environment, Nuclear Engineering and Technology, vol. 54, issue 6 (2022) 2037–2047. doi: 10.1016/j.net.2021.12.025
[34] A. Silva, M. Gouvea, Study on the effect of sample size on type I error, in the first, second and first-two digits excessmad tests, International Journal of Accounting Information Systems, vol. 48 (2023) 100599. doi: 10.1016/j.accinf.2022.100599
[35] Y. Shmelov, S. Vladov, Y. Klimova, M. Kirukhina, Expert system for identification of the technical state of the aircraft engine TV3-117 in flight modes, System Analysis & Intelligent Computing: IEEE First International Conference on System Analysis & Intelligent Computing (SAIC), 08–12 October 2018. 77–82. doi: 10.1109/SAIC.2018.8516864

Research and Software Implementation of Intelligent Method of Energy Consumption Control

Yulii Horichenko[a], Anzhelika Parkhomenko[a], Oleg Pozdnyakov[a], Artem Tulenkov[a], Yaroslav Zalyubovskiy[a], Olga Gladkova[a] and Andriy Parkhomenko[a]

[a] *National University Zaporizhzhia Polytechnic, 64, Zhukovskogo str., Zaporizhzhia, 69063, Ukraine*

Abstract

The popular intelligent methods for improving the efficiency of energy consumption in a home automation system have been investigated in this paper. The intelligent method for controlling energy consumption has been developed. It differs from the existing ones in an integrated approach based on predicting electricity generation using a Feedforward neural network, as well as optimizing the schedule for using electrical appliances based on an improved particle swarm optimization algorithm with the proposed system of priorities. The intelligent support subsystem has been created, the integration of which into the home automation system will allow users to control the processes of generation and consumption of electricity, effectively use household appliances and save resources.

Keywords

Home automation system, energy consumption, intelligent method, artificial neural networks, forecasting, schedule optimization

1. Introduction

In recent years, there has been an active development of intelligent technologies, that are actively being implemented in home automation systems (HAS), which allows bringing the standard system to a new level of comfort [1]. The implementation of intelligent methods makes it possible to study the habits of the residents of the house, predict their behavior and create optimal conditions for life [2]-[3]. One of the most important areas of intelligent technologies usage is the saving of resources, in particular electricity, which is currently a significant objective, especially in the conditions of the unstable economic and political situation in Ukraine.

A large number of HASs are offered on the market, in particular, the most popular are: OpenHAB, HomeAssistant, Majordomo, ioBroker, Domoticz, Jeedom. Their functionality is aimed at monitoring and automated management of various processes in houses and apartments. Thus, household equipped with such a system offers more comfort, flexibility and security [4]-[5]. But at the same time, the existing HASs do not provide intellectual support to users from the point of view of organizing the efficient consumption of electricity generated by alternative sources of electricity (solar and wind power plants) based on the optimal usage of various electrical devices. This would allow residents to gain independence from the external power grid and utility tariffs for electricity and significantly save money by using alternative energy sources.

Thus, the objective of developing a subsystem of intellectual support that will allow considering the interests and habits of household residents and saving resources at the same time is urgent.

CMIS-2023: The Sixth International Workshop on Computer Modeling and Intelligent Systems, Zaporizhzhia, Ukraine, May 3, 2023
EMAIL: princepersiam@gmail.com (Y. Horichenko); parkhomenko.anzhelika@gmail.com (A. Parkhomenko); oleg.pozdnyakov.ua@gmail.com (O. Pozdnyakov); aetulenkov@gmail.com (A. Tulenkov); zalubovskiy.yar@gmail.com (Y. Zalyubovskiy); gladolechka@gmail.com (O. Gladkova); andriy2872073@gmail.com (A. Parkhomenko)
ORCID: 0009-0004-4506-9631 (Y. Horichenko); 0000-0002-6008-1610 (A. Parkhomenko); 0009-0006-3955-802X (O. Pozdnyakov); 0000-0003-4863-4144 (A. Tulenkov); 0000-0002-6847-8778 (Y. Zalyubovskiy); 0000-0002-6834-2854 (O. Gladkova); 0000-0001-8265-0530 (A. Parkhomenko)

The goal of the work is the research and practical implementation of intelligent methods of energy consumption control to increase the efficiency of electrical appliances usage and save electricity in HAS that uses alternative sources of electricity.

2. Related works

As the conducted studies have shown, today the most popular intelligent methods, algorithms and models in the field of efficient energy consumption organization are: particle swarm optimization (PSO) algorithm, genetic algorithm (GA), ARIMA model and artificial neural networks.

In order to reduce electricity costs, researchers suggest using the PSO to optimize the schedule of electrical appliances and solve the objective of minimizing the peak load in the power grid [6]-[10]. The authors of works [11]-[13] use a GA to increase the efficiency of energy consumption in HAS. In the study [14], the authors proposed the usage of the ARIMA method for real-time analysis of electricity consumption in HAS and forecasting of future energy consumption. In works [15]-[22], the authors apply neural networks to predict electricity consumption, in order to use this data in the future to reduce electricity costs and optimize the operation schedule of household electrical appliances. In particular, the authors of [22] investigated the issue of using the main types of neural networks to solve various objectives in HAS. The method for choosing the optimal type of artificial neural network for the same set of historical data was proposed in [22], because of the fact that each artificial neural network has certain features and allows predicting some values and processing various types of data with different accuracy.

Thus, as a result of the review, the existing objectives of organizing efficient energy consumption in HAS and popular methods of solving them were divided into two main groups (Table 1).

Table 1
Objectives and methods for their solution

Objective	Methods
Optimization of the work schedule of household appliances	Particle swarm optimization algorithm [6]-[10] Genetic algorithm [11]-[13]
Forecasting electricity consumption or generation	ARIMA [14], [19] Long Short-Term Memory neural network [15]-[18], [22] Recurrent neural network [19], [20], [22] Feedforward neural network [17], [22] Convolutional neural network [21]

In the considered works, the objective of optimizing the operation schedule of household appliances was solved by automatically scheduling their usage during the period of the day when electricity has a low price. It allows solving the objective of minimizing the peak load in the power grid and reducing utility costs for electricity. The results obtained in studies [6]-[13] showed that the implementation of this approach made it possible to reduce energy costs even up to 55%, which is a fairly high indicator. However, the biggest disadvantage of such scheduling optimization is that the level of comfort of residents can be significantly reduced, because of the fact that it is recommended to use more electrical appliances only at certain times of the day (for example, at night), which is not always convenient. In addition, this approach is not justified if a fixed electricity tariff is set. In the case when the studied HAS uses alternative energy sources (solar and wind power plants), then the objective statement can be changed to optimal planning for the use of electrical appliances during periods when electricity generation from alternative sources is maximum.

Studies have shown that to solve the objective of optimizing the work schedule of household appliances, it is suggested to use the PSO and the GA [6]-[13]. In particular, works [9], [12] compared these two approaches, and the results showed that based on the usage of the PSO it was possible to reduce electricity costs by 8% more than on the basis of the GA, using the same data set. The authors of [23] compared the performance of these approaches and showed that the PSO requires fewer

iterations to find the optimal solution and its accuracy is higher. Thus, it can be concluded that the usage of the PSO [24]-[25] is appropriate for achieving the goal of this work.

The objective of analyzing historical data and forecasting electricity generation assumes that the forecasted data will help in the future to develop a plan for economical usage of electrical appliances and to reduce electricity consumption in HAS. Research by the authors [15]-[22] showed that the highest forecasting accuracy that was achieved is 97%, which is a very high indicator and it allows the implementation of the effective energy consumption analysis and forecasting subsystem. If HAS uses alternative sources of electricity, the forecasted data about energy generation will be used by the optimization algorithm to create an optimal schedule for household appliances, which will allow the user to plan the schedule for the upcoming day.

The conducted studies showed that a number of methods are proposed to be used for forecasting electricity generation: ARIMA, Long Short-Term Memory (LSTM) neural network, Recurrent neural network (RNN), Feedforward neural network (FNN) and Convolutional neural network (CNN). The usage of the ARIMA method and CNN is not advisable, because in studies [19], [21] they showed unsatisfactory results in comparison with other methods. Studies [22] show that it is impossible to create a universal neural network that will be suitable for various objectives of HAS. Therefore, it is necessary to conduct a number of computer experiments and choose a neural network model that will provide the highest prediction accuracy for the studied HAS, taking into account its specific features.

3. Main and anticipated findings

3.1 Analysis and selection of a neural network model

The training set comprises the concatenation of input attributes from the weather forecast obtained through a weather API and the output parameter, i.e., electricity generation. The resulting dataset contains 2,160 records and spans the period from August 8, 2022 to November 13, 2022.

In the course of this research, the following models were considered: FNN [26], RNN and a neural network with LSTM. The implementation of FNN is represented as layers in a sequential order, and data is transferred from one layer to another in a given order until they reach the original layer [27]. This model should take four parameters: a set of values for the input features of the weather forecast (solar radiation power, average temperature, wind speed) and the value of the initial feature (the power of electricity generation). Since the original feature is represented as a continuous value, the activation function must return a value in the range $[0, +\infty)$. The selection of this particular activation function was based on the constraint of electricity generation, which can obtain positive and zero values only. The conducted studies have shown that the best forecasting results are obtained by a model with the following added layers:

- Input layer consisting of 3 neurons.
- Dense layer consisting of 128 neurons and having the ReLU activation function.
- Dense layer consisting of 24 neurons.
- Dense (output) layer consisting of one neuron.

The implementation of RNN is represented by recurrent and dense layers. Unlike the FNN, the number of parameters has been increased. Thus, the set of input feature values now includes the de-scaling time, which has been converted to a zero-based integer. The conducted studies of the implementation of this model showed that the best forecasting results are provided by RNN, which has the following layers:

- Input layer consisting of 4 neurons.
- Simple recurrent layer, which includes 200 neurons and has the ReLU activation function.
- Dense layer consisting of 100 neurons.
- Dense (output) layer consisting of one neuron.

The implementation of the neural network with LSTM has a similar set of features as those used for RNN. The conducted studies showed that the best forecasting results are provided by the neural network with LSTM, which includes the following layers:

- Input layer consisting of 4 neurons.

- LSTM layer, which includes 200 neurons and has the ReLU activation function.
- Dense layer consisting of 100 neurons and having the ReLU activation function.
- Dense (output) layer consisting of one neuron.

Also, for forecasting time series, another network of LSTM was implemented, which has only two parameters: date and capacity of electricity generation. Such a model has the following layers:

- Input layer consisting of 2 neurons.
- LSTM layer, which includes 100 neurons and has the ReLU activation function.
- Dense layer consisting of 100 neurons and having the ReLU activation function.
- Dense (output) layer consisting of one neuron.

Each neural network model was trained for 100 epochs using the Keras library written in Python. The input data set was divided into two parts, where the training sample has a share of 80% of the entire set, and the training sample - 20%. The validation set wasn't used because of small dataset. Splitting into three sets may result in insufficient data to train the model effectively.

The sequential model of Keras library was used to implement all the model types, in which the network is represented as layers in a sequential order, and data is transferred from one layer to another in a given order until they reach the original layer.

As a result of the training of neural network models, as well as the application of a test sample to them, graphs of the dependence of accuracy and costs (during training and during validation) of models from epochs were obtained for all types of neural networks. It can be noted that among the investigated models, FNN stands out significantly because of the fact that it has the highest value of accuracy. To increase the accuracy of forecasting, a number of additional experiments using various optimization algorithms were also conducted [26].

As it is known, gradient methods used to optimize neural networks are among the most popular. The used Keras library contains implementations of various algorithms for the optimization of the gradient descent method. Three algorithms (AdaGrad, RMSProp, and ADAM) can be identified that implement different approaches to make gradient descent more efficient in finding optimal weights. However, these algorithms are often used as "black box" optimizers because it is difficult to find practical explanations of their strengths and weaknesses [28]. A comparison of the training results and the work of the created forecasting models is shown in Table 2.

To compare the implemented models, RMSE (Root Mean Square Error) metric was used, which reflects the root mean square deviation [29]. The obtained results indicate that the developed models of RNN and a neural network with LSTM have a rather large forecasting error. Among these models, we can single out the neural network with LSTM with five parameters, since the RMSE value for the test sample is 635 W.

This shows that the neural network with LSTM is better suited for time series forecasting than RNN, however, the accuracy of such models is not satisfactory. This is explained by the fact that the target parameter is based not only on dependent time series, but also on time-independent indicators, that is, weather conditions.

Therefore, to increase the accuracy of forecasting, it is necessary to have a separate set of historical data for electricity generation by solar and wind power plants. Also, another solution can be the accumulation of historical data for several years, which will make it possible to perform time series forecasting depending on the season.

Realized models based on FNN showed good forecasting results. Among the studied models, the one with four parameters and using the ADAM optimizer turned out to be the best. The Z-score method was used to normalize the training sample, as it more accurately preserves significant changes in indicators. The RMSE value for the test sample is 466 W, which is an acceptable indicator. Therefore, it is advisable to use this model to implement a complex intellectual method for increasing the efficiency of using electrical appliances. Among the main advantages, we can also highlight the speed of training and the small file size of the trained model, which will allow the usage of such a model even on low-power computing devices. Since the accuracy of this model is not high enough, to improve its implementation it is necessary to have a larger amount of historical data, at least from one year.

Since the intelligent support subsystem has a mechanism for automatic retraining of the neural network model, the accuracy of power generation forecasting can be increased after a while.

Table 2
Results of training of implemented models

Criterion of comparison	FNN			RNN	Neural network with LSTM	
Total number of layers	4	4	4	4	4	4
Number of parameters	4	4	4	5	5	2
Optimizer	ADAM	RMSprop	AdaGrad	ADAM	ADAM	ADAM
Error on training set (RMSE)	446,9633	425,1278	812,8156	183,9365	499,6775	1066,8383
An error on the test set (RMSE)	466,1486	492,0719	831,0850	672,0999	635,0425	1066,8383
Size of training set	1728	1728	1728	1728	1728	2112
Size of test set	432	432	432	432	432	48
The file size of the trained model	153,9 Kb	153,9 Kb	153,9 Kb	773,52 Kb	344,8 Kb	166,79 Kb
Study time	21 s	22 s	21 s	1 min 20 s	4 min 36 s	18 min 22 s
Number of batches	24	24	24	24	24	24
Number of epochs	100	100	100	100	100	100

3.2 Complex intelligent method

On the basis of the research of popular intelligent methods in the field of organization of optimal electricity consumption, a complex intelligent method of energy consumption control was developed, using FNN and a basic variant of PSO with some improvements.

The algorithm of the complex method is given in the form of the UML activity diagram (Fig. 1) and involves two main stages: forecasting of electricity generation and creating an optimal hourly schedule for the usage of electrical appliances.

In order to implement a complex intelligent method, the neural network is first trained based on the historical data obtained from HAS. They should include the date and time, information on electricity generation, as well as data on weather conditions. After successful training, the neural network is able to predict the generation of electricity. The input data for the operation of the neural network is the hourly weather forecast for the next day, which can be obtained using a weather service (for example, Visual Crossing). Thus, the forecasted generation information will help to estimate the available electricity for usage and to develop a plan for the efficient application of appliances for the next day, making maximum use of alternative sources of electricity.

The next stage of the implementation of the complex method is the creation of an optimal hourly schedule for the usage of electrical appliances using the PSO. The idea is to automatically plan the use of electrical appliances during the next day, considering the habits of residents, during periods when electricity generation is maximum. However, the operating time of the devices is not limited for the comfort of users. The goal of optimization is to create a schedule that will use as much electricity as possible from alternative power sources to ensure the operation of all planned household appliances, and minimize the consumption of electricity from the external power grid.

Thus, the user will be able to save money significantly and be less dependent on electricity suppliers. Input data for the operation of the optimization algorithm are the forecasted data on electricity generation, as well as the list of household appliances that are planned for usage the following day. The description of available devices includes the following data: name, nominal power (W), planned duration of operation. The user can choose from the list of electrical appliances that he plans to use, as well as specify the time intervals of their usage that are convenient for him.

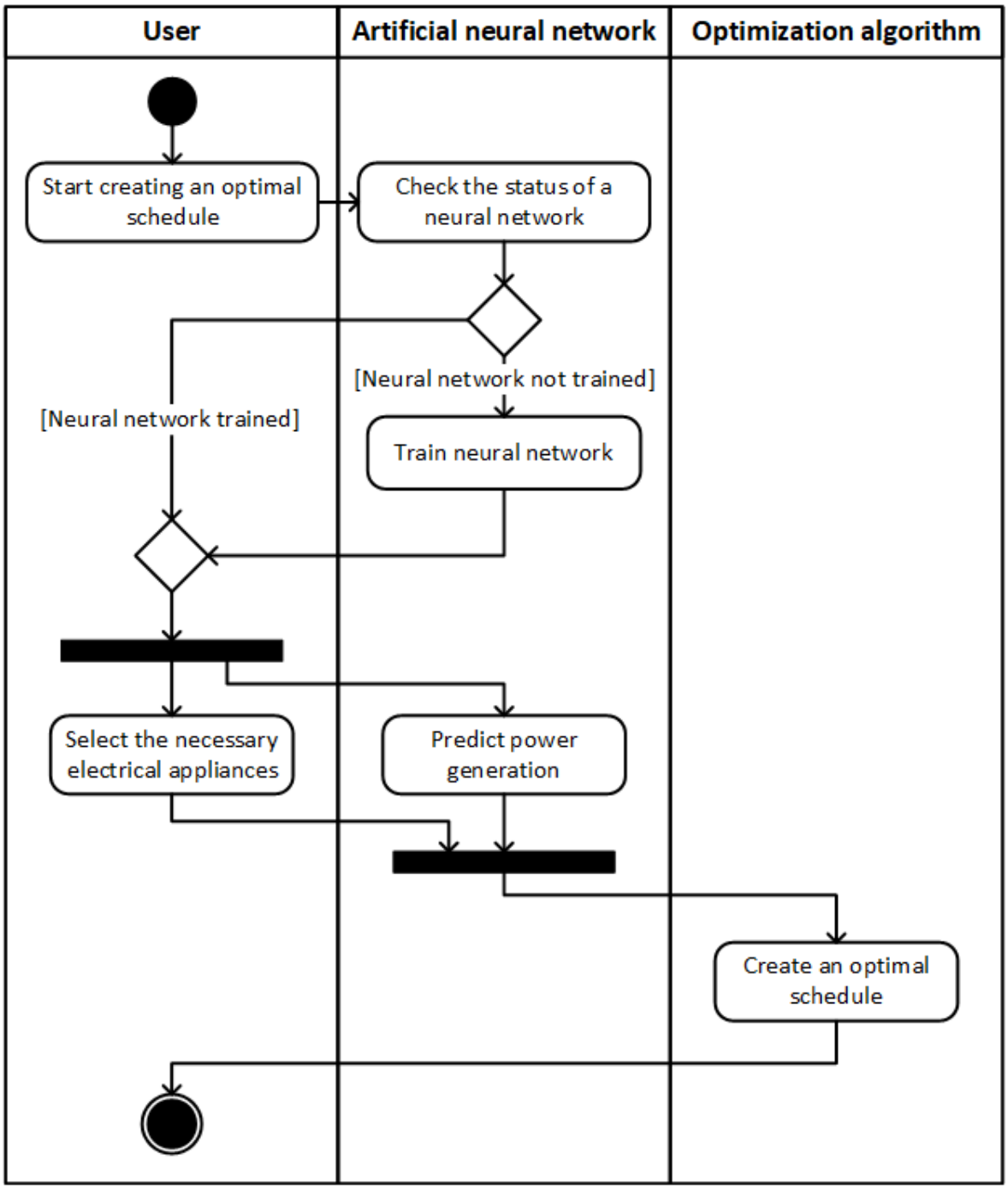


Figure 1: UML diagram of the developed complex intelligent method

Since solar and wind power plants provide the generation of a limited amount of electricity, it was proposed to improve the PSO by introducing a priority system (Fig. 2).

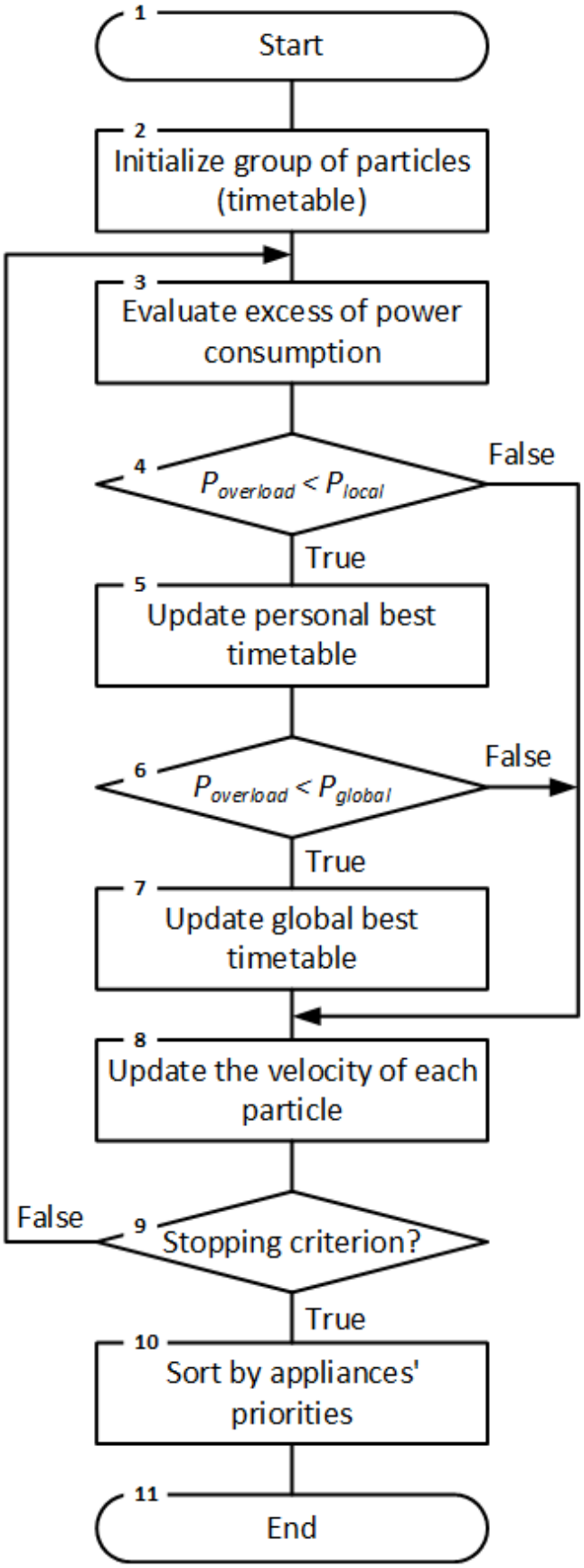


Figure 2: Particle swarm optimization algorithm with priority system

Priority is a number that can take a value from one to five, where five means the highest priority. This will allow the user to choose the most important household appliances for him, and in case of insufficient electricity generation, the algorithm will suggest not to use those with the lowest priority.

The input data of the PSO is the predicted electricity generation for the next 24 hours and a list of household appliances. For each of them the time intervals of use, the number of operation hours and the priority are indicated. The PSO also includes some tuning parameters which influences the performance of this algorithm. These parameters were set as follow: particle numbers m = 100, the max iteration number T_{max} = 10000, cognitive parameter c_1 = 1.49618, social parameter c_2 = 1.49618, inertia weight w = 0.7298. As a result of the work, the optimization algorithm will offer the best schedule for the use of electrical appliances, adjusting to the requirements and habits of the resident, but considering the generation of a certain amount of electricity from alternative sources.

The developed algorithm for monitoring the operation of the power grid provides round-the-clock monitoring of the consumption of electricity generated in the household, as well as supplied from the external power grid, which allows the owner to instantly react to situations and take measures to improve the energy efficiency of HAS.

3.3 Software implementation of the intelligent support subsystem

Based on the developed method and algorithms, an intelligent support subsystem was created, which integrates with the OpenHAB home automation platform and is designed to provide recommendations to the residents of the house on the efficient usage of electrical appliances with the maximum use of energy from alternative sources, as well as with minimal dependence on the external power grid. The developed structural/functional diagram of the subsystem includes the following modules (Fig. 3):

- Module of indicators monitoring, which analyzes energy consumption data in real time and sends a message to the user in case of exceeding electricity consumption.
- Module of electrical appliances editing, that allows the user to add, modify, and delete electrical appliances from the list of HAS.
- Module of schedule optimization, that allows to create an optimal hourly schedule for the operation of household appliances, using the list of electrical appliances that the user plans to use, as well as the forecasted electricity generation.
- Module of forecasting of electricity generation for the next day using the weather forecast downloaded from the weather service and historical energy consumption data stored in the system database. The forecasting module is called when necessary, and the forecasting results are used during the generation of the optimal schedule.

The subsystem is implemented as a web application using the Java programming language and the Spring framework. The electricity generation forecasting module is developed in the Python programming language using the Keras library. All necessary data is stored in the MongoDB database, which is suitable for storing large volumes of information. Storage of historical weather forecast data, as well as forecasted electricity data, is implemented using CSV files.

The web application provides the users with a graphical interface through which they can access and control all subsystem functions and receive important messages using a browser. Web pages for the Spring framework were developed using JSP technology. In the process of creating web pages, the jQuery library was used, which provides a wide range of functions for convenient development. The Bootstrap library was used to implement an adaptive design that will display well on all devices. The FontAwesome library allows to add various icons, which makes the graphical interface more understandable and attractive.

Thus, the developed intelligent support subsystem provides the user with the opportunity to create an optimal schedule of household appliances, adapting to the requirements and habits of residents, using the forecasted electricity generation for the next day.

In addition, the subsystem performs round-the-clock control of energy consumption indicators and, in case of high consumption of electricity from the external power grid, sends a message to the user using a graphical interface.

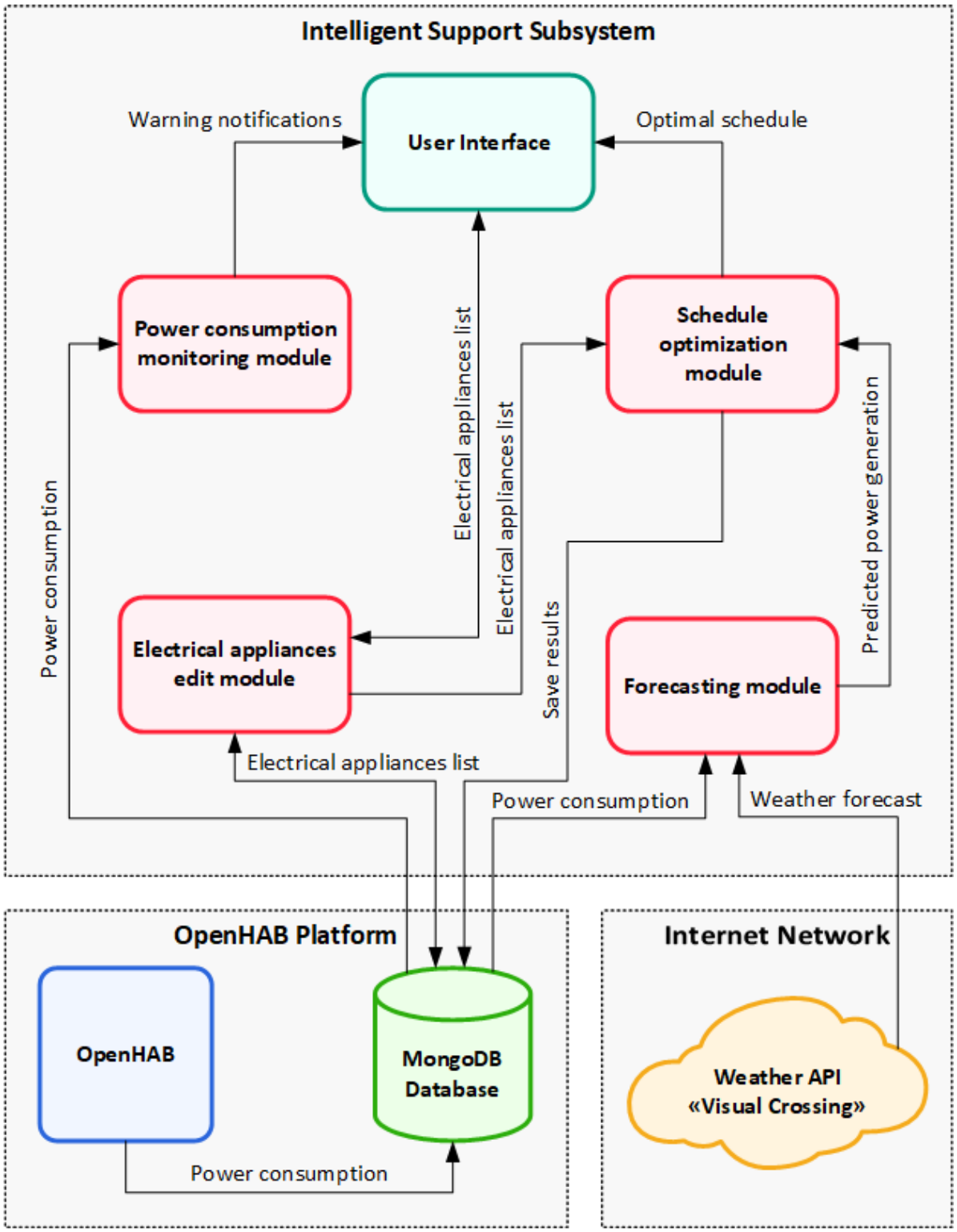


Figure 3: Structural/functional diagram of the developed subsystem

4 Discussion

To date, the developed intelligent support subsystem is at the stage of experimental operation in a real HAS and is integrated with the OpenHAB platform [2].

The studied real HAS is a hybrid system with a hybrid inverter, connection to an external power grid, batteries, as well as alternative energy sources (solar and wind power plants).

Batteries are used as a backup for situations when there is no own generation of electricity, or it is impossible to use an external power grid. The rest of the time they are in a charged state in a buffer mode.

In this operating mode, the analysis of consumption from the external power grid, the values of the charge/discharge currents of batteries and other parameters related to the consumption of electricity is performed.

The main objective of the developed intelligent support subsystem is to minimize consumption from the external power grid, as well as the number of charge / discharge cycles of batteries while maximizing the usage of internally generated energy.

With this mode of operation, there is no deep discharge of the batteries and, accordingly, there is no decrease in the number of working cycles of their work.

However, the system will provide power consumption optimization even in the case when the power consumption from the external power grid is zero, and it is also possible to obtain and analyze the batteries charge/discharge currents.

The improved architecture of the real HAS and its interaction with the integrated intelligent support subsystem is shown in Fig. 4.

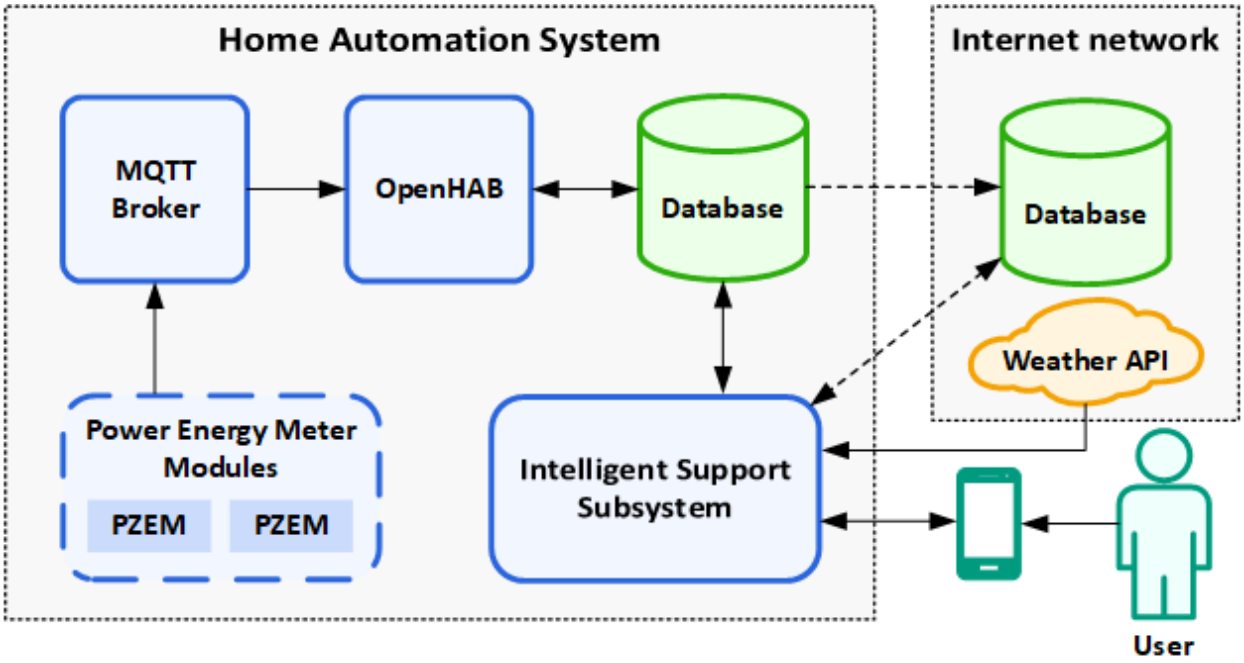


Figure 4: Improved architecture of real HAS

The main component of the investigated real HAS is the OpenHAB platform. Data from the PZEM-004T modules, which are designed to measure input voltage, current load consumption and calculate power consumption in real time, are received by the MQTT broker, which sends them to OpenHAB, which in turn writes them to the database.

The developed intelligent support subsystem interacts with the local database included in HAS and receives all the necessary indicators from the electricity measurement modules, analyzes them in real time, and also uses them for neural network training. The list of the used most powerful electrical appliances is stored in the same database.

The remote database uses a replication mechanism and is an optional element of the architecture, although it is required to perform subsystem performance testing.

Using any device, the user can connect to the intelligent support subsystem using only a browser to create an optimal hourly schedule for using appliances, monitor energy consumption and generation, edit the list of the most powerful electrical appliances, and receive important messages (Fig. 5).

The conducted software testing showed that the developed software functions correctly and meets all the requirements.

Figure 5: The interface of the intelligent support subsystem

5 Conclusion

As a result of the conducted research, a complex intelligent method of controlling energy consumption in HAS was developed based on forecasting electricity generation with maximum use of alternative sources, as well as optimizing the schedule of use of electrical appliances to minimize electricity consumption from the external power grid.

The conducted computer experiments made it possible to choose the best neural network model for forecasting electricity generation, as well as to analyze the factors that affected the accuracy of the forecast. To train neural network models, a set of historical data of electricity consumption, which was accumulated in a real HAS on the OpenHAB platform, was used.

The developed intelligent support subsystem is designed to provide recommendations to the residents of the house on the effective use of electrical appliances in order to minimize energy consumption from the external power grid.

The scientific novelty of the work consists in the development of an intelligent method of energy consumption control, which differs from the existing ones by a comprehensive approach based on forecasting electricity generation using a Feedforward neural network, as well as optimizing the schedule of electrical appliances usage based on the improved PSO with the proposed system of priorities.

The practical significance of the work is that the integration of the developed intelligent support subsystem in HAS will allow controlling the processes of electricity generation and consumption, increasing the efficiency of using household appliances and saving resources.

In future work it's planned to enhance the learning algorithms by combining multiple neural networks that can improve the overall accuracy of power generation prediction to create a more optimized schedule in order to electricity consumption.

In future work, it is planned to improve the learning algorithms by combining several neural networks, that can improve the overall accuracy of electricity generation forecasting to create a more optimized schedule to maximize electricity saving. Also, in the future development of this work, it is

planned to consider more promising architectures of neural networks for solving problems of time series forecasting.

6 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).

7 References

[1] C. Bruhathireddy, G. N. Kodandaramaiah, M. Lakshmipathy, Design and implementation of home automation system using Raspberry Pi, International journal of science, technology & management 03(12) (2014) 94-98.

[2] M. Zadoian, Y. Horichenko, A. Tulenkov, A. Parkhomenko, Data mining to achieve quality of life for home automation users, in: Proceedings of 11th IEEE International conference on Intelligent data acquisition and advanced computing systems: technology and applications, Cracow, Poland, 2021, pp.55-59. doi:10.1109/IDAACS53288.2021.9660836.

[3] A. Tulenkov, Y. Yaremchenko, A. Parkhomenko, Ya. Zalyubovskiy, A. Parkhomenko, M. Kalinina, Adaptation of Smart House system for people with special needs based on wireless technologies, in: Proceedings of 5th IEEE International symposium on smart and wireless systems within the International conference on Intelligent data acquisition and advanced computing systems, Dortmund, Germany, 2020, pp.12-17. doi:10.1109/IDAACS-SWS50031.2020.9297072.

[4] A. Arhipov, A. Tulenkov, A. Parkhomenko, Ya. Zalyubovskiy, A. Parkhomenko, Remote monitoring of electrical equipment for Smart House system safe exploitation, in: Proceedings of 5th IEEE International symposium on smart and wireless systems within the International conference on Intelligent data acquisition and advanced computing systems, Dortmund, Germany, 2020, pp.260-265. doi:10.1109/IDAACS-SWS50031.2020.9297103.

[5] A. Parkhomenko, A. Tulenkov, A. Sokolyanskii, Y. Zalyubovskiy, A. Parkhomenko, A. Stepanenko, The application of the remote lab for studying the issues of Smart House systems power efficiency, safety and cybersecurity, in: Smart Industry & Smart Education, volume 47 of Lecture Notes in Networks and Systems, Springer, Cham, 2018, pp. 395-403. doi:10.1007/978-3-319-95678-7_44.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] F. Iqbal, K. Iqbal, Optimal load scheduling using genetic algorithm to improve the load profile, ADRRI Journal of engineering and technology 4(9(3)) (2021) 1-15. doi:10.55058/adrrijet.v4i9(3).621.
[12] 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, India, 2016, pp. 361-365.
[13] 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.
[14] P. B. Gopikrishna, J. A. Mathew, Power consumption analysis and prediction of a Smart Home using ARIMA model, Rochester, SSRN, 2021, 9 p. doi:10.2139/ssrn.3819512.
[15] 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.
[16] 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.
[17] 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.
[18] 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.
[19] O. D. Diaz-Castillo, A. E. Puerto-Lara, J. A. Saenz-Leguizamon, V. Ducon-Sosa, Prediction of electrical energy consumption through recurrent neural networks, in: Proceedings of the workshops at the 4th international conference on Applied informatics, Buenos Aires, Argentina, 2021, 174-182.
[20] 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.
[21] 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.
[22] V. Teslyuk, A. Kazarian, N. Kryvinska, I. Tsmots, Optimal artificial neural network type selection method for usage in Smart House systems, Sensors 21(1):47 (2021) 1-14. doi:10.3390/s21010047.
[23] F. D. Wihartiko, H. Wijayanti, F. Virgantari, Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem, IOP conference series: Materials science and engineering 332:012020 (2018) 1-6. doi:10.1088/1757-899X/332/1/012020.
[24] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the International conference on Neural networks, Perth, Australia, 1995, pp. 1942-1948. doi: 10.1109/ICNN.1995.488968.
[25] C. Li, D. C. Coster, Improved particle swarm optimization algorithms for optimal designs with various decision criteria, Mathematics 10 (13) (2022) 1-16. doi:10.3390/math10132310.
[26] S. O. Subbotin, Neural networks: theory and practice, Zhytomyr, 2020, 184 p.
[27] Keras – Models, 2023. URL: https://tutorialspoint.com/keras/keras_models.htm/.
[28] A. Oppermann, Optimization in deep learning: AdaGrad, RMSProp, ADAM, 2023. URL: https://artemoppermann.com/optimization-in-deep-learning-adagrad-rmsprop-adam/.
[29] Root-mean-square error in R programming, 2020. URL: https://geeksforgeeks.org/root-mean-square-error-in-r-programming/.

A study of temporal and recurrent neural networks for CO2 emission forecasting

Oleg Rudenko[a], Oleksandr Bezsonov[a], Nataliia Serdiuk[a] and Kateryna Pasichnyk[a]

[a] *Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine*

Abstract

Time series forecasting is a rapidly growing field of research and provides many opportunities for future work not only in the field of forecasting, but also in many other areas of life. A large number of interesting and noteworthy projects and technologies have already been made in this direction, and various methods of combining forecasting models with ANNs have been proposed in the literature. The advantage of the ANN methods proposed in this paper is that they provide a methodology to approximate real data, i.e., the estimation of the weight vector does not depend on any model. This frees one from model-based selection procedures and sample data assumptions. When nonlinear systems are still in a state of development, it is possible to conclude that the ANN approach offers a competitive and reliable method for system analysis, prediction, and control.According to the Scripps Institution of Oceanography, last year the concentration of carbon dioxide in the atmosphere reached a maximum level of 416.43 ppm (parts per million) for the first time in human history. Every year, this number grows. Therefore, monitoring and forecasting of this indicator is necessary for timely mitigation of the consequences of climate change and ensuring the global climate security of mankind. This paper proposes forecasting the level of carbon dioxide in the atmosphere based on artificial neural networks, namely: recurrent neural network (RNN) and temporal convolutional neural network (TCN).

Keywords [1]

Carbon dioxide concentrations, basic recurrent architectures, temporal convolutional neural networks, app Jupyter Notebook

1. Introduction

Climate change is undoubtedly one of the most serious problems facing humanity, which is why experts consider it an existential threat to our species, that is, one that can lead to irreversible consequences. According to climate data from open sources, the increase in the frequency of extreme weather events is associated with climate change, and decisive action is needed worldwide to address this problem. Otherwise, millions of people will suffer, and their quality of life will decrease significantly in the years to come.

Greenhouse gases such as carbon dioxide (CO2) and methane (CH4) trap heat in the atmosphere, thereby keeping our planet warm and friendly to biological species. Despite this, human activities, such as fossil fuel burning and energy generation, emit huge amounts of greenhouse gases, resulting in excessive increases in the Earth's average global temperature. According to the Scripps Oceanographic Institute [1], last year the concentration of carbon dioxide in the atmosphere for the first time in human history reached the maximum level - 416.43 ppm (parts per million). This number,

[1]The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: oleh.rudenko@nure.ua (O. Rudenko); oleksandr.bezsonov@nure.ua (O. Bezsonov); nataliya.serdyuk@nure.ua (N. Serdiuk); kateryna.pasichnyk1@nure.ua (K. Pasichnyk);
ORCID: 0000-0003-0859-2015 (O. Rudenko); 0000-0001-6104-4275 (O. Bezsonov); 0000-0002-0107-4365 (N. Serdiuk); 0000-0001-9228-2380 (K. Pasichnyk)

unfortunately, increases every year. Therefore, control and forecasting of this indicator is necessary for timely mitigation of the consequences of climate change and ensuring the global climate security of mankind.

Considering the urgency of the problem, it was decided to make a comparative prediction of the level of carbon dioxide in the atmosphere for 2023 based on artificial neural networks (ANN) [2], namely: recurrent neural network (RNN) and temporal convolutional neural network (TCN). After that, an analysis was carried out on the subject of which neural network forecasting method from the selected ones showed the best results according to the criteria of forecast accuracy [3, 4].

The forecast dataset was taken from an open source dataset from the Mauna Loa Observatory (MLO), a volcano in Hawaii. It houses a research center that has been monitoring the atmosphere since 1950, and its remote location provides ideal conditions for recording climate data. In 1958, Charles David Keeling organized a CO2 monitoring program and began recording scientific evidence of rapidly increasing CO2 concentrations in the atmosphere. The CO2 dataset was uploaded to Mauna Loa from the Scripps Institution of Oceanography and includes monthly atmospheric CO2 concentrations in parts per million (ppm) from 1958 to 2022.

2. Building forecasting models in the Jupyter Notebook environment

Software implementation, training of neural networks, obtaining graphical and numerical results will be performed in the Jupyter Notebook application.

To begin with, we import a set of libraries required for the project, namely: Pandas [5], Mathplotlib, as well as various functions from the darts and statsmodels library [6].

After importing the libraries, you need to load the selected CO2 data sample into a Pandas data frame (DataFrame), and be sure to apply data preprocessing methods to it, such as removing empty lines, setting a monthly date and time index, and removing null values.

After preprocessing the data set, we use the plot() function to create a simple line graph to look for a trend or seasonal pattern.

Figure 1 shows a graph of a dataset of dates and levels of atmospheric carbon dioxide from March 1958 to the end of 2022.

As can be seen from the graph in Figure 1, there is a clear upward trend, which highlights the fact that the concentration of CO2 in the atmosphere has been increasing rapidly in recent decades. In addition, there is also seasonality due to the planet's natural carbon cycle, as plants absorb and release CO2 in different seasons of the year. In particular, when plants begin to grow in the spring, they clean the atmosphere of CO2 through photosynthesis. Conversely, when trees lose their leaves in the fall, CO2 concentrations increase through respiration.

In this particular case, we have already identified the components (seasonality, trend) by qualitatively analyzing the linear graph (Figure 1), but the seasonal decomposition (Figure 2) helps a lot in more complex cases.

In the next step, the plot_acf() function of the statsmodels library was used to construct the autocorrelation function of the time series, linear dependencies between its lag values were approximated. Autocorrelation is the correlation of the levels of deviations from the trend with each other, that is, the correlation within the same time series, but with different shifts in time. The autocorrelation of time series levels, if it is significant, indicates the presence of a trend, that is, it serves as one of the methods of trend detection. Programmatically, this is described as follows:

```
fig, ax = plt.subplots(figsize = (8,5))
plot_acf(df, ax = ax)                                    (1)
plt.show()
```

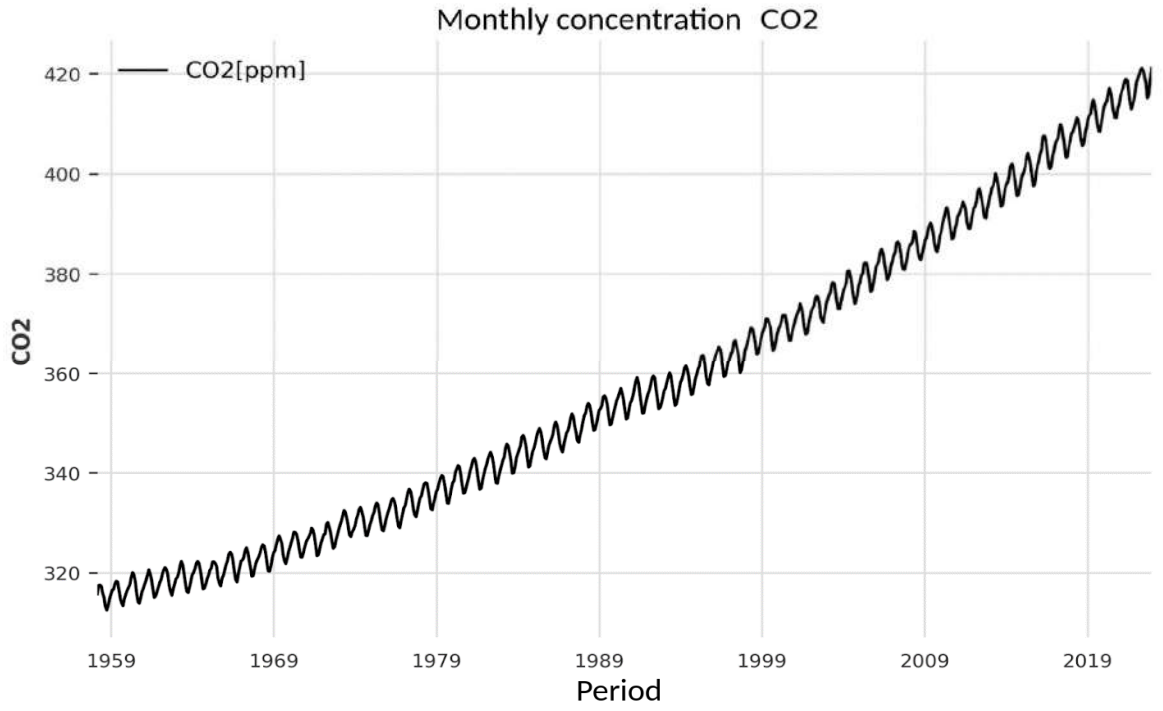


Figure 1: CO2 concentration in the atmosphere since 1958

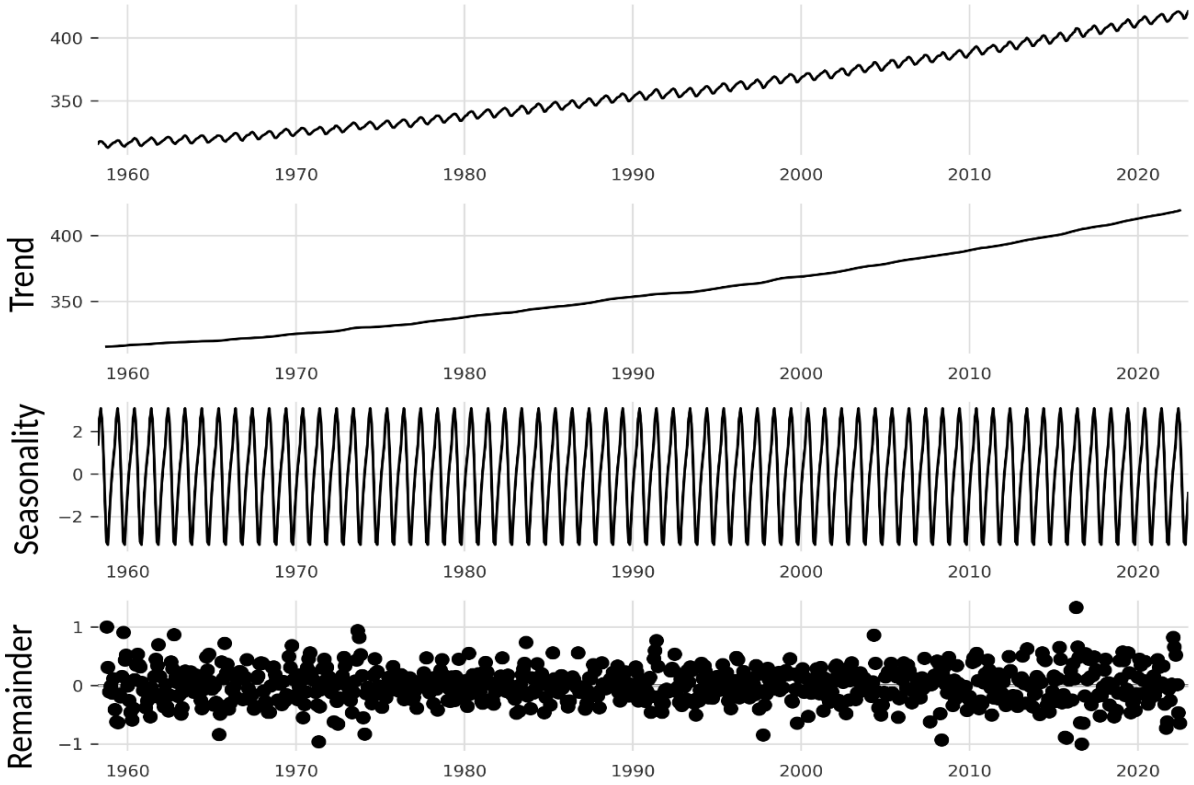


Figure 2: Trend, seasonality and residuals based on the data

The autocorrelation technique consists of sequential calculation of the autocorrelation coefficients of deviations with different shifts in time. Figure 3 shows the result of the autocorrelation function.

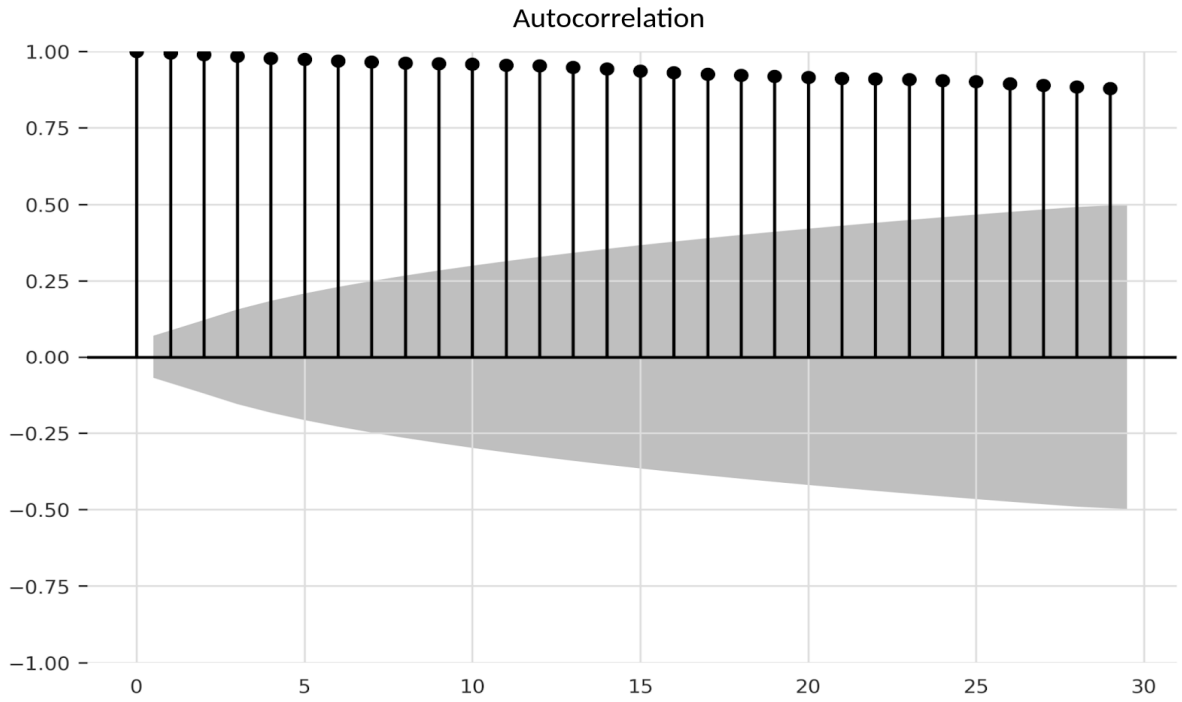


Figure 3: Autocorrelation function

From the trend of the time series, we can see that the autocorrelation is high for small lags and starts to gradually decrease after lag 5. The autocorrelation function should also pick out the seasonal component of the time series, but in our case the seasonal component is not noticeable.

3. Time series forecasting

At this stage, training of selected neural network forecasting models [7] was carried out on the uploaded data set on the level of CO2 in the atmosphere. After the obtained results, a comparison of their effectiveness was performed in order to choose the most accurate model [8, 9] and create a forecast for 2023 based on it.

A number of criteria are used to assess the accuracy of the model. Let's briefly consider each of them:

- The mean absolute error (Mean Absolute Error, MAE) is calculated as the mean absolute difference between the values of the model setting (one step ahead in the example forecast) and the observed historical data
- Mean Squared Error (RMSE) is the square root of the MSE metric. It refers to the same scale as the observed data values. Any small deviation can significantly affect the error rate
- The average percentage of errors (Mean Absolute Percent Error, MAPE) is the average absolute difference in percentages between the values of the model setting and the values of the observed data and the symmetrical average percentage of errors (English: Symmetric Mean Absolute Percent Error, SMAPE)
- Coefficient of determination. In fact, it is a measure of the quality of the model - it is the normalized root mean square error. If it is close to one, then the model predicts the data well, if it is close to zero, then the quality of forecasts can be compared with linear forecasting [8]

Let's start the setup process by loading a pandas data frame into a TimeSeries object as required by the Darts library:

```
series = TimeSeries.from_dataframe(df)
start = pd.Timestamp('123115')
df_metrics = pd.DataFrame()
```
(2)

Additional functions plot_backtest() and print_metrics() will allow you to plot predictions and display model accuracy metrics, namely MAE, RMSE, MAPE, SMAPE, and R^2.

3.1. Creation of a temporal convolutional network forecasting model

We will consider prediction models based on neural networks, namely, we will create an algorithm for learning a recurrent neural network and a temporal convolutional neural network. For this task, the Darts library is used, which contains standard tools for working with selected neural networks. Convolutional neural networks are a promising field in machine and deep learning [10], and the TCN model was created for even better data analysis, learning and prediction [11]. Recurrent neural networks have become the standard choice in research [12,14] as well as in practical applications [13]. Despite this, the temporal convolutional neural model is an alternative architecture that gives promising results, so its performance will now be tested in practice [15].

First of all, let's set the parameters of the TCN neural network for training:

```
  model = TCNModel(
....input_chunk_length=24,
....output_chunk_length=12,
....n_epochs=100,
....dropout=0.1,
....dilation_base=3,
....weight_norm=True,
....kernel_size=5,
....num_filters=3,
....random_state=0,
)
```
(3)

The history_forecasts() function and other service functions are used to test the temporal convolutional network model and display the results. In addition, time series are normalized using the Scaler() class.

The program code describes the given parameters and functions for training and outputting graphical results.

After a series of settings and manipulations with network parameters, training and adjustment of results, we get a graph with a forecast based on the TCN model (Figure 4).

According to the results of the accuracy criteria of this model (Table 1), this model can best predict the level of carbon dioxide in the atmosphere for the coming years. It was possible to achieve the ideal MAPE and SMAPE of 0.1%.

Table 1

Accuracy criteria of the TCN neural network

MAE	RMSE	MAPE	SMAPE	R^2
0.41	0.63	0.1	0.1	0.96

It cannot be argued that the use of neural networks for forecasting tasks is always the best choice, because everything depends on the task and the goal of the research. Each of the methods is successful in its own way. For comparison, we will also train a recurrent neural network.

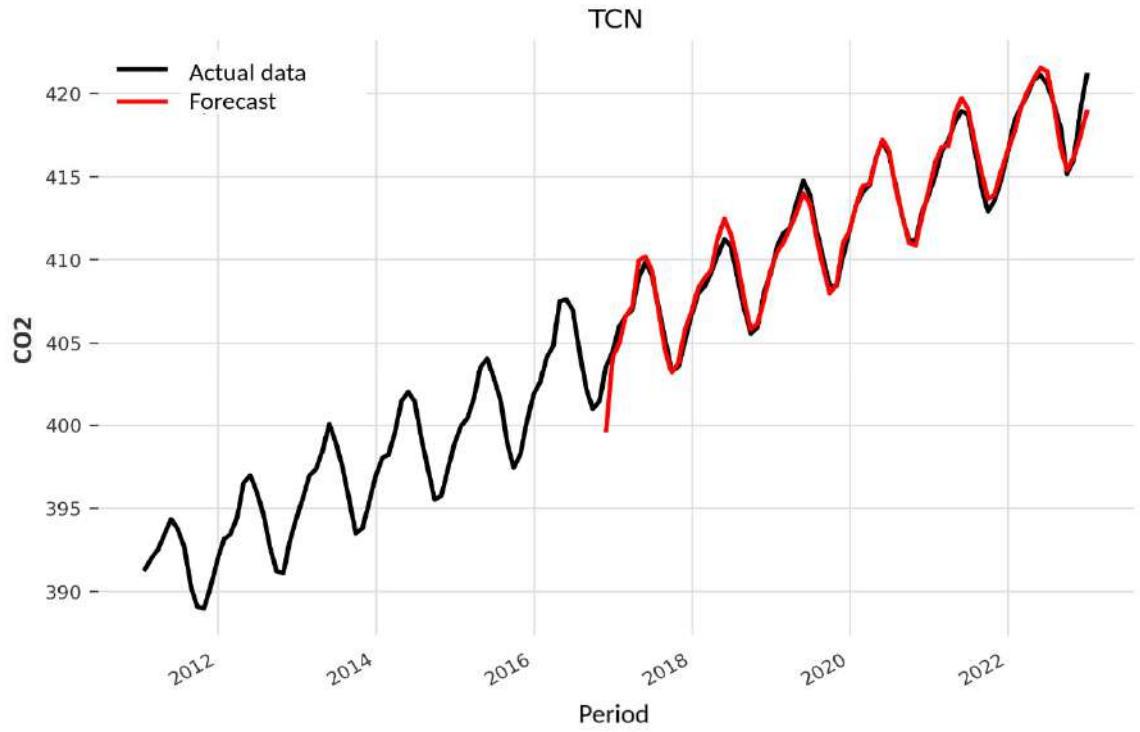


Figure 4: Prediction graph based on TCN network

3.2. Creation of a prediction model based on a recurrent neural network

Modern networks in which connections between nodes form a certain cycle are called backpropagation or recurrent networks. Because of this internal state, the network can exhibit dynamic behavior over time. In recurrent networks, communication between neurons can come not only from the lower layer to the upper one, but also from the neuron to "itself", more precisely, to the previous value of this neuron or other neurons of the same layer. This allows to display the dependence of the variable on its eigenvalues at different points in time [14].

Recurrent neural networks are often compared to temporal convolutional neural networks due to the fact that both are able to learn effectively and are actively used in various tasks, not only in prediction, but also in other areas of deep learning.

The learning process is similar to TCN. Without special skills and programming skills, it is possible to get a forecast thanks to machine learning libraries, in our case it is Darts.

Let's set the parameters for training the RNN network:

```
model = RNNModel(
    model="RNN",
    hidden_dim=40,
    dropout=0,
    batch_size=24,
    n_epochs=50,
    optimizer_kwargs={"lr": 1e-3},
    log_tensorboard=True,
    random_state=40,
    training_length=60,
    input_chunk_length=12,
    force_reset=True,
    save_checkpoints=True,
)
```

(4)

The process of selecting the best input parameters of the recurrent neural network took place at the expense of theoretical skills, as well as at the expense of repeated testing of network training on certain parameters. Sometimes the process of learning the network took up to an hour and the graph showed unsuccessful attempts to learn the network, but after optimizing the parameters, the result was ready in just 5-10 minutes. Below is the code with the functions for the RNN network.

```
  model_name = 'RNN'
  plt.figure(figsize = (8, 5))
  scaler = Scaler()
  scaled_series = scaler.fit_transform(series)
  forecast = model.historical_forecasts(scaled_series, start=
start,
  forecast_horizon=12, verbos=True)
  plot_backtest(series, scaler.inverse_transform(forecast), m
odel_name)
  df_dl = print_metrics(series, scaler.inverse_transform(fore
cast), model_name)
  df_metrics = df_metrics.append(df_dl)
  plt.show()
  df_dl
```

(5)

The process of learning this network was not easy and required a lot of CPU power. Figure 5 shows the final result of training the RNN network on input data about the CO2 level in the atmosphere.

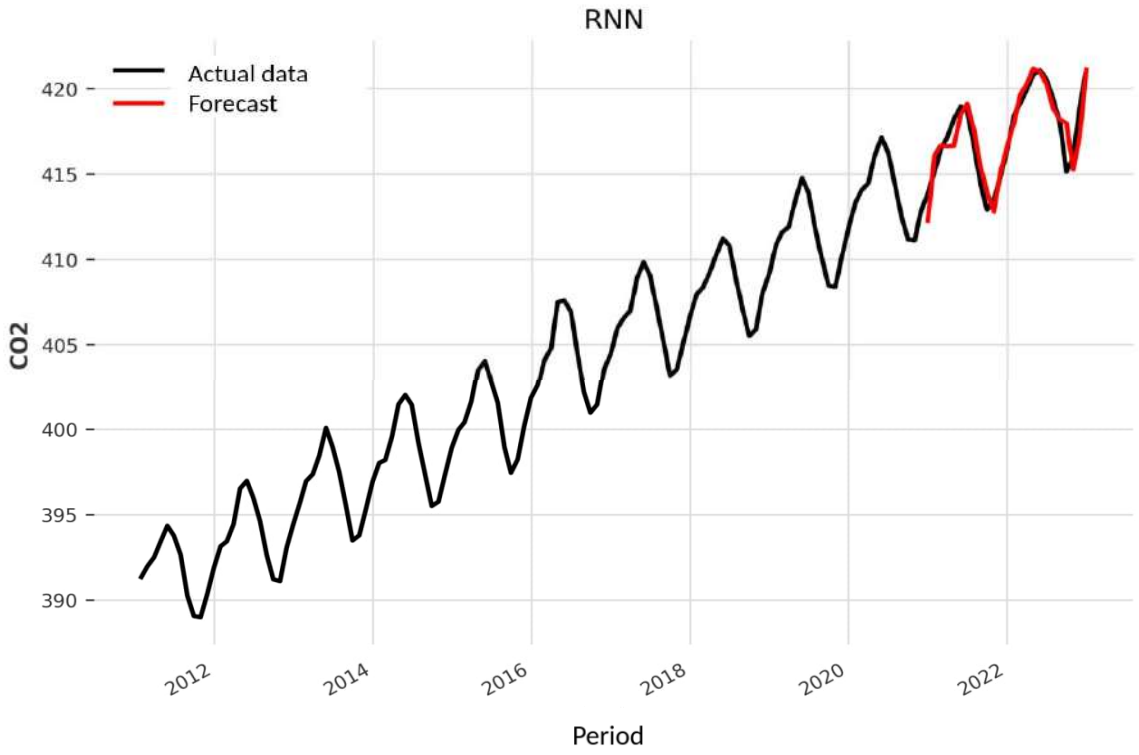


Figure 5: Prediction graph based on RNN network

As for the accuracy criteria of the model (Table 2), they differ significantly from TCN.

It can be seen from Table 2 that recurrent neural networks should not be used for forecasting in this forecasting problem, since the symmetric average relative error indicator is 0.17%, and the

coefficient of determination is almost a tenth worse compared to TCN and the exponential smoothing method.

Table 2
Accuracy criteria of a recurrent neural network

MAE	RMSE	MAPE	SMAPE	R^2
0.72	0.96	0.17	0.17	0.84

4. Comparative analysis of the obtained research results

In order to visually present the effectiveness of the researched methods [16] for solving the problem, the results of the accuracy criteria of forecasts for each of them were collected and a comparative analysis was carried out.

Summary Table 3 contains the metrics of forecast accuracy criteria that were used in the work, namely: MAE, RMSE, MAPE, SMAPE and, according to each of the forecasting models: two models based on ANN.

Table 3
Metrics of forecast accuracy criteria for each of the studied forecasting methods

Forecasting method	Forecast accuracy criteria				
	MAE	RMSE	MAPE	SMAPE	R^2
RNN	0.72	0.96	0.17	0.17	0.84
TCN	0.41	0.63	0.1	0.1	0.96

The high efficiency of the TCN model is due to its ability to accept a sequence of any length and output it as a sequence of the same length as the input, as well as preventing information leakage from the future to the past due to the use of causal convolutions.

Recurrent neural networks are famous for their effectiveness in unsegmented handwriting and speech recognition tasks. They are often compared to temporal convolutional neural networks, but usually the latter have more advantages over the former.

5. Creation of a forecast of the level of carbon dioxide for 2023

After comparing all the results of forecasting methods used in this work, our forecast of CO2 concentration in the atmosphere for 2023 will be based on the temporal convolutional neural network, because it gave the best results.

Input parameters for creating a forecast for 2023 are copied from the TCN model. In order to fit the model and then display the results, the fit() function is used:

```
  model_name = 'Forecast'
  plt.figure(figsize = (8, 5))
  scaler = Scaler()
  scaled_series = scaler.fit_transform(series)
  model.fit(scaled_series)                                          (6)
  forecast = model.predict(12)
  plot_backtest(series, scaler.inverse_transform(forecast),
model_name)
  print(forecast.pd_dataframe())
```

Figure 6 shows the resulting graph with a forecast for the next year based on a temporal convolutional neural network.

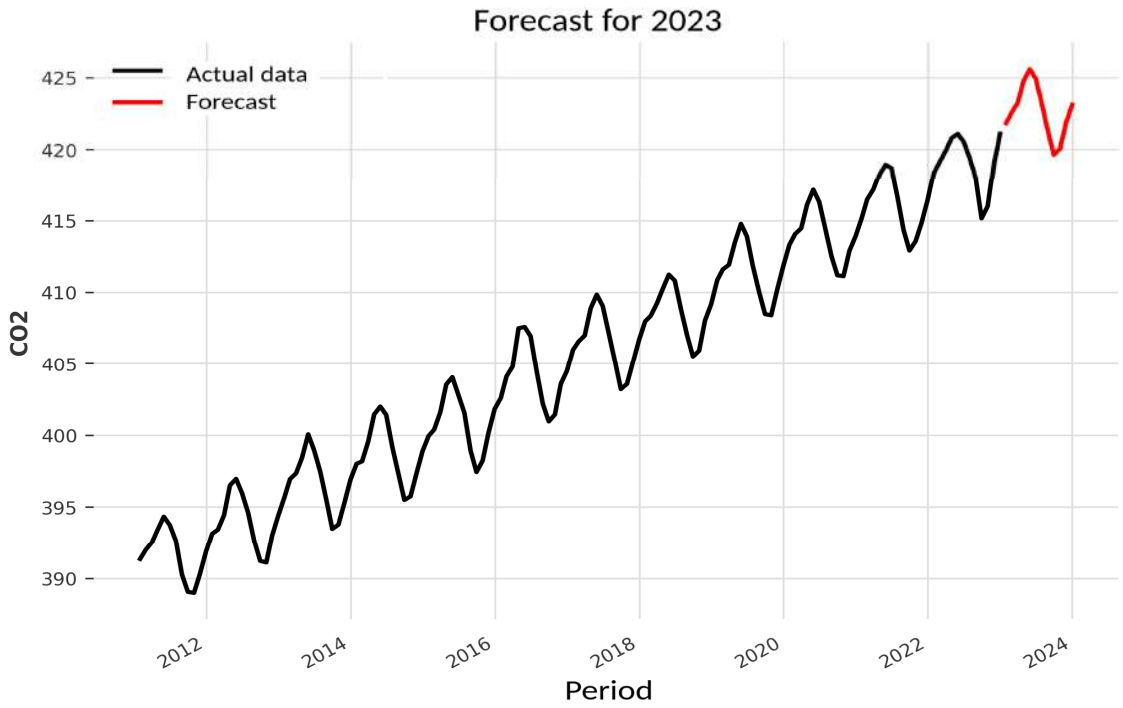


Figure 6: The result of the forecast of the level of CO2 in the atmosphere for 2023

It can be visually observed that the components of the time series were successfully determined by the model. Table 4 contains the numerical indicators of the level of CO2 in parts per million corresponding to each month of 2023.

Table 4
Forecast for 2023 created by the TCN model

Date in year-month-day format	CO2 level, (ppm)
2023-01-31	421.836774
2023-02-28	422.576312
2023-03-31	423.262611
2023-04-30	424.778723
2023-05-31	425.581957
2023-06-30	424.922701
2023-07-31	423.185616
2023-08-31	421.267827
2023-09-30	419.635597
2023-10-31	420.052928
2023-11-30	421.851032
2023-12-31	423.111864

Unfortunately, the level of carbon dioxide is constantly increasing, which is a reason to worry and take decisive action to save the environment from the consequences. With the help of a customized forecasting model based on a temporal convolutional neural network, forecasts can be made several years ahead. Undoubtedly, humanity must develop a strategy to reduce CO2 levels in the atmosphere in order to prevent further climate change, a negative impact on all living organisms and the planet as a whole.

6. Conclusions

It should be noted that a perfect forecast is impossible due to the presence of a large number of factors that are difficult to estimate with a high percentage of accuracy. Therefore, instead of searching for an ideal forecast, it is much more important to have good knowledge in the area of existing models and their correct application depending on the specifics of the data and subject area, the ability to adapt to non-ideal forecasts.

Due to the fact that the forecast depends on past data, its reliability and accuracy will decrease in proportion to how far into the future the task of calculating it is. It is worth noting that the accuracy of the forecast and the costs of its implementation are interrelated. The best forecasts are not necessarily the most accurate. Factors such as its purpose and data availability play an important role in determining the desired level of accuracy.

The dynamic behavior of most time series in our real life, with its autoregressive and legacy moving average terms, makes it difficult to forecast nonlinear time series that contain legacy average terms using computational intelligence methodologies such as neural networks. This model method allows you to determine specific models at low computational costs.

The results show that temporal convolutional neural networks convincingly outperform basic recurrent architectures in a wide range of sequence modeling tasks.

At the moment, new approaches to forecasting time series are emerging, the purpose of which is to overcome the problems of assessing the state of the market, dimensionality of models, identifying signs of higher-level systems, climate, etc. These approaches are based on the application of such branches of modern mathematics as: evolution, the theory of stochastic modeling, the theory of catastrophes and the theory of self-organizing systems, including genetic algorithms and fuzzy logic.

This work can be useful in the future for solving the problems of excessive amount of carbon dioxide in the ambient atmosphere and the prediction model based on TCN can be optimized due to better setting of hyperparameters. This method is more complex and takes more time, but can show significant improvements in prediction results.

7. References

[1] NOAA: Global Monitoring Laboratory, 2022. URL: https://gml.noaa.gov/.

[2] Rudenko O. G., Bodyanskyi E. V. Artificial neural networks: a study guide. Kharkiv: SMIT Company LLC, 2006. 404 p.

[3] Ratnadip Adhikari, Agrawal R. K. An Introductory Study on Time Series Modeling and Forecasting. LAP Lambert Academic Publishing, 2013. 76 p.

[4] Lara-Benítez P., Carranza-García M., Luna-Romera J. M., Riquelme J. C. Temporal Convolutional Networks Applied to Energy-Related Time Series Forecasting. Applied Sciences. 2020. Vol. 10, no. 7, p. 2322.

[5] Pandas: User guide, 2019. URL: https://pandas.pydata.org/docs/user_guide/

[6] Darts: Time Series Made Easy in Python, 2022. URL: https://unit8co.github.io/darts/

[7] Song G., Li H., Witt S.F. Recent developments in econometric modeling and forecasting. Journal of Travel Research. 2005. Vol. 44, 82–99.

[8] Archer, B. H. Demand forecasting and estimation. In J. R. B. Ritchie, & C. R. Goeldner (Eds.). New York: Wiley. 1987. 75–85.

[9] Bodyanskiy Ye., Kokshenev I., Kolodyazhniy V. An adaptive learning algorithm for a neo-fuzzy neuron // Proc. 3rd Int. Conf. of European Union Soc. for Fuzzy Logic and Technology (EUSFLAT 2003). – Zittau, Germany, 2003. – P. 375–379.

[10] L.J. Cao, Francis E.H. Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting. IEEE Transaction on Neural Networks. 2003. Vol. 14, 1506-1518.

[11] C. Müller i S. Guido, Introduction to machine learning with Python: a guide for data scientists. 1st ed. Sebastopol: CA: O'Reilly Media, Inc, 2016, 398 p.

[12] O. Rudenko, O. Bezsonov, i O. Romanyk, Neural network time series prediction based on multilayer perceptron. Development Management. 2019. Vol. 17, vol. 1. P. 23–34.

[13] How to Build a Tensorflow Training Pipeline. Medium: Towards Data Science, 2022. URL: https://medium.com/towards-data-science/how-to-quickly-build-a-tensorflow-training-pipeline-15e9ae4d78a0
[14] Bai S., Kolter J. Z., Koltun V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv, Arp. 19, 2018.
[15] Smith J. E. A Temporal Neural Network Architecture for Online Learning. arXiv, Feb. 22, 2021.
[16] T. Rashid, Make Your own neural network. s.l.: CreateSpace Independent Publishing Platform, 2016. 274 p.

Intellectualization Method and Model of Complex Technical System's Failures Risk Estimation and Prediction

Vladimir Vychuzhanin [a], Natalia Shibaeva [a], Alexey Vychuzhanin [a], Nickolay Rudnichenko [a]

[a] *Odessa Polytechnic National University, Shevchenko Avenue 1, Odessa, 65001, Ukraine*

Abstract

The article describes the developed intellectualization method and model of the complex system failures risk technical condition assessment and prediction by diagnostic features. The developed model has a relative insensitivity to incomplete data about the system and is considered as a conceptual one for an intelligent system for assessing and predicting the complex technical systems failures risk on network infrastructures. Developed method and model can take into account the hierarchical levels of subsystems (components), intersystem (interelement) links when searching for the failures causes. Proposed method and model allow us to control the risk of failures in complex technical systems when information about failures in their structures is received. In addition, the application of the method and model makes it possible to predict trends in the risk of system failures, taking into account changes in the risk of failures, in order to further choose a strategy for their recovery.

Keywords

Complex technical system, diagnostics, forecasting, model, intelligent system, Bayesian belief network, performance, failure risk assessment.

1. Introduction

The diversity of the composition and the increase in the number of complex technical systems (CTS), for example, installed on ships, is accompanied by an increase in the failure rate of such systems. This leads to the need to repair CTS equipment, which means to ship downtime, so probably losses associated with the time spent on repairs and the resources used might be increased significantly. The use of intelligent systems for assessing and predicting the technical condition (TC) of complex systems can significantly extend the life cycle of CTS, including ships [1]. Estimation, prediction of the risk of CTS failures involves the development of methods and technologies of artificial intelligence in order to identify and prevent possible failures in the operation of CTS [2]. This article is devoted to this problem solution.

2. Description of Problem

The operational reliability of recoverable CTSs is effectively achieved by the strategy of operating systems with TC control based on technical diagnostic systems [3,4]. When designing, manufacturing and operating CTS for various purposes, the following hierarchical structure of the CTS diagnostic model is used system (consisting of subsystems); subsystem (functional devices consisting of components); component (concentrated structure containing elements); intercomponent communication or links (distributed structure containing links between components).

CMIS-2023: The Six International workshop on Computer Modeling and Intelligent Systems, May 12, 2022, Zaporizhzhia, Ukraine.
EMAIL: v.v.vychuzhanin@op.edu.ua (V.Vychuzhanin); nati.sh@gmail.com (N. Shibaeva); vychuzhanin.o.v@op.edu.ua (A. Vychuzhanin); nickolay.rud@gmail.com (N.Rudnichenko)
ORCID: 0000-0002-6302-1832 (V.Vychuzhanin); 0000-0002-7869-9953 (N. Shibaeva); 0000-0001-8779-2503 (A. Vychuzhanin); 0000-0002-7343-8076 (N.Rudnichenko)

Subsystems and components are referred to as functional elements (FE). Intersystem and intercomponent links form a group of functional inner-connections (FI) or links.

In design, manufacture and operation, the reliability of CTS is ensured by methods and means specific to each stage of the systems "life cycle". Reliability of ship CTS is estimated based on the results of FE and FI TC diagnosing and can be assessed in the failures risk form [5,6,7].

During CTS operation, prediction of their TC based on diagnostics helps to reduce the risk of system failures [8]. At the same time, the CTS shipboard FE and FI failures risk assessment should take into account their structure (hierarchy and topology of systems), functional states (operability, failure) of their subsystems (components), intersystem (intercomponent) connections, as well as systems data incompleteness.

Studies of ship CTS (energy, electric power, etc.) reliability models show us that the defeat of any subsystems, components in systems generates a significant number of possible scenarios and options for the development of such systems emergency states, as a result, it leads to possible marine accidents.

Available statistics of marine accidents and incidents related to CTS failures are reflected in the well-known databases Global Integrated Shipping Information System - Marine Casualties and Incidents [9], Marine Accident Investigation Branch reports [10], Marine Accident Reporting Scheme reports [11], National Transportation Safety Board NTSB [12], Casualty and Events [13]. Moreover, according to statistics, one of the main CTS - ship power plant (SPP) accounts for 60-80% of all failures of ship systems [14].

Improving the strategy for the operation and maintenance of ship equipment can be achieved by solving the problems of choosing: a rational strategy for servicing CTS equipment; the level of diagnosis and structural parameters characterizing its TC; diagnostic methods and diagnostic parameters; algorithms for extracting diagnostic information; forecasting methods.

Currently, the volume of implementation of automation, digitalization and artificial intelligence (AI) technologies in various industries continues to grow. Intellectualization of estimation and prediction in complex systems failures risk is a task related to the development of methods and technologies that allow automatically determining the risk (probability) of failures in CTS and predicting their possible consequences.

Assessing and predicting the risk of CTS failures is a task that requires the ability to process large amounts of data, analyze them and find patterns and relationships between various system operation parameters. In this case, the use of intelligent technologies can greatly simplify and speed up the process of data analysis.

In accordance with the requirements of the Register of Maritime Navigation, all modern ships must be equipped with automation systems for technical means using digital technologies, as well as AI technologies [15,16]. So, for example, an AI system for power plants should monitor the state and control of engines and their systems, auxiliary mechanisms, power supply systems in accordance with the plan for their maintenance and inspection.

The conceptual basis for the intellectualization of the solution of interrelated problems of CTS TC diagnostics and prediction are traditional for the class of unstructured and poorly formalized tasks, such as the impossibility of obtaining complete and objective information for making adequate decisions and, due to this circumstance, the need to involve informal (subjective, heuristic) information; the presence of uncertainty in the initial data, as well as the presence of ambiguity (multiple options) in the process of finding a solution; the necessity to develop and justify the desired solutions to the problem under severe time constraints determined by the course of controlled processes; the necessity to correct and introduce additional information into the process of finding solutions, the interactive (dialogue, human-machine) nature of the logical inference of solutions. Taking these factors into account forces us to abandon traditional algorithmic methods and decision-making models and move on to intelligent system technologies [17].

To successfully solve the problem of ensuring the reliability of ship CTS, it is necessary to remove a number of uncertainties, each of which is quite complex and significant. Such uncertainties include: incomplete data on external, internal impacts on systems and on the state of such systems; uncertainty in the behavior of systems.

The removal of the listed uncertainties can be based on solving the problems of assessing the risk of CTS failures and its prediction with relative insensitivity to incomplete data on systems [18].

Intellectualization of automated diagnostic systems involves solving a number of interrelated tasks of a structural, functional, informational and organizational nature, which should be provided at the design stage of TC CTS diagnostic systems [17].

There are various algorithms, models and methods for predicting TC CTS. AI models are actively developing, in particular, neural networks. To solve the problem of classifying TC based on diagnostic results, for example, a probabilistic neural network (PNN network - Probabilistic Neural Network [19]) is used.

However, the main problem for the productive operation of a neural network is the need for a significant amount of statistical data, which is difficult to obtain in real conditions due to a number of reasons (high cost of the systems under study, high costs of testing, limited time, etc.).

Lack of a clear understanding in the choice of neural network architecture for solving various types of problems (pattern recognition, approximation, forecasting, etc.) and areas of application also complicates their application.

In AI, knowledge representation models are actively developing - Bayesian networks [20]. Bayesian networks allow us to combine a priori (initial) knowledge about an object with new (experimental) data to obtain aposteriori (after experimental) estimates. One of the advantages of using Bayesian belief networks (BBN) for TC diagnostics is their ability to work with uncertain and incomplete data.

Instead of using rigid rules and thresholds, which may be ineffective in case of complex and ambiguous situations, the BBN might be useful to provide the possible technique for estimation of the probability (risk of failures) of various system states based on the available data.

Using BNN for CTS implementation let us to reduce the probability (risk of failures) of false reactions and improves diagnostics accuracy. In addition, the BBN can be used to analyze not only individual components of the system with different performance, but also their interactions, connections, which allows us to identify both individual faults and their interactions, which can be useful in solving complex problems associated with failures in systems.

To assess the risk of CTS failures based on the BBN, modern and available software technologies are used (Microsoft Bayesian Network Editor, Bayes Net Toolbox for Matlab, GeNIe, Smile, AgenaRisk, Analytica, Bayes Server, Hugin Expert). In addition, there are ready-made libraries and modules for Python, C++, C#, MatLab, R, VB.NET on various operating systems (Windows, Linux, macOS).

Thus, the problems associated with ensuring the reliable operation of the CTS require improvement and the search for new methods, models and algorithms implemented in the form of problem-oriented programs.

They should be aimed at prompt detection of emergency conditions of equipment, at solving problems of system failures risk assessing and predicting under conditions of relative insensitivity to incomplete data on FE and FI, which have different operability.

Since all modern ships must be equipped with AI-based technology automation systems, the implementation of approaches based on such methods, models and algorithms should be aimed at ensuring the reliable operation of shipboard CTS.

So, taking into account the specifics and existing problems in ensuring reliability during the CTS shipboard operation, the development and development intellectualization methods, models for estimating and predicting the risk of failures of complex systems by diagnostic features are important for the new technologies development performance aimed at ensuring complex systems safety and reliability.

Statement of the problem: intellectualization of the CTS TC assessment by diagnostic features, to substantiate the forecast failures risk in subsystems (components), intersystem (intercomponent) connections of systems with different performance.

Purpose of the work:

- ensuring the reliability and safety of work, as well as reducing the risk of CTS failures by solving causes determining problem of their failures;
- formation of principles for the construction and intelligent system operation for assessing and predicting the risk of CTS failures with different performance, their constituent FE and FI;

- intellectualization method and model for the TC estimation and predicting complex systems failures risk by diagnostic features development. The results might be used relatively insensitive to incomplete data about systems based the priori information about failures that connects the types of technical condition FE and FI of complex systems and their diagnostic features.

3. Intellectualization method and model of complex systems failures risk technical state estimation and prediction by diagnostic features

The proposed method is based on a formalized generalized intellectualization model of TC failures risk estimation and prediction of complex systems FE and FI by diagnostic features, which can be described in the following form:

$$< G, S(C), I_S(I_C), R_{S(C)}, R_{I_S(I_C)}, L >, \quad (1)$$

where: $S(C)$ set FE CTS;

$I_S(I_C)$ set FI CTS;

$R_{S(C)}, R_{I_S(I_C)}$ set of diagnostic failure risk assessments FE and FI CTS;

G acyclic directed graph;

L mapping relationships between sets $S(C), I_S(I_C)$ and $R_{S(C)}, R_{I_S(I_C)}$, based on the fault tree of the CTS diagnostic model.

A failures risk diagnostic assessment set of FE and FI CTS

$$R\left\{R_{S(C)_{n(m)}}, R_{I_{S(C)_{a(z)}}}\right\}, \quad (2)$$

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

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

where $r_{S(c)_{n(m)}}$ FE CTS failure risk;

$r_{i_{s(c)_{a(z)}}}$ FI CTS failure risk;

$n_{S(c)}$ FE CTS number;

$m_{S(c)}$ FE CTS hierarchical level number;

$N_{S(C)}$ FE CTS quantity;

$M_{S(C)}$ FE CTS hierarchical level quantity;

a intersystem link number;

z interconnect number;

A number of interconnections;

Z number of intercomponent bonds.

Created model for FE, FI failures risk determining:

$$< P_{S(C)_{n(m)}}, P_{I_{S(C)_{a(z)}}}, D_{S(C)_{n(m)}}, D_{I_{S(C)_{a(z)}}}, e_{S(c)_{n(m)}}, e_{I_{s(c)_{a(z)}}} >, \quad (3)$$

where $P_{S(C)_{n(m)}}, P_{I_{S(C)_{a(z)}}}$ conditional failure probabilities FE and FI respectively;

$D_{S(C)_{n(m)}}, D_{I_{S(C)_{a(z)}}}$ failure damage FE and FI respectively;

$e_{s(c)_{n(m)}}, e_{I_{s(c)_{a(z)}}}$ FE and FI weight given their hierarchy in CTS respectively.

Failure risk for $n(m)$ FE CTS:

$$R_{S(C)_{n(m)}} = D_{S(C)_{n(m)}} \cdot P_{S(C)_{n(m)}}(t). \tag{4}$$

Failure risk for $a(z)$ FI CTS

$$R_{I_{S(C)_{a(z)}}} = D_{I_{S(C)_{a(z)}}} \cdot P_{I_{S(C)_{a(z)}}}(t). \tag{5}$$

The total failure risk assessment CTS, taking into account the failures risk assessment of FE, FI is determined

$$\mathrm{R} = \sum_{\mathrm{s(c)}=1}^{\mathrm{S(C)}} \sum_{n(m)=1}^{N(M)} (\mathrm{R}_{\mathrm{s(c)n(m)}} \cdot e_{S(c)_{n(m)}}) + \sum_{i_{s(c)}=1,a(z)=1}^{I_{S(C)}} \sum^{A(Z)} (R_{i_{s(c)a(z)}} \cdot e_{I_{s(c)a(z)}}). \tag{6}$$

The failure probability of FE and FI is determined by the formulas:

$$P_{S(C)_{n(m)}} \cdot \lambda(t)_{S(C)_{n(m)}} = \frac{\alpha_{S(C)_{n(m)}} \cdot \exp(-\alpha_{S(C)_{n(m)}} \cdot T_{S(C)_{n(m)}})}{\exp(-\alpha_{S(C)_{n(m)}} \cdot T_{S(C)_{n(m)}})} = \alpha_{S(C)_{n(m)}}, \tag{7}$$

$$P_{I_{S(C)_{a(z)}}} \cdot \lambda_{I_{S(C)_{a(z)}}}(t) = \frac{\alpha_{I_{S(C)_{a(z)}}} \cdot \exp(-\alpha_{I_{S(C)_{a(z)}}} \cdot T_{I_{S(C)_{a(z)}}})}{\exp(-\alpha_{I_{S(C)_{a(z)}}} \cdot T_{(I_{S(C)_{a(z)}})}} = \alpha_{I_{S(C)_{a(z)}}} \tag{8}$$

where λ failure rate;

α distribution parameter, taken according to the test results equal to $\alpha \approx 1/\hat{T}_0$, $\hat{T}_0$ mean time to failure estimate.

Quantifying damage to FE from failure $n(m)$ subsystem (component) to determine the risk of failure

$$D_{S(C)_{n(m)}} = \{d_{s(c)_{n(m)}} \mid s(c) = \overline{1, S(C)}, n = \overline{1, N}, m = \overline{1, M}\}, \tag{9}$$

where $d_{s(c)_{n(m)}}$ damage from subsystem (component) failure CTS.

Quantifying damage FI from failure $a(z)$ intersystem (intercomponent) communication:

$$D_{I_{s(c)a(z)}} = \{d_{I_{s(c)a(z)}} \mid i_{S(c)} = \overline{1, I_{S(C)}}, a = \overline{1, A}, z = \overline{1, Z}\}, \tag{10}$$

where $d_{i_{s(c)a(z)}}$ damage from failure of intersystem (intercomponent) communication.

According to the established conditional failure probabilities and damages from failures FE and FI according to (4), (5), their risk of failures is determined.

The intellectualization model of complex systems TC failures risk estimation and prediction by diagnostic features using the BBN apparatus is a synthesis of reliability and diagnostic models. In the technical condition diagnostics model, the BBN is used to assess the risk of system failure (probability).

To create a diagnostic TC model, it is necessary to determine the risk of failure (conditional probabilities) for each node in the network. This data is derived from expert knowledge and analysis of historical data.

After determining the risk of failure (conditional probabilities), the model can be used to estimate (predict) the CTS state.

To do this, the model determines the risk of failure for system's each state using information about the current values of the system's state and the failures risk determined for each node in the network.

The technique for building a model based on BBN can be represented as follows:

1. Building a BBN:

1.1 Vertices and intersystem (intercomponent) Bayesian networks are created, denote subsystems (components) of CTS, according to their TC:

1.1.1 Each subsystem (component) can be in the following technical condition:

$Work_{n_{S(C)}}^{<m_{S(C)}>}$ operational state $n_{S(C)}$ subsystem (component) $m_{S(C)}$ level;

$Not\_work_{n_{S(C)}}^{<m_{S(C)}>}$ failure partial (complete) $n_{S(C)}$ subsystem (component) $m_{S(C)}$ level.

1.1.2. Each intersystem (intercomponent) connection is in the states:

$Work_{a(z)_{I_{S(C)}}}^{<b,q>}$ operational state $a(z)_{I_{S(C)}}$ intersystem (intercomponent) communication $b(q)$ level;

$Not\_work_{a(z)_{I_{S(C)}}}^{<b,q>}$ failure partial (complete) $a(z)_{I_{S(C)}}$ intersystem (intercomponent) communication $b(q)$ level, where b number of hierarchical level of system interconnection; q number of hierarchical level of component interconnection.

1.2. The connections between the nodes of the Bayesian network are indicated, denoting subsystems (components), intersystem (interelement) CTS connections and diagnostic appropriate assessments R.

2. BBN parameters are specified:

2.1. Initial failure risk for FE and FI CTS, assuming that they are all operational before the start of the CTS

$$R(Work_{n_{S(C)}}^{<m_{S(C)}>})_{t=0} = F(P(Work_{n_{S(C)}}^{<m_{S(C)}>})_{t=0}) = 0; \quad (11)$$

$$R(Work_{a(z)_{I_{S(C)}}}^{<b,q>})_{t=0} = F(P(Work_{a(z)_{I_{S(C)}}}^{<b,q>})_{t=0}) = 0.$$

2.2. Initial failure risk for FE and FI CTS, assuming that they are all inoperable before the start of the CTS

$$R(Not\_work_{n_{S(C)}}^{<m_{S(C)}>})_{t=0} = F(P(Not\_work_{n_{S(C)}}^{<m_{S(C)}>})_{t=0}) = 1; \quad (12)$$

$$R(Not\_work_{a(z)_{I_{S(C)}}}^{<b,q>})_{t=0} = F(P(Not\_work_{a(z)_{I_{S(C)}}}^{<b,q>})_{t=0}) = 1.$$

2.3. Risk of failure of FE and FI CTS at the current moment of time, provided that some FE and FI failed at the previous moment of time

$$R((Not\_work_{n_{S(C)}}^{<m_{S(C)}>})_t /(Not\_work_{n_{S(C)}}^{<m_{S(C)}>})_{t-1}) = 1; \quad (13)$$

$$R((Not\_work_{a(z)_{I_{S(C)}}}^{<,b,q>})_t /(Not\_work_{a(z)_{I_{S(C)}}}^{<b,q>})_{t-1})) = 1.$$

2.4. FE and FI CTS failure risk at the current moment of time, while they are in a working state at the current moment of time, which also were operable at the previous moment of time

$$R((Work_{n_{S(C)}}^{<m_{S(C)}>})_t /(Work_{n_{S(C)}}^{<m_{S(C)}>})_{t-1}) = \frac{e^{-\lambda_{n_{S(C)}}^{<m_{S(C)}>} t}}{e^{-\lambda_{n_{S(C)}}^{<m_{S(C)}>} (t-1)}} = e^{-\lambda_{n_{S(C)}}^{<jm_{S(C)}}} = 0; \quad (14)$$

$$R((Work_{a(z)_{I_{S(C)}}}^{<b,q>})_t /(Work_{a(z)_{I_{S9C0}}}^{<b,q>})_{t-1}) = \frac{e^{-\lambda_{a(z)_{I_{S(C)}}}^{<b,q>} t}}{e^{-\lambda_{a(z)_{I_{S(C)}}}^{<b,q>} (t-1)}} = e^{-\lambda_{a(z)_{I_{S(C)}}}^{<b,q>}} = 0.$$

2.5. FE and FI CTS failure risk at the current moment of time, subject to failure of FE and FI at the current moment of time, provided that it were operational at the previous moment of time

$$R((Not\_work_{n_{S(C)}}^{<m_{S(C)}>})_t /(Work_{n_{S(C)}}^{<m_{S(C)}>})_{t-1}) = (1 - e^{-\lambda_{n_{S(C)}}^{<m_{S(C)}>}}) \cdot D_{S(C)_{n(m)}}; \quad (15)$$

$$R((Not\_work_{a(z)_{I_{S(C)}}}^{<b,q>})_t /(Work_{a(z)_{I_{S(C)}}}^{<b,q>})_{t-1}) = (1 - e^{-\lambda_{a(z)_{I_{S(C)}}}^{<<b,q>}}) \cdot D_{I_{S(C)_{a(z)}}}$$

4. Experiments and results analysis

As an example, when modeling the BBN SPP, various failure risk values input element were selected and the predicted failures risk values and operability of the FE and FI of the power plant for 20,000 hours of installation operation were determined. Symbols of subsystems, components of the ECS are given in Table 1.

The purpose of using BBN in assessing how the risk of loss of performance due to the risk of failures FE and FI CTS is posteriori.

The a priori data are dynamically recalculated and form a posterior failure risk estimate, which is a priori information, to process the new information. Post hoc inference is based on procedures for analyzing data obtained from the use of BBN.

Failure risk value predicted distributions for blocks and links of the multilevel structure diagnostic model (shown in Fig. 1) in BBN with a serial connection, for example, SPP subsystems IE - CAS - SPP and interconnections IE-CAS, CAS - SPP and SPP operation for 20,000 hours for SPP shown in Fig.2.

From the given a priori and a posteriori data based on the SPP study results included in the SPP multi-level structure, for 20,000 hours of operation for a conditionally accepted SPP failure risk value input component of 0.26 and possible failures of IE, CAS and interconnections IE - CAS, CAS - SPP according to posterior data, the predicted risk of SPP failure will change from 0.25 to 0.68.

Table 1
Symbols of subsystems, components of the SPP

Subsystem (component) number $n_{s(c)}$	Hierarchical number subsystem (component) level ($m_{s(c)}$)	Subsystem (component) name	Symbol	Subsystem (component) weight $e_{s(c)_{n(m)}}$
1	1	Input element	IE	0,26
2	2	Firefighting system	FFS	0,01
3	2	Compressed air system	CAS	0,047
4	2	Main engine manual control	MCME	0,035
5	3	Control system	CS	0,081
6	3	Main engine remote automated control system	RACSME	0,01
7	3	Intermediate component	P1	0,01
8	3	Ship power plant	SPP	0,09
9	4	Main engine	ME	0,16
10	4	Ballast drainage system	BDS	0,019
11	4	Emergency drive propulsion and steering complex	ED PSC	0,01
12	4	Propulsion and steering complex control system	CSPSC	0,081
13	4	Boiler plant	BP	0,13
14	5	Power transfer from main engine to propeller	TPMEP	0,003
15	5	Intermediate component	P2	0,01
16	6	Propulsion and steering complex	PSC	0,01
17	7	Output component	EXIT	0,26

Similarly, from a retrospective results analysis obtained by modeling SPP, FE and FI are identified that affect the overall system performance.

From the research results, as reflected in the diagnostic model, it follows that the maximum non-operating state during the operation of the SPP 20000 hours corresponds to the CSPSC subsystem directly.

Since the CSPSC subsystem is dependent in the level of the SPP layered structure, then in the future it is necessary to check the subsystem in order to find its failure cause.

The implementation of the intellectualization method and model for estimation and prediction CTS failures risk in the conducted studies, carried out by modeling on a priori and a posteriori data, determines the FE and FI SPP that have the greatest impact on the performance of the main engine and the operation of the entire system at various points in time.

Carrying out studies of emergency situations, analysis of incidents in the CTS will determine the causes of the accident FE and FI SPP.

Thus, based on the intellectualization of FE and FI CTS TC estimation by diagnostic features, it is possible to substantiate the failure risk prediction for FE and FI systems with a large number of variable parameters and with different operability.

The considered principle of intelligent system functioning (due to its structure) in terms of the technical and technological foundations of construction on the SPP example reflected in the method and model for assessing and predicting the FE and FI CTS risk failures.

Method and model implementation with different operability and incomplete data let us to assess and predict the risk of CTS failures on network infrastructures with relative insensitivity to incomplete data.

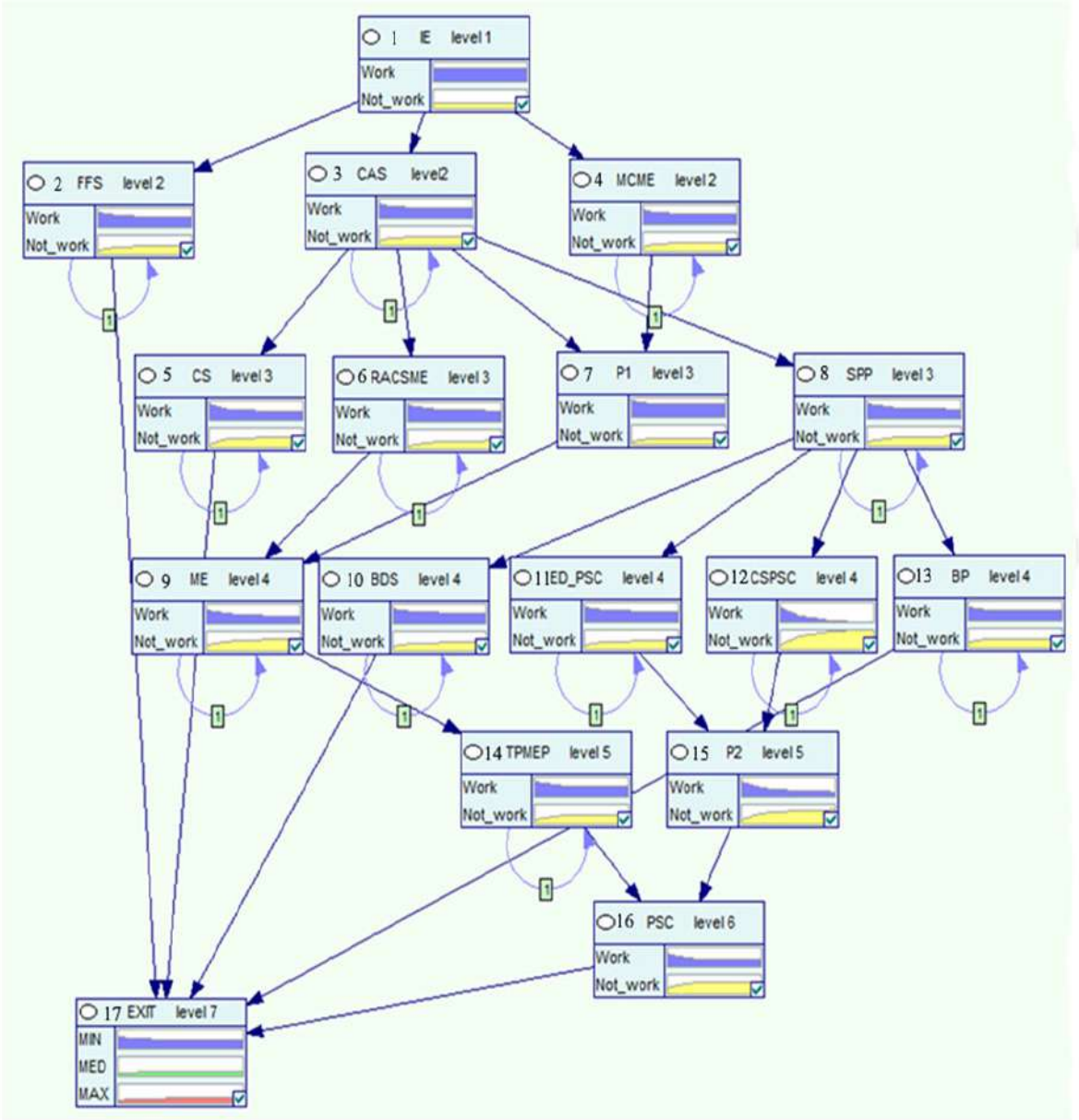


Figure 1: Diagnostic model of the technical condition of the SPP using the BBN in the GeNIe environment

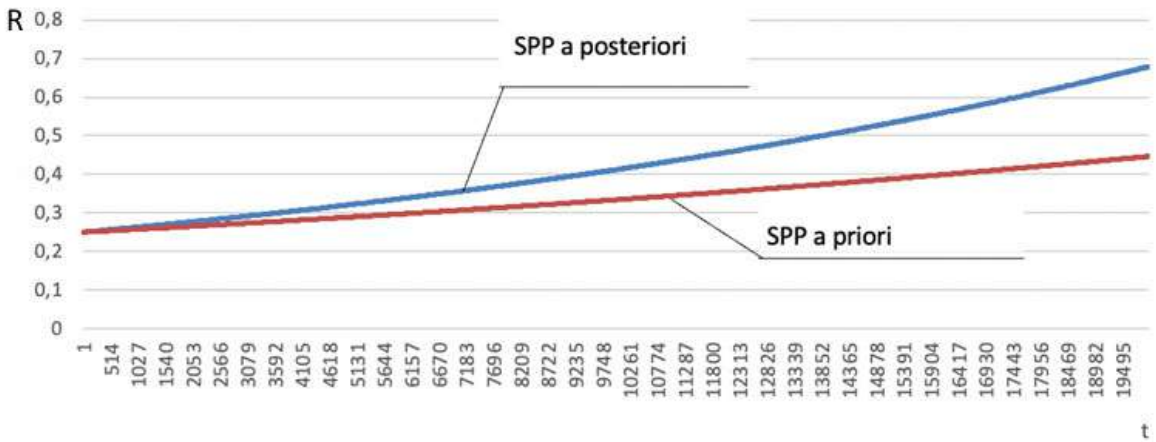


Figure 2: A posteriori and a priori estimates of the risk SPP for the input component failure risk of the ECS 0.26

5. Conclusion

Intellectualization of the CTS TC assessment by diagnostic features, the failures risk predicted values in subsystems (components), intersystem (intercomponent) connections of systems with different operability and incomplete data are substantiated.

In order to ensure the reliability and safety of work, as well as reduce CTS failures risk the all our main tasks were solved. It was determining the causes of their failures, we formated of principles for the construction and operation of an intelligent system for assessing and predicting CTS risk failures with different performance, their constituent FE and FI; development of a method and model for the intellectualization TC estimation and predicting complex systems failures risk by diagnostic features that are relatively insensitive to incomplete data about systems, based on the use of a priori information about failures, linking the types of technical condition FE and FI of complex systems and their diagnostic features in the bounce risk form.

The use of the developed method and model, which takes into account the hierarchical levels of subsystems (components), intersystem (interelement) links when searching for the causes of failures in complex technical systems, allows us to control the risk of failures in systems when information about failures in their structures is received. The application of the method and model makes it possible to predict trends in the risk of system failures, with updated changes in the risk of failures of individual subsystems (components), intersystem (interelement) links in order to further choose a strategy for their recovery.

6. References

[1] V. Vychuzhanin, N. Rudnichenko, Devising a method for the estimation and prediction of technical condition of ship complex systems, Eastern-European Journal of Enterprise Technologies, 84 (2016) 4-11. DOI:10.15587/1729-4061.2016.85605.

[2] V. Vychuzhanin, N. Rudnichenko, T. Otradskya, I. Petrov, Data Mining Information System for Complex Technical Systems Failure Risk Evaluation, in: Proceedings of The Fifth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2022), CEUR-WS, 3137, 2022, pp.250-261. DOI: 10.32782/cmis/3137-21.

[3] ISO 13381-1:2015 Condition monitoring and diagnostics of machines – Prognostics – Part 1: General guidelines, 2015.

[4] V. Vyuzhuzhanin, N. Rudnichenko, N. Shibaeva, Data Control in the Diagnostics and Forecasting the State of Complex Technical Systems, Herald of Advanced Information Technology, 2 (2019) 183-196. DOI 10.15276/hait.03.2019.2.

[5] IEC 31010:2019 Risk management - Risk assessment techniques, 2019.

[6] V. Vychuzhanin, N. Rudnichenko, Assessment of risks structurally and functionally complex technical systems, Eastern-European Journal of Enterprise Technologies, 2 (2014) 18-22. DOI: 10.15587/1729-4061.2014.19846.

[7] M. Asuquo, J. Wang, L. Zhang, G. Phylip-Jones, An integrated risk assessment for maintenance prediction of oil wetted gearbox and bearing in marine and offshore industries using a fuzzy rule base method, in Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ, 2020. DOI:10.1177/1475090219899528.

[8] M. Zhang, J. Montewka, T. Manderbacka, P. Kujala, S. Hirdaris, A Big Data Analytics Method for the Evaluation of Ship - Ship Collision Risk reflecting Hydrometeorological Conditions, Reliability Engineering & System Safety, 213 (2021). DOI:10.1016/j.ress.2021.107674.

[9] Marine Accident Investigation Branch (MAIB) reports, 2022. URL: https://gisis.imo.org/Public/Default.aspx.

[10] Marine Accident Investigation Branch reports, 2022. URL: https://www.gov.uk/maib-reports.

[11] MARS Reports, 2022. URL: https://www.nautinst.org/resource-library/mars/mars-reports.html.

[12] Investigation Report, 2021. URL: https://ntsb.gov/investigations/AccidentReports/Pages/marine.aspx.

[13] Casualty and Event Intelligence – Keeping You One Step Ahead, 2020. URL: https://ihsmarkit.com/products/casualty-and-events.htm.

[14] Accident-investigation, 2022. URL: http://emsa.europa.eu/implementation-tasks/accident-investigation.
[15] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, M. Figari, Machine Learning Approaches for Improving Condition-based Maintenance of Naval Propulsion Plants, in Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment 230 (1), 2016, pp. 136–153. DOI:10.1177/1475090214540874.
[16] Z. Liu, J. Liu, F. Zhou, R. Liu, N. Xiong, A Robust GA/PSO-Hybrid Algorithm in Intelligent Shipping Route Planning Systems for Maritime Traffic Networks, Journal of Internet Technology, 19 (2018) 1635–1644. DOI:10.3966/160792642018111906001.
[17] C. Moreno, E. Espejo, A performance evaluation of three inference engines as expert systems for failure mode identification in shafts, Engineering Failure Analysis, 53 (2015) 24–35 DOI:10.1016/j.engfailanal.2015.03.020.
[18] 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, Volume 1670, pp.1-8. DOI:10.1088/1742-6596/1670/1/012014.
[19] D. Specht, Probabilistic neural networks, Neural Networks, 3 (1990) 109–118. DOI:10.1016/0893-6080(90)90049-Q.
[20] D. Handayani, W. Sediono, Anomaly Detection in Vessel Tracking: A Bayesian Networks (Bns) Approach, International Journal of Maritime Engineering (RINA Transactions Part A) 157 (2015) 145–152. DOI:10.3940/rina.ijme.2015.a3.316

Credibilistic Fuzzy Clustering Method Based on Evolutionary Approach of Crazy Wolfs in Online Mode

Alina Shafronenko[a], Yevgeniy Bodyanskiy[a], Iryna Pliss[a]

[a] *Kharkiv National University of Radio Electronics, Nauky ave 14, Kharkiv, 61166, Ukraine*

Abstract

The problem of big data credibilistic fuzzy clustering is considered. This task was interested, when data are fed in both batch and online modes and has a lot of the global extremums. To find the global extremum of the objective function of plausible fuzzy clustering, a modification of the crazy gray wolfs algorithm was introduced, which combines the advantages of evolutionary algorithms and global random search. It is shown that different search modes are generated by a unified mathematical procedure, special cases of which are well-known algorithms for both local and global optimization.

The proposed approach is quite simple in computational implementation and is characterized by high speed and reliability in tasks of multi-extreme fuzzy clustering.

Keywords 1

Fuzzy clustering, credibilistic fuzzy clustering, adaptive goal function, evolutionary algorithms, gray wolf optimization

1. Introduction

The task of classification in the mode of self-learning (clusterization) of multidimensional data is an important part of traditional intellectual analysis.

One of the main directions of computational intelligence is the so-called evolutionary algorithms, which are mathematical models of the evolution of biological organisms. The problem of clustering associated with vector observations often arises in many tasks of intelligent data analysis, such as Data Mining, Dynamic Data Mining, Data Stream Mining, Big Data Mining, Web Mining [1, 2] and primarily in fuzzy data clustering, when the processing of vector observations with different levels of probability, reliability, etc. can belong to more than one class.

The most effective are Kohonen's self-organizing maps [3] and evolutionary algorithms, which can improve data clustering in cases where data is processed in online modes.

From a computational point of view, the problem of clustering is reduced to finding the optimum of multi-extremal functions of a vector argument with the help of gradient procedures, which are repeatedly launched from different starting points. For speed up the process of finding extremums by using the ideas of evolutionary optimization.

Evolutionary computation [4] and swarm intelligence are considered the fastest growing algorithms that employ computational intelligence to solve optimization problems. The evolutionary swarm intelligence includes genetic algorithm [5], biogeography-based optimizer [6], evolutionary strategies [7], population-based incremental learning [8], genetic programming [9], and differential evolution [10].

Swarm intelligence [11] is another powerful form of the computational intelligence used to solve the optimization problems. Swarm intelligence algorithm using the ideas of evolutionary optimization, which includes algorithms inspired by nature, swarm algorithms, population algorithms,

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: alina.shafronenko@nure.ua (A. Shafronenko); yevgeniy.bodyaskiy@nure.ua (Y. Bodyanskiy); iryna.pliss@nure.ua (I. Pliss)
ORCID: 0000-0002-8040-0279 (A. Shafronenko); 0000-0001-5418-2143 (Y. Bodyanskiy); 0000-0001-7918-7362 (I. Pliss)


and imitate the natural swarms or communities or systems such as fish schools, bird swarms, cat swarms, bacterial growth, insect's colonies and animal herds etc. Evolutionary swarm intelligence algorithms include many algorithms such ant colony optimization [12], particle swarm optimization (PSO) [13], artificial bee colony [14], ant lion optimizer (ALO) [15], moth-flame optimization, dragonfly algorithm, sine cosine algorithm, whale optimization algorithm, multi-verse optimizer, grey wolf optimizer (GWO) [16], firefly algorithm, cuckoo search, and many others.

The purpose of the paper is to consider the problem of data fuzzy clustering that is received for processing in both batch and online modes. To find the global optimum of a multi-extremal objective function, to develop a method that can find the global extremum of complex functions.

2. Problem Statement

Baseline information for solving the tasks of clustering in a batch mode is the sample of observations, formed from N n-dimensional feature vectors $X = \{x_1, x_2, ..., x_N\} \subset R^n, x_k \in X, k = 1, 2, ..., N$.

The problem of fuzzy clustering of data arrays is considered in the conditions when the formed clusters arbitrarily overlap in the space of features. The source information for solving the problem is an array of multidimensional data vectors, formed by a set of vector-observations $X = (x(1), x(2), ..., x(k), ...x(N)) \subset R^n$ where k in the general case the observation number in the initial array, $x(k) = (x_1(k), ..., x_i(k), ..x_n(n))^T$. The result of clustering is the partition of this array on m overlapped classes with centroids $c_q \in R^n$, $q = 1, 2, ..., m$, and computing of membership levels $0 \le u_q(k) \le 1$ of each vector-observation $x(k)$ to every cluster c_q.

3. Credibilistic Fuzzy Clustering Method

Alternatively, to probabilistic and possibilistic procedures [17,18] it was introduced credibilistic fuzzy clustering approach using as its basis the credibility theory [19, 20] 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 [3].

$$Goal\left(u_q(k), c_q\right) = \sum_{k=1}^{N} \sum_{q=1}^{m} u_q^{\beta}(k) d^2\left(x(k), c_q\right) \tag{1}$$

where $u_q(k)$ - membership level, c_q - centroid, d - Euclidian metric, β - fuzzifier;

with constraints

$$\begin{cases} \sum_{q=1}^{m} u_q(k) = 1, \\ 0 < \sum_{k=1}^{N} u_q(k) < N. \end{cases} \tag{2}$$

Solution of nonlinear programming problem using the method of Lagrange indefinite multipliers leads to the well-known result

$$\begin{cases} u_q(k) = \dfrac{\left(d^2\left(x(k), c_q\right)\right)^{\frac{1}{1-\beta}}}{\sum\limits_{l=1}^{m}\left(d^2\left(x(k), c_l\right)\right)^{\frac{1}{1-\beta}}}, \\ c_q = \dfrac{\sum\limits_{k=1}^{N}\left(u_q(k)\right)^{\beta} x(k)}{\sum\limits_{k=1}^{N}\left(u_q(k)\right)^{\beta}} \end{cases} \tag{3}$$

coinciding with $\beta = 2$ with a popular method of Fuzzy C-Means of J. Bezdek (FCM) [21].

If the data are fed to processing sequentially, the solution of the nonlinear programming problem (1), (2) using the Arrow-Hurwitz-Uzawa algorithm leads to an online procedure:

$$\begin{cases} u_q(k+1) = \dfrac{\left(d^2\left(x(k+1), c_q(k)\right)\right)^{\frac{1}{1-\beta}}}{\sum\limits_{l=1}^{m}\left(d^2\left(x(k+1), c_l(k)\right)\right)^{\frac{1}{1-\beta}}}, \\ c_q(k+1) = c(k) + \eta(k+1) u_q^{\beta}(k+1) * \\ *\left(x(k+1) - c_q(k)\right). \end{cases} \tag{4}$$

The goal function of credibilistic fuzzy clustering has the form [19, 22] close to (1)

$$Goal\left(Cred_q(k), c_q\right) = \sum_{k=1}^{N}\sum_{q=1}^{m} Cred_q^{\beta}(k) d^2\left(x(k), c_q\right) \tag{5}$$

with "softer" than (2) constraints:

$$\begin{cases} 0 \le Cred_q(k) \le 1, \text{ for all } q \text{ and } k, \\ \sup Cred_q(k) \ge 0{,}5, \text{ for all } k, \\ Cred_q(k) + \sup Cred_l(k) = 1, \\ \text{for any } q \text{ and } k, \text{for which } Cred_q(k) \ge 0{,}5. \end{cases} \tag{6}$$

It should be noted that the goal functions (1) and (5) are similar and that there are no rigid probabilistic constraints in (6) on the sum of the membership in (2).

In the procedures of credibilistic clustering, there is also the concept of fuzzy membership, which is calculated using the neighborhood function of the form

$$u_q(k) = \varphi_q\left(d\left(x(k), c_q\right)\right) \tag{7}$$

monotonically decreasing on the interval $[0, \infty]$ so that $\varphi_q(0) = 1, \varphi_q(\infty) \to 0$.

Such a function is essentially an empirical similarity measure of [23] related to distance by the relation

$$u_q(k) = \frac{1}{1 + d^2\left(x(k), c_q\right)}. \tag{8}$$

Note also that earlier it was shown in [16] that the first relation (3) for $\beta = 2$ can be rewritten as

$$u_q(k) = \left(1 + \frac{d^2\left(x(k), c_q\right)}{\sigma_q^2}\right)^{-1}, \tag{9}$$

where

$$\sigma_q^2 = \left(\sum_{\substack{l=1 \\ l \ne \varepsilon}}^{m} d^2\left(x(k), c_l\right)\right)^{-1} \tag{10}$$

which is a generalization of the function (8) (for $\sigma_q^2 = 1$ (8) coincides with (10)) and satisfies all the conditions for (7).

In batch form the algorithm of credibilistic fuzzy clustering in the accepted notation can be written as [20, 22]

$$\begin{cases} u_q(k) = \left(1 + d^2\left(x(k), c_q\right)\right)^{-1}, \\ u_q^*(k) = u_q(k)\left(\sup u_l(k)\right)^{-1}, \\ Cred_q(k) = \dfrac{1}{2}\left(u_q^*(k) + 1 - \sup\limits_{l \neq q} u_l^*(k)\right), \\ c_q = \dfrac{\sum\limits_{k=1}^{N}\left(Cred_q(k)\right)^{\beta} x(k)}{\sum\limits_{k=1}^{N}\left(Cred_q(k)\right)^{\beta}} \end{cases} \tag{11}$$

and in the online mode, taking into account (9), (10):

$$\begin{cases} \sigma_q^2(k+1) = \dfrac{1}{\sum\limits_{\substack{l=1 \\ l \neq q}}^{m} d^2\left(x(k+1), c_l(k)\right)}, \\ u_q(k+1) = \left(1 + \dfrac{d^2\left(x(k+1), c_q(k)\right)}{\sigma_q^2(k+1)}\right)^{-1}, \\ u_q^*(k+1) = \dfrac{u_q(k+1)}{\sup u_l(k+1)}, \\ Cred_q(k+1) = \dfrac{1}{2}\left(u_q^*(k+1) + 1 - \sup\limits_{l \neq q} u_l^*(k)\right), \\ c_q(k+1) = c_q(k) + \eta(k+1) Cred_q^{\beta}(k+1)\left(x(k+1) - c_q(k)\right). \end{cases} \tag{12}$$

From the point of view of computational implementation, algorithm (12) is not more complicated than procedure (4) and, in the general case, is its generalization to the case of credibilistic approach to fuzzy clustering.

4. Evolutionary Approach of Crazy Wolfs

To find the global extremum of functions (1) and (5), it is advisable to use bio-inspired evolutionary algorithms for particle swarm optimization [24], among which the so-called gray wolf algorithm (GWO) can be noted as one of the fastest coding ones [25]. According to the algorithm proposed in [25], gray wolves live together and hunt in groups.

The search and hunting process can be described as follows:

$$C_{\alpha} = \left|B_1 * GW_{\alpha} - X(t)\right|,$$
$$C_{\beta} = \left|B_2 * GW_{\beta} - X(t)\right|,$$
$$C_{\delta} = \left|B_3 * GW_{\delta} - X(t)\right|,$$

if the prey is found, they first track, pursue and approach it.

If the prey runs, then the gray wolves chase, surround and watch the prey until it stops moving, which can be written as

$$GW_1 = GW_\alpha - A_1 * C_\alpha,$$
$$GW_2 = GW_\beta - A_2 * C_\beta,$$
$$GW_3 = GW_\delta - A_3 * C_\delta,$$

where GW - wolf movement function.

After the prey is found, the attack begins:

$$GW(t) = \frac{GW_1 + GW_2 + GW_3}{3}.$$

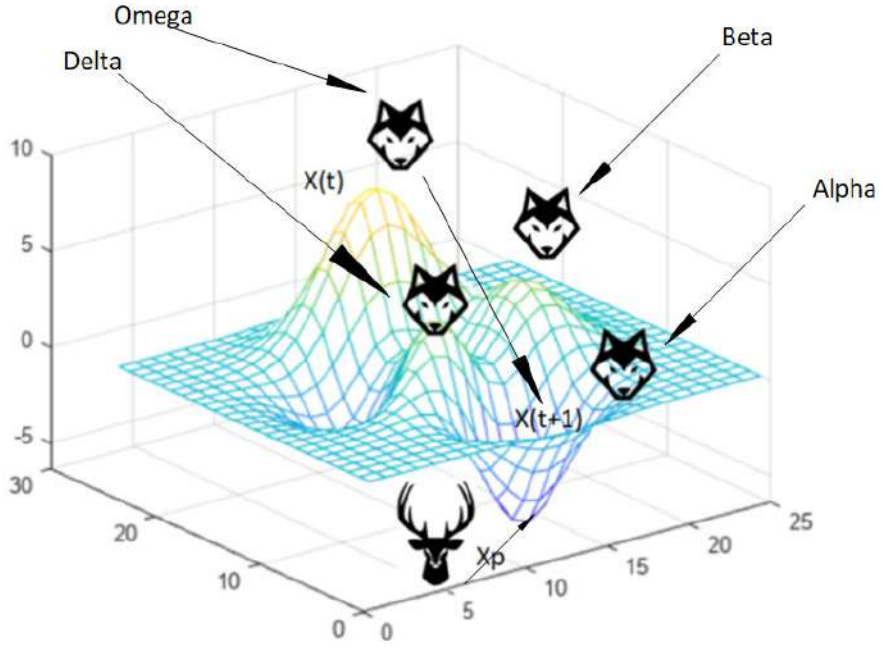


Figure 1: How $\alpha-, \beta-, \delta-$ wolfs are defined in GWO

In a case, when $|A| > 1$ - gray wolves flee from dominants, which means that omega wolves will flee from prey and explore more space; if $|A| < 1$ they approach the dominants, which means that δ-wolves will follow the dominants that approach the prey.

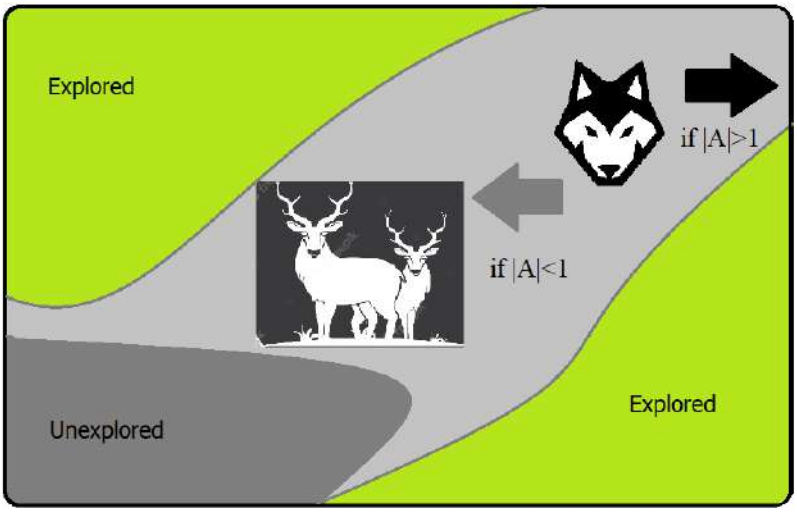


Figure 2: Exploration and exploitation of wolf in GWO

To achieve a proper trade-off between scouting and hunting, an improved gray wolf algorithm is proposed.

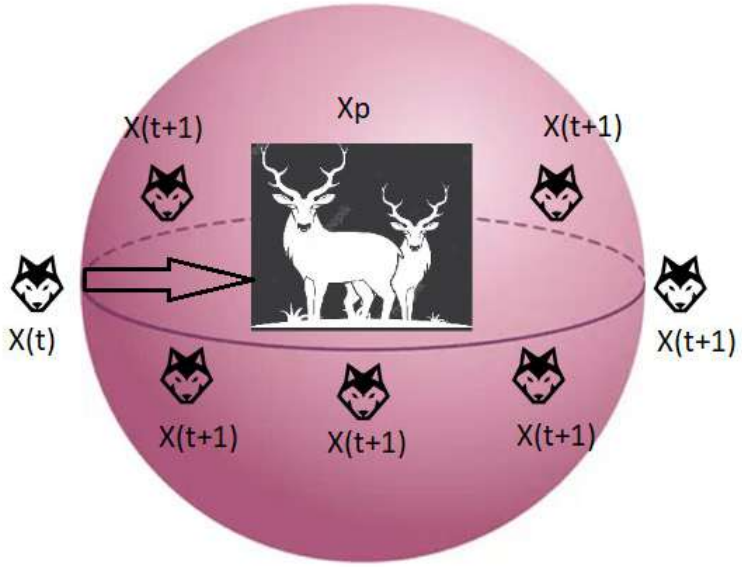


Figure 3: Scheme of operation of the gray wolf algorithm

The change in the positions of the wolves is described as follows:

$$C = \left| B * X_P(t) - X(t) \right|, \tag{10}$$

$$X(t+1) = X_P(t) - A * C + \Xi(\tau), \tag{11}$$

where X_P - prey position, X - position of the wolf, $\Xi(\tau)$ - a random perturbation that introduces an additional scan of the search space.

To increase the reliability of finding exactly the global extremum of the objective function, you can use the idea of "crazy cats" - optimization [26, 27], modified by the introduction of a random walk, which has proven its effectiveness when solving multi-extremal problems. By introducing, according to L. Rastrigin, an additional search perturbation, it is possible to write down the movement of the wolf in the form:

$$\Xi(\tau) = \gamma \Xi(\tau - 1) - \delta \left(X(t+1) - X(t) \right) + \sigma^2 H(k) \tag{12}$$

where γ - parameter for correction of disturbance characteristics, $0 \le \delta \le 1$ - self-learning speed parameter, σ^2 - white noise variance $H(\tau)$.

In this way, the probability of finding the global extremum of the adopted objective function increases, which ultimately increases the effectiveness of the fuzzy clustering process.

5. Experimental Research

The proposed method was tested on well-known multiextremal function such as Ackley's function, presented in form (13), Fig. 4:

$$f(x) = -20 \exp(-0.2\sqrt{0.5(x^2 + y^2)}) - \exp(0.5\cos(2\pi y)) + e + 20. \tag{13}$$

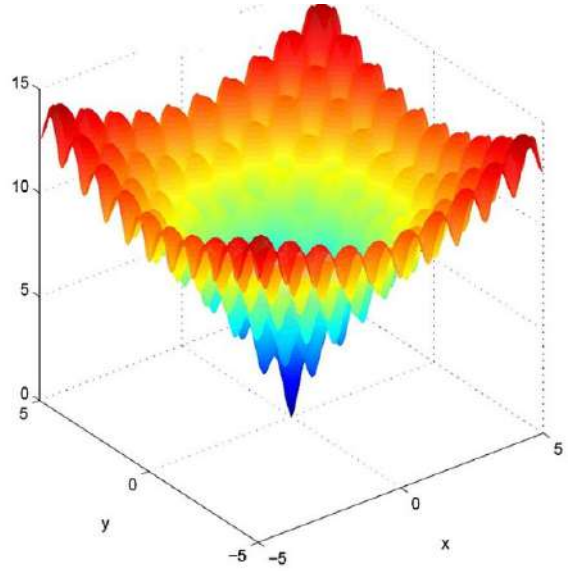


Figure 4: Ackley's function

Pseudocode of proposed method can write in the form of next steps presented in the Table 1:

Table 1
Pseudocode of Crazy Wolfs method

Description	Pseudocode
Setup options	Data sampling Swarm population Control parameter Stop criterion
Initialization	Initial positions of dominant gray wolves
Search	If this is not a stopping criterion, then the calculation of the new value of the fitness function Position update Restrictions on the positions of wolves Update α, β and δ Update stop criteria End

The behavior of Crazy GWO method can tested on multiextremal Ackley's function. Analyzes was provide on 100 iterations. On Fig. 5 demonstrate search history of global extremum of proposed function (a), the trajectory of the first variable of the first wolf (b), fitness history of all wolves (c) and the parameter A (d)

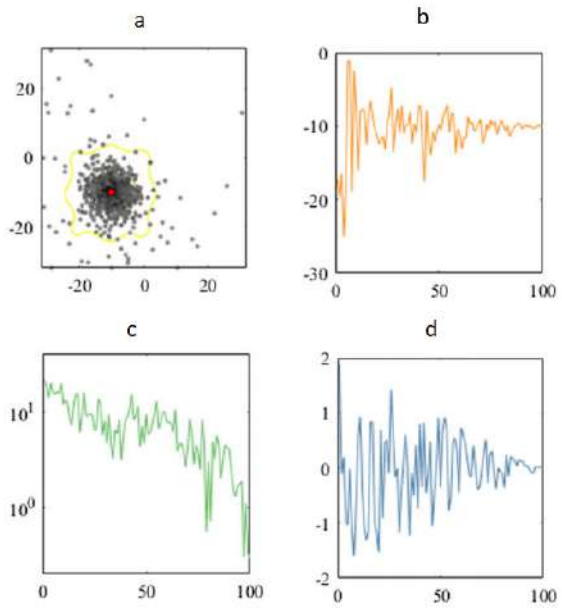


Figure 5: The history work of Crazy GWO method on 100 iterations

For further validating the searching accuracy of Crazy GWO method (CGWO) with GWO and PSO algorithms.

Table 2
Numerical statistics results

Accuracy	CGWO	GWO	PSO
Best	**1.5099e − 017**	7.5495e − 013	0.0035
Ave	1.2204e − 016	1.0048e − 012	0.0055
Worst	2.2204e − 014	1.4655e − 013	0.0082

Two swarm intelligence algorithms (PSO algorithm and GWO algorithm) are selected to carry out the simulation contrast experiments with the proposed CGWO so as to verify its superiority on the convergence velocity and searching precision. The simulation convergence results on the adopted testing functions are shown in Fig. 6:

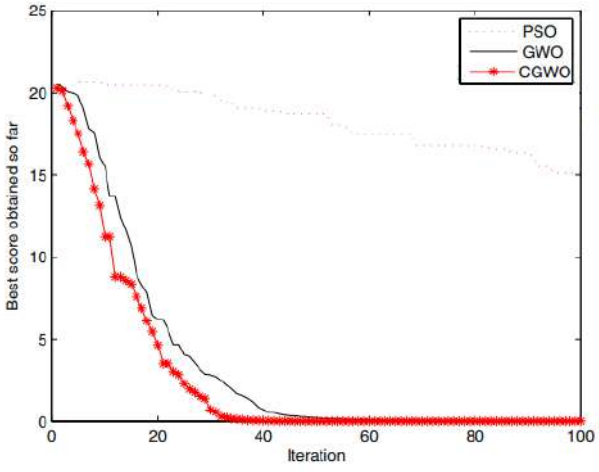


Figure 6: Convergence curves Ackley's function

6. Discussion

Analyzing the results of the obtained experimental studies and comparative analysis of the method of clustering data arrays based on the crazy gray wolf algorithm with clustering methods based on both the classical approach to data clustering and more exotic ones, the proposed method demonstrates sufficiently high results.

The main advantages of the proposed method are the simplicity of mathematical calculations, the speed of working with data, regardless of the type, size, and quality of the analyzed sample. It should be noted the accuracy of the data clustering method based on the improved gray algorithm and the obtained clustering results, which is achieved with the help of the optimization procedure of the evolutionary algorithm.

Conclusion

The problem of credibilistic fuzzy clustering of data using a crazy gray wolf algorithm is considered. The proposed method excludes the possibility of "getting stuck" in local extrema by means of a double check of finding the dominant wolf in the extremum and, in comparison with the given calculation error, allows to reduce the number of procedures starts. A feature of the proposed modified optimization procedure is the possibility of managing random disturbances that determine the properties of the modified the random walk, that has proven effectiveness when solving multi-extremal problems.

The approach is quite simple in numerical implementation, allows finding global extrema of complex functions, which is confirmed by the results of experimental research.

7. Acknowledgement

The work is supported by the state budget scientific research project of Kharkiv National University of Radio Electronics “Development of methods and algorithms of combined learning of deep neuro-neo-fuzzy systems under the conditions of a short training sample”.

8. References

[1] R. Xu, D.C. Wunsch, Clustering. Hoboken, N.J. John Wiley & Sons, Inc., 2009.
[2] C.C. Aggarwal, Data Mining: Text Book. Springer, 2015.
[3] T. Kohonen, Self-Organizing Maps. Berlin: Springer, 1995, 362 p. DOI: 10.1007/978-3-642-56927-2.
[4] T. Back, D.B. Fogel, Z. Michalewicz, Handbook of evolutionary computation. Oxford University Press, New York, 1997.
[5] L. Davis, Handbook of genetic algorithms. Van Nostrand Reinhold, New York, 1991.
[6] D. Simon, Biogeography-based optimization. IEEE Trans Evol Comput, 2008, 12(6):702–713.
[7] N. Hansen, S. Kern, Evaluating the CMA evolution strategy on multimodal test functions. International conference on parallel problem solving from nature. Springer, 2004, pp 282–291
[8] M. Hohfeld, G. Rudolph, Towards a theory of populationbased incremental learning. Proceedings of the 4th IEEE conference on evolutionary computation, 1997.
[9] J.R. Koza, Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge, 1992.
[10] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim, 1997, 11(4), pp.341–359.
[11] A. P. Engelbrecht, Fundamentals of computational swarm intelligence. Wiley, Hoboken, 2006.
[12] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput, 1997, 1(1), pp.53–66.
[13] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory. Proceedings of the sixth international symposium on micro machine and human science, 1995. MHS’95. IEEE, pp 39–43

[14] D. Karaboga, An idea based on honey bee swarm for numerical optimization. Technical report, technical report-tr06, Erciyes University, engineering faculty, computer engineering department, 2005.
[15] S. Mirjalili, The ant lion optimizer. Adv Eng Softw 83, 2015, pp.80–98.
[16] E. Elhariri, N. El-Bendary, A.E. Hassanien, A. Abraham, Grey wolf optimization for one-against-one multi-class support vector machines. 2015 7th international conference of soft computing and pattern recognition (SoCPaR). IEEE, 2015, pp 7–12.
[17] F.Höppner, F. Klawonn, R. Kruse, T. Runker, Fuzzy Clustering Analysis: Methods for Classification, Data Analysis and Image Recognition. Chichester, John Wiley &Sons, 1999, 289p.
[18] R. A Krishnapuram, "Possibilistic Approach to Clustering". IEEE Transactions on Fuzzy Systems, May 1993. Vol. 1. pp. 98-110. DOI: 10.1109/91.227387
[19] J. Zhou, Credibilistic clustering: the model and algorithms. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 2015. Vol. 23, №4. pp. 545-564. DOI: 10.1142/S0218488515500245.
[20] J. Zhou, Credibilistic clustering algorithms via alternating cluster estimation. Journal of Intelligent Manufacturing. 2017. Vol. 28. pp.727-738. DOI: 10.1007/s10845-014-1004-6.
[21] J.C. Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithms/ Bezdek J.C.// IEEE Transactions on Pattern Analysis and Machine Intelligence". IEEE, 1980. Vol. 2, №1. pp. 1- 8. DOI: 10.1109/TPAMI.1980.4766964.
[22] Shafronenko A., Bodyanskiy Ye., Klymova I., Holovin O., Online credibilistic fuzzy clustering of data using membership functions of special type. Proceedings of The Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020), April 27-1 May 2020. Zaporizhzhia, 2020. Access mode: http://ceur-ws.org/Vol-2608/paper56.pdf.
[23] Bodyanskiy Ye, Shafronenko A., Mashtalir S., Online robust fuzzy clustering of data with omissions using similarity measure of special type. Lecture Notes in Computational Intelligence and Decision Making-Cham: Springer, 2020. pp.637-646.
[24] Shafronenko A., Bodyanskiy Ye. Pliss I., Patlan K. Fuzzy Clusterization of Distorted by Missing Observations Data Sets Using Evolutionary Optimization. Proceedings "Advanced Computer Information Technologies (ACIT'2019)", Česke Budejovice, Czech Republic, June 5-7, 2019, pp. 217-220. doi: 10.1109/ ACITT.2019.8779888
[25] Shafronenko A. Yu, Bodyanskiy Ye. V., Pliss I.P. "The Fast Modification of Evolutionary Bioinspired Cat Swarm Optimization Method". 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL), Sozopol, Bulgaria, 6-8 Sept. 2019, pp. 548-552. doi: 10.1109/CAOL46282. 2019.9019583
[26] Saha S.K., Kar R., Mandal D., Ghoshal S.P. IIR filter design with craziness based particle swarm optimization technique. World Academy of Science, Engineering and Technology-2011-pp.1628-16356
[27] Sarangi A., Sarangi S.K., Mukherjee M., Panigrahi S.P. System identification by crazy-cat swarm optimization. Proc. Int. Conf. on Microwave, Optical and Communication Engineering-Bhubaneswar, India - 2015-Doc.18-29 pp. 439 – 442.

Application of vision transformers and 3D convolutional neural networks for sign language cluster recognition

Serhii Smirnov[a] and Nataliia Kuznietsova[a]

[a] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", ave. Peremohy 37, Kyiv, 03056, Ukraine*

Abstract

In this study the overview of the methods for sign language recognition was done and the existing datasets in this area were analyzed. It was shown how based on the real data to develop and test different approaches. The models based on the Vision Transformer (ViViT) and 3D Convolutions CNN (3dCNN) using different batch sizes were built and compared. It was also shown how to learn models on different data sizes and to search the compromise between accuracy, speed and overfitting of the models. Our research provides valuable insights into the strengths and limitations of different models of the task, not solved tasks and offer a direction and possible improvements of existed methods in this area by using vision transformers.

Keywords 1

Gesture recognition, computer vision, neural networks, deep learning, hand shape recognition, sign language interpreter, vision transformers, 3D convolutions.

1. Introduction

Sign language is a visual language used by people who are deaf or hard of hearing to communicate with each other and with hearing individuals. It involves using a combination of hand gestures, facial expressions, and body language to convey meaning. While sign language is an effective mean of communication, it can be challenging for non-signers to understand and communicate with sign language users. This has led to the development of sign language recognition technology, which uses computer algorithms to interpret and translate sign language into spoken or written language. Sign language recognition has the potential to improve communication and inclusion for people who are deaf or hard of hearing. It also poses unique challenges, such as the necessity for accurate hand shape and movement detection, real-time recognition, and dealing with the complexity and variability of different sign languages. Advantages and new achievements in machine learning, computer vision, and sensor technology give now the possibility to overcome these challenges and make sign language recognition more accurate, efficient, and accessible. Machine learning techniques, such as deep learning, can then be used to learn a mapping between these visual features and the corresponding sign language gestures. While CNNs have limitations in capturing long-term dependencies and global context, which are crucial for complex image understanding tasks such as object detection and segmentation, special transformers have gained significant popularity in the field of computer vision in recent years due to their ability to process sequential data such as images and videos. In this article we will compare two approaches for sign language recognition and define for the real practical task which of them is more effective and perspective for improvements for next studies.

2. Sign language recognition problem statement

Sign language is an essential mode of communication for deaf or hard-of-hearing individuals. Sign language recognition (SLR) is a challenging task, as sign languages are highly complex, with a wide

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: sergej.smirnov.sss@gmail.com (Smirnov S.); natalia-kpi@ukr.net (Kuznietsova N.)
ORCID: 0000-0002-5495-0680 (Smirnov S.); 0000-0002-1662-1974(Kuznietsova N.)

range of variations and nuances. SLR involves the hand gestures identification, facial expressions, and body movements to interpret the meaning of a sign [1, 2]. The goal of SLR is to develop systems that can recognize and translate sign language into written or spoken language which enable communication between hearing and non-hearing individuals. In this research we will discuss the problems, limitations, not solved issues, and existing solutions in SLR.

One of the primary challenges in SLR is the complexity of sign languages. There are over 300 sign languages used worldwide, each with its own grammar, vocabulary, and dialects. Furthermore, sign languages are highly context-dependent, with the meaning of signs often varying depending on the speaker's location, age, gender, and culture. Thus, developing an SLR system that can accurately recognize and interpret the nuances of different sign languages is a significant challenge.

Another challenge in SLR is the variability in signing styles. Signers may use different hand shapes, positions, and movements to convey the same message. Moreover, the speed and duration of signs can vary, adding further complexity to the task. Thus, SLR systems must be robust to variations in signing styles, as well as to variations in lighting conditions and camera angles. To address these challenges, researchers have proposed various techniques, such as data augmentation, transfer learning, and multi-modal fusion, which combine visual and other modalities such as audio or depth information.

One of the main challenges in sign language recognition is collecting and annotating large datasets of sign language gestures, which are required to train and evaluate machine learning models. This can be particularly difficult for sign languages that are not widely spoken or documented. A limitation of SLR is the lack of large-scale annotated datasets. While there are several datasets available for SLR, they are relatively small, limiting the performance of machine learning algorithms. A brief comparison of existed datasets for sign recognition task is made in Table 1. Moreover annotating sign language data is a time-consuming and challenging task, as it requires the expertise of sign language experts.

Table 1
Sign language recognition datasets comparison

Id	Name	Country	Classes	Samples	Language level	Availability
1	DGS Kinect 40	Germany	40	3000	Word	Contact author
2	RWTH-PHOENIX-Weather	Germany	1200	45760	Sentence	Publicly available
3	SIGNUM	Germany	450	33210	Sentence	Contact author
4	GSL 20	Greek	20	~840	Word	Contact author
5	Boston ASL LVD	USA	3300+	9800	Word	Publicly available
6	PSL Kinect 30	Poland	30	300	Word	Publicly available
7	PSL ToF 84	Poland	84	1680	Word	Publicly available
8	LSA64	Argentina	64	3200	Word	Publicly available
9	MSR Gesture 3D	USA	12	336	Word	Publicly available
10	DEVISIGN-G	China	36	432	Word	Contact author
11	DEVISIGN-D	China	500	6000	Word	Contact author
12	DEVISIGN-L	China	2000	24000	Word	Contact author
13	IIITA-ROBITA	India	23	unknown	Word	Contact author
14	Purdue ASL	USA	unknown	unknown	Word/ Sentence	Request DVDs/HD
15	CUNY ASL	USA	unknown	~33000	Sentence	Unknown
16	SignsWorld Atlas	Arabia	multiple types	unknown	Handshape, Words, Sentences	Unknown
17	LSA-T	Argentina	translation	14880	Sentence	Publicly available
18	LSFB-CONT	Belgium	6883	85000+	Word, Sentence	Publicly available

19	LSFB-ISOL	Belgium	400	50000+	Word	Publicly available
20	WLASL	EEUU	2000	21083	Word	Publicly available

Another issue in SLR is the lack of real-time performance. SLR systems often require high computational resources, making it challenging to achieve real-time performance on mobile devices or in low-resource settings.

Researchers have proposed various solutions to deal with these challenges and limitations [3–12]. There is an approach that proposes to use different deep learning algorithms, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to recognize the signs from video sources. These algorithms have shown promising results in SLR, achieving state-of-the-art performance on several benchmark datasets.

Another solution is to use depth sensors, such as Microsoft Kinect, to capture 3D motion data, which can be used to recognize signs accurately [8]. Depth sensors are advantageous as they can capture the 3D shape and position of the signer's hands, providing more robust and accurate sign recognition.

Furthermore, researchers have proposed the use of transfer learning, where pre-trained models on large datasets such as ImageNet are fine-tuned on sign language datasets. This approach has shown to improve the performance of SLR models, particularly for low-resource sign language datasets.

So, the SLR is a challenging task that requires robust and accurate recognition of complex hand gestures, facial expressions, and body movements. While several solutions have been proposed to address the limitations and challenges of SLR, there are still several unsolved issues, such as real-time performance, variability in signing styles, and lack of large-scale annotated datasets. With the development of more advanced algorithms and the availability of larger annotated datasets, it is hopeful that these challenges can be addressed, enabling better communication between hearing and non-hearing individuals. Overall, sign language recognition is still an important problem with potential applications in fields such as assistive technologies, education, and communication for deaf and hard-of-hearing individuals.

3. Brief overview of sign language recognition methods

There are various methods for sign language recognition (SLR) that have been proposed and tested over the years. Here are some of the most widespread methods for SLR:

1. ***Template matching*** is a simple and intuitive approach where the hand movements of the signer are matched against a predefined set of templates to recognize the sign [3, 4]. The template matching method involves capturing a series of hand poses and storing them as templates. During recognition, the input sequence is compared to each template, and the sign is identified based on the closest match. While this method is easy to implement, it is limited by the need for manually defining the templates and the inability to handle variations in signing styles.
2. ***Hidden Markov Models (HMMs)*** are probabilistic models used in speech recognition as a tool that can model the temporal dependencies in sign language by capturing the transitions between hand shapes and movements. The method involves training an HMM on a dataset of sign language gestures and using it to recognize signs in new sequences [5, 6]. However, HMMs can be limited in capturing the complex and context-dependent variations in sign language.
3. ***Support Vector Machines (SVMs)*** as a type of machine learning algorithm can classify sign language sequences by finding the hyperplane that separates the data points into their respective classes [7]. SVMs have been shown to achieve good accuracy in SLR, but they require large amounts of training data and may not be robust to variations in signing styles.
4. ***3D Depth Sensors*** using 3D depth sensors, such as Microsoft Kinect give the possibility to capture the 3D shape and position of the signer’s hands. This approach has the advantage of being more robust to variations in lighting and camera angles, and it can capture the depth

information of the hand movements[8, 9]. The depth information can be used to recognize signs more accurately.

5. ***Deep Learning*** methods, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been used in SLR and have shown significant improvements in performance. CNNs can learn the spatial features of sign language sequences, while RNNs can capture the temporal dependencies between hand movements [10, 11]. Additionally, attention mechanisms can be used to focus on the most relevant parts of the sequence, improving the accuracy of recognition.

Several approaches to deep learning could be defined for the relevant tasks and could be quite efficient in the real application.

PoseTCN is a deep learning model that uses temporal convolutional networks (TCNs) to capture the temporal dependencies in sign language gestures. PoseTCN takes as input a sequence of 3D hand pose data and outputs the recognized sign. The model uses dilated convolutions to increase the receptive field of the network and improve the model's ability to capture long-term dependencies [12, 13].

PoseTGCN is a deep learning model that uses a graph convolutional network (GCN) to capture the spatial dependencies between the joints in sign language gestures, and a temporal convolutional network (TCN) to capture the temporal dependencies. The model takes as input a sequence of 3D joint positions and outputs the recognized sign. The GCN operates on a graph structure where the joints are nodes and the edges represent the spatial relationships between them [14, 15]. The TCN operates on the resulting feature maps and uses dilated convolutions to capture long-term dependencies.

Inflated 3D ConvNet (I3D): I3D is a deep learning model that uses a 3D convolutional neural network (CNN) to extract spatio-temporal features from sign language gestures. The model takes as input a sequence of RGB or depth frames and as the outputs gives the recognized sign. The 3D CNN is pre-trained on large-scale video datasets, such as Kinetics or Sports-1M, and fine-tuned on the sign language recognition task [16, 17]. The pre-training allows the model to learn generalizable features that can be applied to sign language gestures.

Sign Language Transformers (SLT) is a transformer-based model that uses self-attention mechanisms to learn the spatial and temporal features of sign language gestures. SLT takes as input a sequence of RGB or depth frames and as the outputs the recognized sign. The model uses a pre-trained backbone network, such as ResNet or EfficientNet, to extract visual features from the frames, which are then fed into a transformer encoder-decoder architecture [18, 19]. The attention mechanisms in the model allow it to focus on the most relevant parts of the sequence and improve recognition accuracy. Now this approach is widely used in continuous sign language translation.

Transformers were originally designed for natural language processing (NLP) tasks where they exceed in capturing long-range dependencies and global context. They achieved this by incorporating self-attention mechanisms that allow the model to weigh the importance of different parts of the input sequence when making predictions. The same mechanism can be applied to images by treating each pixel or patch as a token, allowing the model to attend to different parts of the image when making predictions. Another advantage of transformers in computer vision is their ability to handle variable input sizes without requiring resizing or cropping. This is important for tasks such as object detection and segmentation where the size and aspect ratio of the objects can vary significantly. Additionally, transformers can leverage pre-training on large data amounts, allowing them to learn useful representations that can be fine-tuned on smaller datasets for specific tasks. Overall, the use of transformers in computer vision has shown promising results, outperforming traditional CNN-based architectures on various benchmarks and achieving state-of-the-art results on challenging tasks such as image captioning and visual question answering.

In summary, there are various deep learning models that can be used for sign language recognition, such as PoseTGCN, I3D, PoseTCN, and Sign Language Transformers (SLT). These models differ in their architecture and their ability to capture spatial and temporal dependencies in sign language gestures.

In Ukraine the task of sign recognition is really actual and important in context of the war and necessity to develop the special governments to support and provide assist and inclusion in society the

people who were suffered from the war and have problems with hearing. There are several works of national scientists who have investigated the problem of sign recognition and proposed special techniques and systems for communication and translating into the sign language [20–21]. Nevertheless, there are still unsolved issues and necessity of new approaches and adapting the existed methods is quite high.

4. Practical task of video interpretation for sign language

The ***goal*** in this article was to test mentioned above approaches and to build the simple transformer-based model for sign language recognition and compare its efficiency with the standard approach of using 3D convolutions. The idea was to clarify and define on a very first level (without any neural networks tricks or additional approaches) which approach could be more accurate for our task.

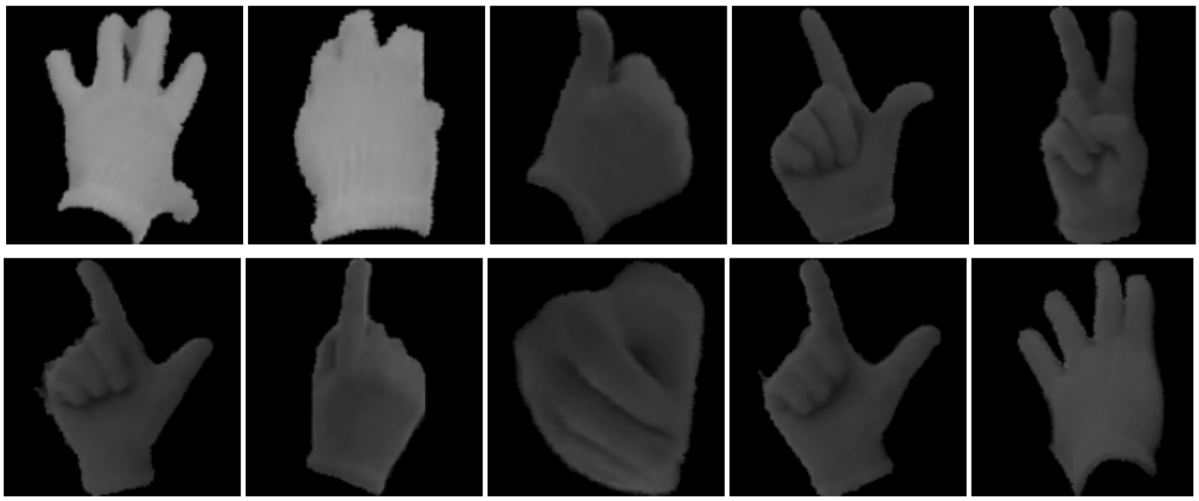
4.1. Dataset

For our experiments the LSA64 dataset [22] was used. LSA64 is a dataset for Argentinian Sign Language (LSA) and it represents a collection of video sequences designed for the task of sign language recognition in the Argentinian Sign Language. The dataset was created by researchers at the National University of Córdoba in Argentina, and contains 64 different LSA signs performed by 20 signers (10 male and 10 female).

The videos were recorded in a controlled environment using a high-definition camera and have a resolution of 1280x720 pixels at 25 frames per second. Each sign was performed five times by each signer and results were presented in a total of 6400 video sequences.

The LSA64 dataset also includes ground-truth annotations for each video, indicating the start and end frames of each sign. These annotations were performed manually by experts in LSA sign language.

The LSA64 dataset is quite challenging dataset due to variations in signing speed, camera viewpoint, and lighting conditions, making it a valuable resource for researchers working on developing robust and accurate sign language recognition algorithms (see some examples on Figure 1 and Figure 2).



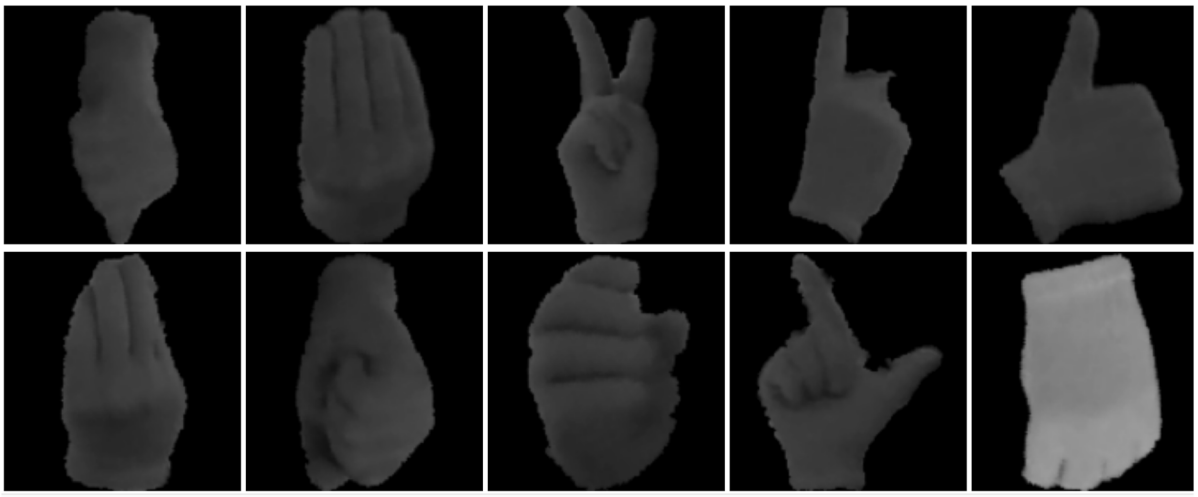


Figure 1: Screenshots of some examples of LSA64 dataset for sign language recognition

Figure 2: Example of one video storyboard

For our experiments firstly we made some labels for the dataset and then clustered them into 3 logical groups. That was done to have more labels in each of the ground-truth classes. The classes are: “colors” signs (consists of the next initial classes: “red”, “green”, “yellow”, “light-blue”), “food” signs (consists of the next initial classes: “sweet milk”, “water”, “food”) and “verbs” signs (consists of the next initial classes: “help”, “thanks”). The labels were randomly splitted into the train-test sets: 385 labels went to train and 165 labels - to the test set. Note that for experiments the proportion of each class was saved in both train and test sets. The general proportion of classes in data are: 36,3636% of the first class, 36,3636% of the second class, 27,2727% of the third class.

4.2. Used approaches

In this paragraph the transformer-based and 3D convolution-based models that were used in the experiments are described more detailed.

4.2.1. 3D convolutions

A neural network that uses 3D convolutions for video analysis typically consists of multiple layers of 3D convolutional, pooling, and fully connected layers [23].

3D convolutions are a type of convolutional layer that considers the spatial and temporal dimensions of the input data. In the case of video analysis, the input is a sequence of frames, and the 3D convolutional layer applies a kernel to each frame and its neighboring frames in the temporal dimension to extract features that capture both spatial and temporal information. This allows the model to learn patterns and movements over time, which is crucial for tasks such as action recognition and gesture recognition (see Figure 3) [24].

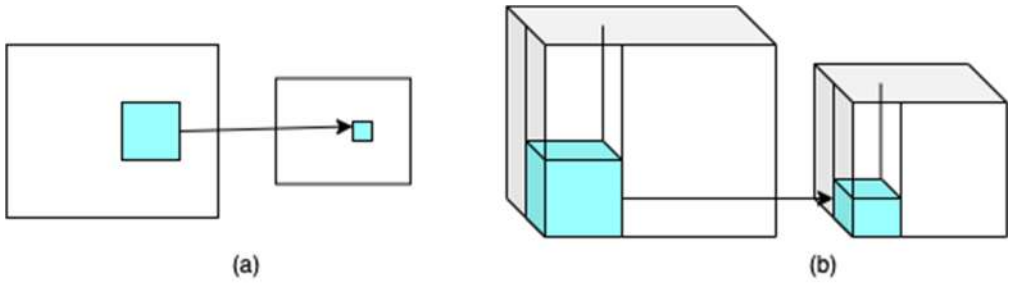


Figure 3: Comparison of 2D (a) and 3D (b) convolutions

After the 3D convolutional layers, pooling layers are often used to downsample the feature maps and reduce the spatial dimensionality of the data. This helps to reduce the parameters number in the model and prevent overfitting.

Finally, fully connected layers are used to classify the input video sequence into one or more classes. These layers take the flattened feature maps from the convolutional layers and apply a set of weights to produce a probability distribution over the possible classes.

4.2.2. Vision transformer

The approach of Video Vision Transformer (ViViT) [25, 26] involves dividing the video into small spatiotemporal regions of interest, called tubelets, and processing them using self-attention mechanisms similar to those used in the NLP models.

Here is a brief overview of how the ViViT model with tubelet embeddings works:

1. Tubelet Extraction: The first step is to extract tubelets from the input video. This can be done using a variety of techniques such as object detection, tracking, or motion analysis. Each tubelet is represented as a sequence of T frames, where each frame is a H x W x C tensor representing the pixel values of the video frame (see Figure 4).

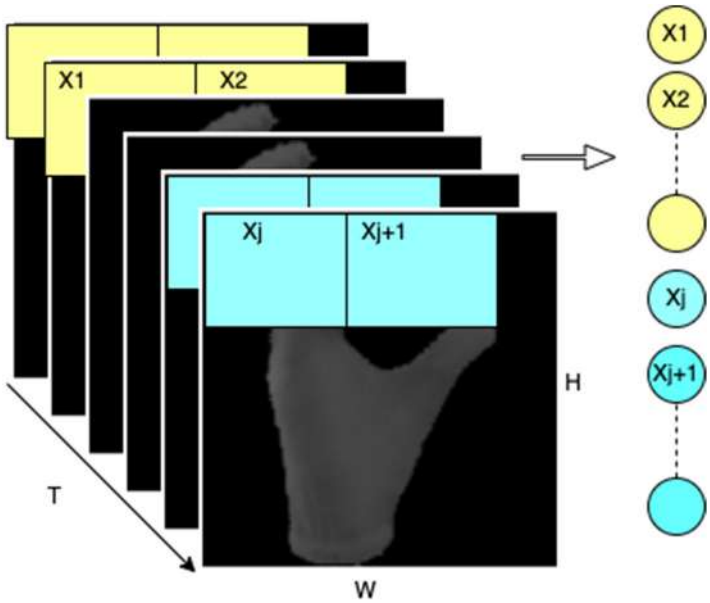


Figure 4: Tubelet embedding

2. Flattening and Linear Projection: Each frame in the tubelet is flattened into a sequence of patches, and these patches are then linearly projected into a higher-dimensional embedding

space of size D using a trainable linear layer. This results in a sequence of patch embeddings for each frame in the tubelet.

3. Multi-Head Self-Attention: The projected sequences are then passed through multi-head self-attention layers. Each layer computes attention weights between all pairs of patch embeddings in the sequence, and uses these weights to compute a weighted sum of the patch embeddings. This allows the model to attend to different regions of the tubelet depending on the task at hand.
4. Feedforward Network: After each self-attention layer, the output is passed through a feedforward network with a ReLU activation function. This network applies a linear transformation to the input, followed by a non-linear activation function. This helps the model capture more complex relationships between the patch embeddings in the sequence.
5. Aggregation: Finally, the output of the last self-attention layer is aggregated across all frames in the tubelet to obtain a single vector representation for the tubelet. These vectors are then passed through a linear layer to predict the class label for the entire tubelet.

By processing tubelets using self-attention mechanisms, ViViT with tubelet embeddings is able to better capture the spatiotemporal relationships between different regions of the video, resulting in improved performance on video recognition tasks such as action recognition and video classification.

5. Modelling & Results

In our experiments we used the vision transformer (ViViT) and 3D convolutions CNN (3dCNN) for comparison on our dataset, which was trained on the different batch sizes (3, 32, 128, 385 (the length of train dataset)). Each of the selected models was trained on 30 epochs, with learning rate equal to 1e-5, with input size of the videos equal to (25, 64, 64, 3). During the training the history of the accuracies is stored, so as a result only those model weights are saved and loaded for that approaches that had the best test accuracy. That was done to prevent the possible model overfitting. On Table 2 the models comparison table is presented, where top-1 (also known as accuracy) and top-2 are the top-K accuracy metrics calculated by the formula 1 below:

$$ttop-K\ accuracy = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \sum_{j=1}^{k} 1\left(\widehat{f_{i,J}} = y_i\right) \quad (1)$$

where $\widehat{f_{i,J}}$ is the predicted class for the i-th sample corresponding to the j-th largest predicted score, y_i is the corresponding true value, k is the number of guesses allowed and 1(x) is the indicator function. Top-K accuracy is often used for sign language recognition problems because it is a useful metric for evaluating the models performance dealing with a large number of possible signs and variations, as well as accounting for the flexibility required in recognizing signs.

Table 2
Comparison of vision transformer (ViViT) and 3D convolutions CNN (3dCNN) approaches

Approach name	test top-1	test top-2	train top-1	train top-2
ViViT/3	**69.7%**	89.7%	**87.01%**	89.7%
ViViT/32	64.85%	**90.3%**	78.96%	**90.3%**
ViViT/128	63.03%	89.09%	63.9%	89.09%
ViViT/385	60.0%	83.03%	65.97%	83.03%
3dCNN/32	63.64%	75.76%	58.18%	73.77%
3dCNN/128	52.12%	72.73%	51.17%	72.73%

In table 2 it is seen that the ViViT models with small batch sizes have higher quality. But having in mind the accuracy on training dataset as well we can notice that such models are more easily

overfitted. That means that ViViT model is faster in training flow to get the high quality (see Figure 5). On the other hand, for future investigations more approaches for fixing the model overfitting issue should be applied (e.g., augmentation), especially by working with small data amounts. Such logic could be traced on the plots below (Figures 6-11). Note, that in Figure 6-11 the loss and accuracy are normalized to be presented on the same scale, which gives the opportunity to analyze overfitting issues there.

We see that from the 20th epoch the accuracy for 3D convolution network (3dCNN/128) are extremely increasing. From the other hand, the accuracy on both training and test datasets for ViViT/3 and ViViT/32 models (Figure 6-7) as well as test losses are growing up and at the same time the losses on training dataset are decreasing. It also confirmed that the model was good learned on training dataset but on the test dataset it faced with some troubles. As for 3D convolution networks (Figures 10-11) the losses on both training and test datasets are decreasing with similar speed as well as accuracy are increasing.

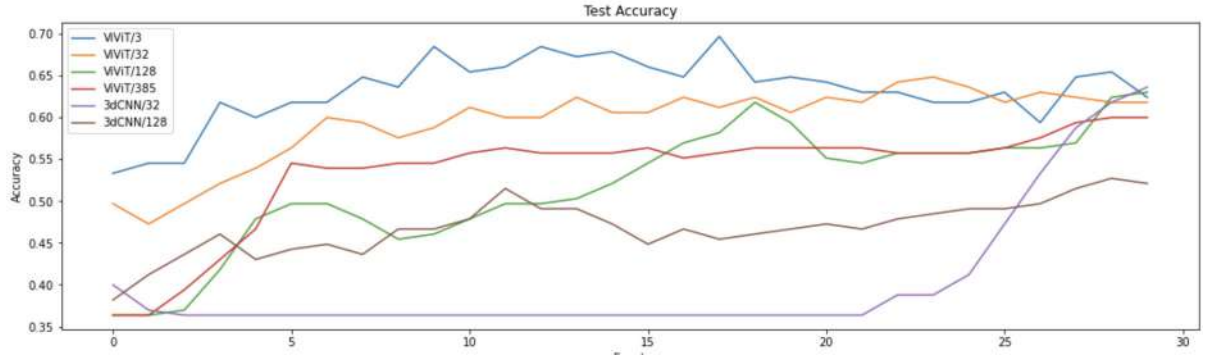


Figure 5: Accuracy on a test set for the different models on epochs scale

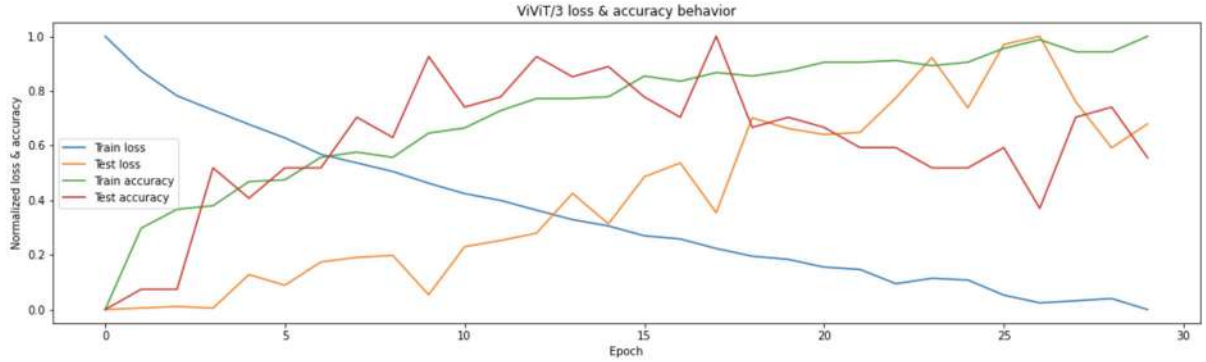


Figure 6: Train and test loss and accuracy comparison of ViViT/3 model on epochs scale

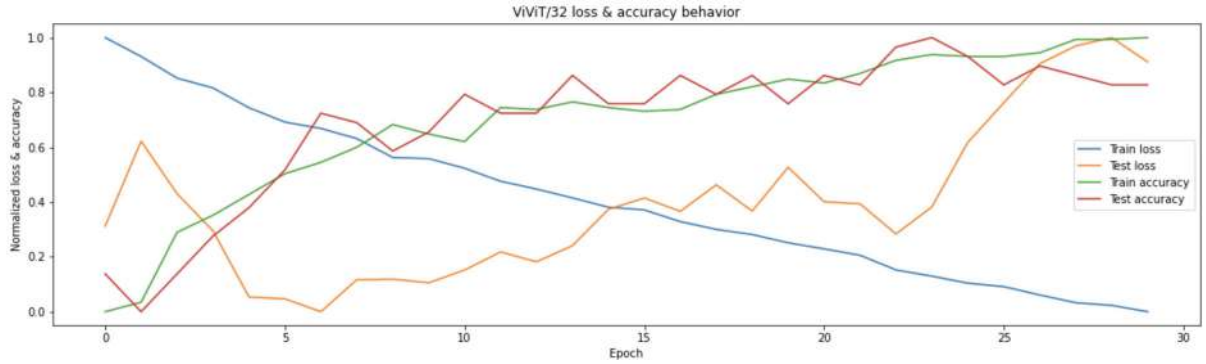


Figure 7: Train and test loss and accuracy comparison of ViViT/32 model on epochs scale

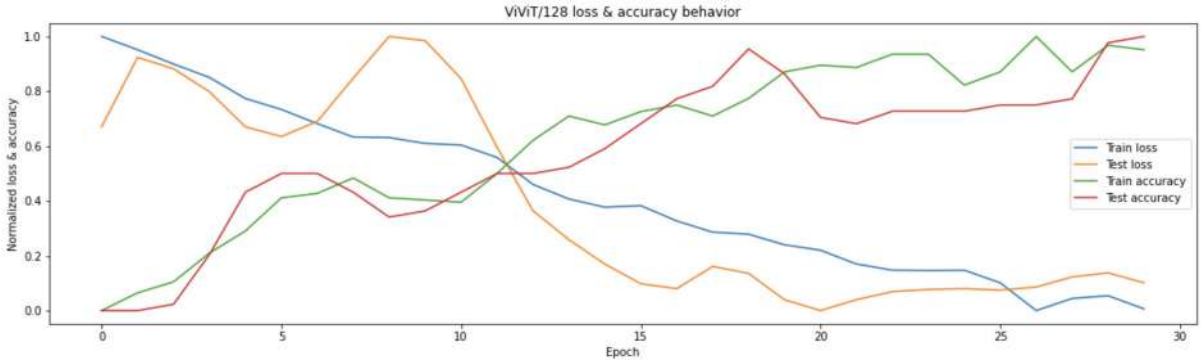


Figure 8: Train and test loss and accuracy comparison of ViViT/128 model on epochs scale

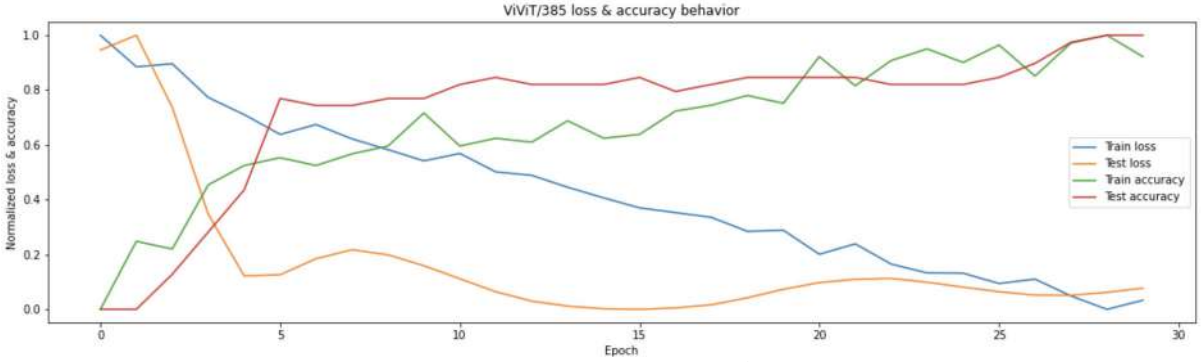


Figure 9: Train and test loss and accuracy comparison of ViViT/385 model on epochs scale

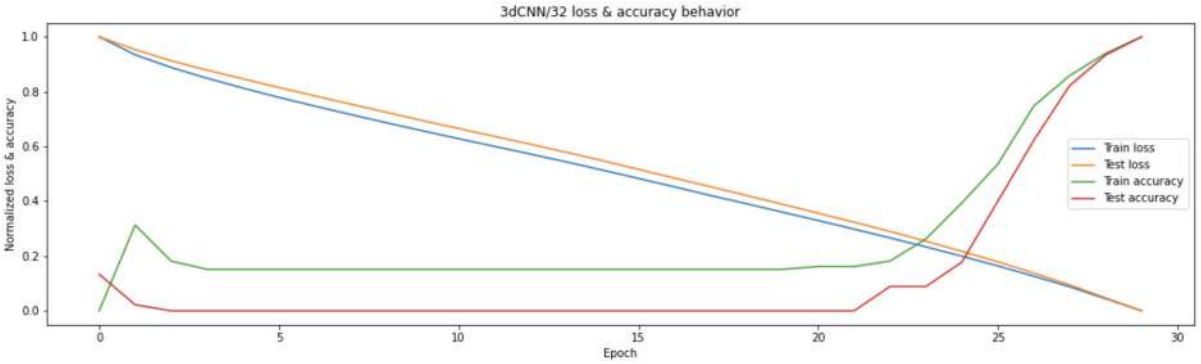


Figure 10: Train and test loss and accuracy comparison of 3dCNN/32 model on epochs scale

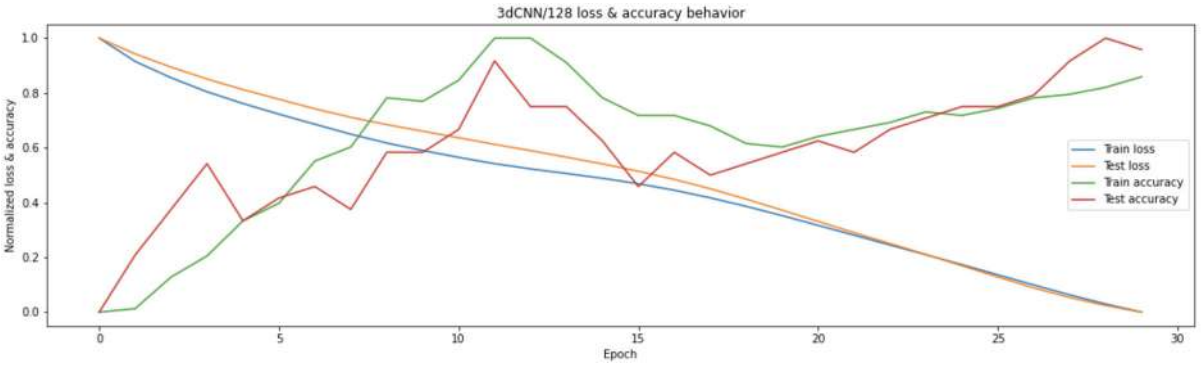


Figure 11: Train and test loss and accuracy comparison of 3dCNN/128 model on epochs scale

6. Conclusion

In this paper we discussed the most relevant practices and approaches for sign language recognition. While significant progress has been made in sign language recognition using modern methods, there are still some important issues that remain unsolved:

1. Large variability in sign language. Sign language can vary widely across different regions, cultures, and even individuals. This variability poses a significant challenge for sign language recognition systems, which must be robust to these variations.
2. The availability of large, diverse datasets is crucial for training and evaluating machine learning models for sign language recognition. However, there is still a limited availability of such datasets, particularly for less widely spoken sign languages.
3. Real-time sign language recognition is important for many applications, such as assistive technology and communication. However, real-time recognition remains a challenge, as it requires processing sign language videos in real-time, which can be computationally intensive.
4. Sign language gestures can be occluded or noisy due to factors such as clothing, lighting, and background clutter. Handling these occlusions and noise is still a challenge for sign language recognition systems.
5. Sign language recognition systems are typically trained on a limited set of sign language gestures, which can impact their ability to recognize new or rare signs.

The task of sign recognition in this paper was solved for real dataset. It was shown how to implement the existed approaches on different sizes of the existed data and what to do to receive the higher accuracy. Our experiments aimed to compare the performance of the Vision Transformer (ViViT) and 3D Convolutions CNN (3dCNN) models on sign language recognition. We trained both models on different batch sizes and evaluated their accuracy using the Top-K metric. Our analysis shows that ViViT models with small batch sizes achieved higher quality, but were more prone to overfitting. The best obtained numerical results were 69,7% for ViViT/3 on test-top1 and 89,7% on test-top2 and for ViViT/32 were achieved the accuracy 89,7% on test-top1 and 90,3% for test-top 2. To address the issue of overfitting, future investigations should explore approaches such as data augmentation to improve the generalization of ViViT models, especially when working with limited amounts of data.

The practical value of this paper is that it was also shown on real dataset how and why it is needed to search the compromise between speed, accuracy and overfitting issues as well as between length of the dataset and how existed methods needed to improve.

Overall, our results provide valuable insights into the strengths and limitations of different models for sign language recognition, as well as their practical implementation. It was offered in the paper a starting point for further research for sign language recognition by using vision transformers and additional approaches in conjunction with, for example, pose estimation, hands recognition, etc., with vision transformers before classification itself.

7. References

[1] W. Hameed and A.A. Al-Jumaily, "Sign language recognition: Dataset and challenges", in Proceedings of the 2019 International Conference on Innovations in Intelligent Systems and Applications, pp. 1-7, 2019.

[2] Sehili, M. E. A., & Melkemi, M. (2021). Challenges and Opportunities in Sign Language Recognition. IEEE Access, 9, 52470-52487. doi:10.1109/ACCESS.2021.3062127.

[3] V. Murino, C. S. Regazzoni, Template Matching Techniques in Computer Vision: Theory and Practice, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 743-760. doi:10.1109/34.85661.

[4] Dong, L., Jiang, S., Huang, Q., & Li, W. (2010). Template matching-based human action recognition in videos. Pattern Recognition, 43(3), 1199-1206. doi:10.1016/j.patcog.2009.07.022.

[5] Makris, D., & Ellis, T. (2006). Sign language recognition using hidden Markov models. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 36(3), 514-524. doi:10.1109/TSMCB.2005.856082.

[6] Garg, G., Sharma, S., & Saraswat, M. (2016). Sign Language Recognition Using Hidden Markov Models and HOG Features. In 2016 International Conference on Signal Processing and Communication (ICSC) (pp. 717-722). IEEE. doi:10.1109/ICSC.2016.7953781.
[7] Kumari, N., Gupta, N., & Sharma, S. K. (2016). A Review on Hidden Markov Models and Support Vector Machines in Sign Language Recognition. International Journal of Computer Applications, 138(7), 6-12. doi:10.5120/ijca2016908784.
[8] El-Fishawy, Z., Rizk, M., & Abdel-Wahab, M. A. (2018). Sign Language Recognition Using Kinect Sensor: A Review. In 2018 11th International Conference on Developments in eSystems Engineering (DeSE) (pp. 191-196). IEEE. doi:10.1109/DeSE.2018.00042.
[9] Guo, X.-L., & Yang, T.-T. (2016). Gesture recognition based on HMM-FNN model using a Kinect. Journal on Multimodal User Interfaces, 11. doi:10.1007/s12193-016-0215-x.
[10] Xia, W., Zhai, X., & Liu, Y. (2019). Sign language recognition using deep learning models: A review. ACM Transactions on Accessible Computing, 12(3), 1-30. doi: 10.1145/3301417.
[11] Zhang, C., Yang, J., & Kim, G-M. (2017). Sign Language Recognition with Convolutional Neural Networks Trained on Synthetic Data. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). doi: 10.1109/AVSS.2017.807856.
[12] Keze Wang, Xiaoguang Zhao, and Jing Liu. Sign Language Recognition Using Temporal Convolutional Networks and Skeleton Data. IEEE Access, vol. 7, pp. 158074-158083, 2019. doi: 10.1109/ACCESS.2019.2951043.
[13] R. Girdhar, G. Gkioxari, L. Torresani, and M. Paluri, "PoseTCN: Efficient Convolutional Neural Networks for Human Pose Estimation and Action Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 3332-3341. doi: 10.1109/CVPR.2019.00344.
[14] Deng, Z., Wan, J., & Xie, X. (2020). PoseTGCN: A Temporal Graph Convolutional Network for 3D Human Pose Forecasting. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 67-77. doi:10.1109/TNNLS.2020.3010193.
[15] Andrea Vanzo, Carlo Ciliberto, and Alberto Montagner, "Sign Language Recognition Based on Pose Estimation with Temporal Graph Convolutional Networks," Sensors 20(13), 3759, July 2020. doi:10.3390/s20133759.
[16] Kardoostsiami, A., Shafiee, M. J., & Plataniotis, K. N. (2021). Sign Language Recognition with Inflated 3D Convolutional Networks. IEEE Transactions on Multimedia, 23, 313-326. doi:10.1109/TMM.2020.3043619.
[17] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2015). Learning Spatiotemporal Features with 3D Convolutional Networks for Gesture Recognition. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4489-4497). doi: 10.1109/ICCV.2015.510.
[18] Efstratios Gavves, Thomas van de Weijer, and Jan C. van Gemert. Transferring Knowledge from Text to Sign Language Video Recognition with Transformers. 2021. doi: 10.1109/CVPRW50498.2021.00443.
[19] Chung, J.S., Kim, J., & Kim, J. (2021). Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. arXiv preprint arXiv:2103.01197.
[20] S. Kondratiuk, I. Krak, V.A. Kuznetsov, A. Kulias, Using the Temporal Data and Three-dimensional Convolutions for Sign Language Alphabet Recognition, CEUR Workshop Proceedingsthis link is disabled, 2022, 3137, pp. 78–87.
[21] S. Kondratiuk, I. Krak, A. Kylias, V. Kasianiuk. Fingerspelling Alphabet Recognition using Cnns with 3d Convolutions for Cross Platform Applications. Advances in Intelligent Systems and Computing. Vol. 1246 AISC. 2021, pp.585-596. doi:10.1007/978-3-030-54215-3_37.
[22] Ronchetti, F., Quiroga, F., Estrebou, C., Lanzarini, L., and Rosete, A. LSA64: A Dataset of Argentinian Sign Language. In XXII Congreso Argentino de Ciencias de la Computación (CACIC), 2016.
[23] Heng Wang, Yang Wang, and Gang Zeng, “Sign Language Recognition Using 3D Convolutional Neural Networks,” in Proceedings of the 24th ACM international conference on Multimedia (MM ’16), Amsterdam, The Netherlands, October 15-19, 2016, pp. 1033–1042. doi:10.1145/2964284.2964318.

[24] Lu, C., Chen, Y., & Lu, H. (2019). Sign Language Recognition Using 3D Convolutional Neural Networks with Softmax Probability Map. IEEE Access, 7, 116256-116267. doi: 10.1109/ACCESS.2019.2937045.
[25] Yuan, L., Chen, Y., Wang, T., Gan, W., Liu, Z., & Kornilov, S. (2021). ViViT: A Video Vision Transformer for Efficient Video Recognition. arXiv preprint arXiv:2103.15691.
[26] Elnaggar, A., Mahmoud, A., Abdou, A., Elgammal, A., & Abdel-Razek, M. (2021). ViViT-Sign: A Video Vision Transformer for Sign Language Recognition. arXiv preprint arXiv:2104.07441.

Neuroevolution methods for organizing the search for anomalies in time series

Serhii Leoshchenko[a], Andrii Oliinyk[a], Sergey Subbotin[a], Matviy Ilyashenko[a] and Tetiana Kolpakova[a]

[a] *National University "Zaporizhzhia Polytechnic", Zhukovskogo street 64, Zaporizhzhia, 69063, Ukraine*

Abstract

This paper is devoted to the problem of detecting and classifying anomalies for time series data. Some of the important applications of time series anomaly detection are healthcare, fraud detection, and system failure recognition.

Despite scientists extensive experience in detecting anomalies in time series, most methods look for individual objects that differ from ordinary objects, but do not take into account the specifics of the data sequence [1].

In this paper, a method for detecting anomalies and clastering time series based on neuro-evolutionary approaches is proposed. Prediction-based methods are used to detect anomalies: statistical and deep neural networks are used. Classical clustering methods that accept statistical parameters of series were used for clustering.

Keywords 1

Forecasting, time series, clustering, neuroevolution, genetic method

1. Introduction

Effective operation of complex technological systems requires monitoring and various analytical methods that allow you to monitor, manage, and preemptively change parameters. Monitoring is usually provided by standard tools (in most cases, a fairly reliable data collection and visualization system) [2, 3]. But creating effective analytical tools requires additional research, experiments, and a good knowledge of the subject area. As a rule, there are four main types of data analytics [4]:

- descriptive analytics visualizes accumulated data, including transformations for clarity and interpretability. Descriptive analytics is the simplest type of analysis, but also the most important one for applying other analysis methods;
- with the help of diagnostic analytics, they investigate the cause of events in the past, while identifying trends, anomalies, the most characteristic features of the described process, looking for causes and correlations (relationships);
- predictive analytics predicts likely outcomes based on identified trends and statistical models derived from historical data;
- administrative analytics allows you to get the optimal solution to a production problem based on predictive analytics. For example, it can be optimizing the operation parameters of equipment or business processes, or a list of measures to prevent an emergency.

Modeling methods, including machine learning, are usually used for predictive and prescriptive analytics. The effectiveness of these models depends on the high-quality organization of data

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: sergleo.zntu@gmail.com (S. Leoshchenko); olejnikaa@gmail.com (A. Oliinyk); subbotin@zntu.edu.ua (S. Subbotin); matviy.ilyashenko@gmail.com (M. Ilyashenko); 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); 0000-0003-4624-4687 (M. Ilyashenko); 0000-0001-8307-8134 (T. Kolpakova)

collection, processing, and preliminary analysis. The listed types of analytics differ both in the complexity of the models used and in the degree of human participation [2-6].

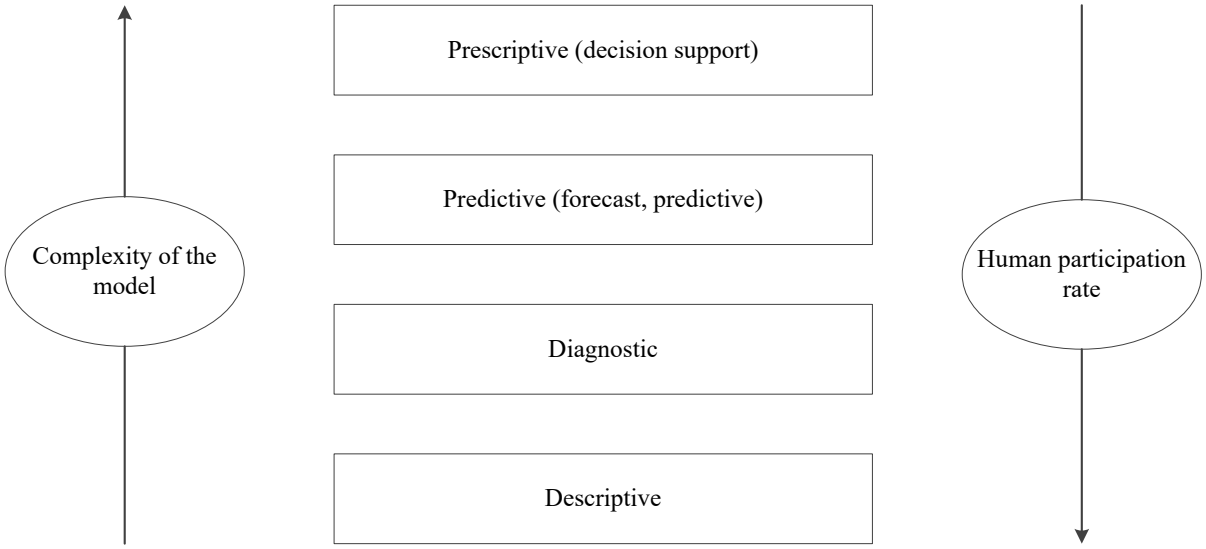


Figure 1: Comparison of model differences based on human participation and the complexity of the models themselves

There are a lot of areas of application of analytics tools-information security, banking sector, public administration, medicine and many other subject areas. Often, the same method works effectively for different subject areas, so developers of analytics systems create universal modules containing different algorithms [5].

For many technological systems, monitoring results can be represented as time series. The properties of the time series are [6]:

- linking each measurement (sample, discrete) to the time of its occurrence;
- equal time distance between measurements;
- ability to restore the behavior of the process in the current and subsequent periods from data from the previous period.

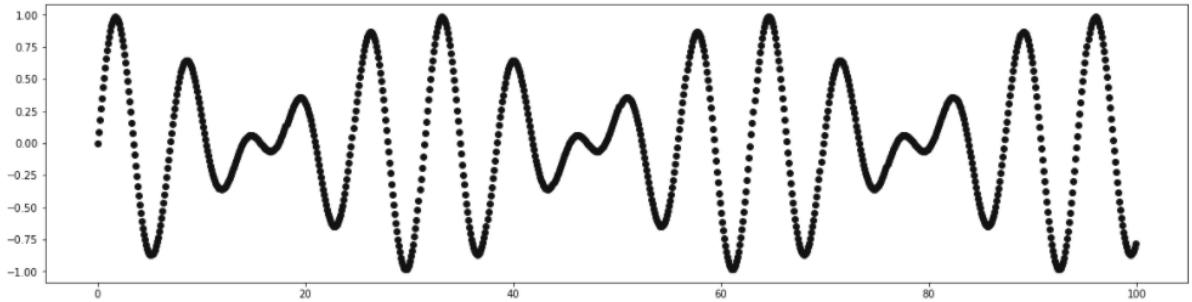


Figure 2: Example of a time series

Time series can describe more than just numerically measurable processes. The use of various methods and model architectures, including deep neural networks, allows you to work with data from natural language processing, computer vision tasks, etc. For example, a chat message can be converted into numeric vectors (embedding) that appear sequentially at a certain time, and the video is nothing more than a matrix of numbers that changes over time [7].

So, time series are very useful for describing the operation of complex devices and are often used for typical tasks: modeling, forecasting, feature selection, classification, clustering, pattern search, anomaly search [6]. Examples of such usage include an electrocardiogram, changes in the value of

shares or currency, weather forecast values, changes in network traffic, engine operation parameters, and much more.

Time series have typical characteristics that accurately describe the nature of the time series [2-6]:

- period: a time interval of constant length for the entire series, at the ends of which the series takes similar values,
- seasonality: periodicity property (season the same as period),
- cycle: characteristic changes in a series related to global causes (for example, cycles in the economy), there is no constant period,
- trend: a trend towards a long-term increase or decrease in the values of a series.

Time series may contain anomalies. An anomaly is a deviation in the standard behavior of a process. The method of machine search for anomalies uses data about the operation of the process (data sets) [7]. Depending on the subject area, there may be different types of anomalies in the dataset. It is customary to distinguish between several types of anomalies [8].

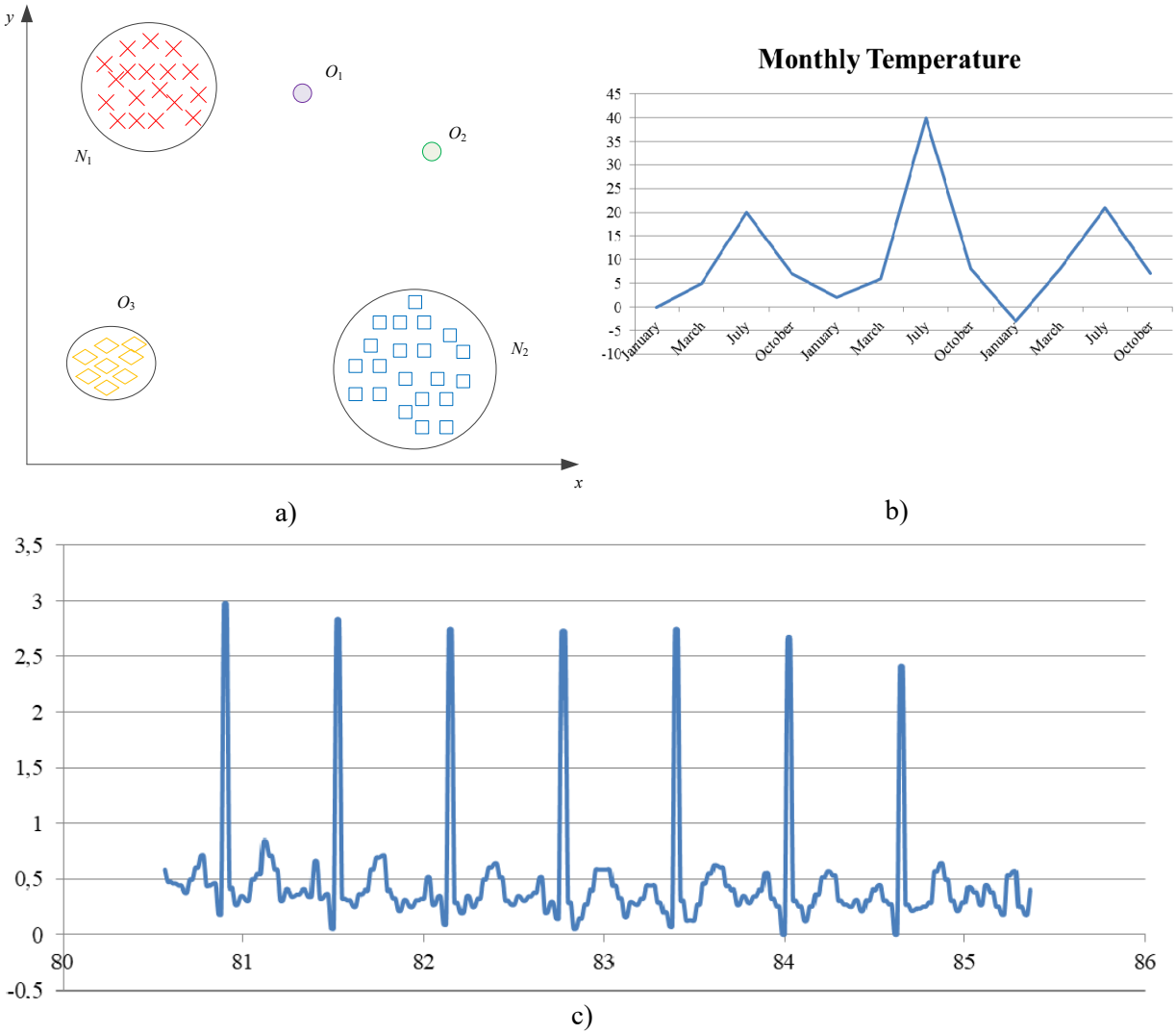


Figure 3: Examples of various anomalies in time series: a) point O_1 and O_2 and collective O_3 anomalies in two-dimensional data; b) contextual anomaly in the time series; c) collective anomaly in the time series (ECG): arrhythmia

1. Point anomalies. They occur in situations where a single instance of data can be considered as absolutely abnormal in relation to others [8].
2. Contextual anomalies. They are observed if the instance is abnormal in a certain context, or when a certain condition is met (therefore also called conditional) [8].

3. Collective anomalies. Occur when a sequence of related data instances (for example, a time series graph) is abnormal relative to the rest of the data [8].

An individual instance may not be a deviation, but the joint appearance of such instances will be a collective anomaly.

1.1. Anomaly detection strategies

Detecting point anomalies often requires some kind of system model. If the system does not have a deterministic mathematical model or such a model is too difficult to build, then a statistical model must be available. Depending on the method of constructing a statistical model, the following approaches are distinguished [8-10].

1. Recognize anomalies in case supervised learning. This technique requires a full-fledged training sample, including enough representatives of the normal and abnormal classes of values.

The technique is applied in 2 stages: first, training takes place on data that manually indicates normal and abnormal points. Then recognition occurs when new data is classified based on the constructed model [8-10].

It is usually assumed that the statistical properties of the model do not change over time, and such a change often requires repeated training.

The main difficulty of such methods is the formation of data for training. In addition to the obvious labor costs, often the abnormal class is also worse represented than the normal one, which can lead to inaccuracies in the resulting model [8-10].

2. Recognition of anomalies partially supervised learning. Similar to the previous one, but the training data represents only a normal class. A system trained in a normal class can determine whether data belongs to it, thus identifying abnormal data by exclusion.

3. Recognition in case unsupervised learning. In the absence of a priori information, this is the only possible option. Recognition in unsupervised learning: free mode is based on the assumption that abnormal data is quite rare. Therefore, only those that are farthest from the average values are indicated as anomalous [8-10]. Applying this technique to streaming data is difficult because it is necessary to have an idea of the entire data array to have a good estimate of the average and expected deviations.

1.2. Anomaly recognition methods

Classification. This method is based on the fact that the normal behavior of a system can be determined by one or more classes. Then an instance that does not belong to any of the classes is anomalous. This method usually uses the “partially supervised learning” approach. Basic mechanisms: neural networks, Bayesian networks, rule-based reference vector method [7, 9].

Clustering. This method is based on grouping similar values into clusters, and does not require knowledge of the properties of possible deviations. Anomaly detection is based on the following assumptions [8, 10]:

- normal data instances belong to a cluster
- normal data is closer to the center of the cluster, abnormal data is further away
- normal data forms large dense clusters, while abnormal data forms small and scattered ones.
- one of the simplest clustering methods is the *k*–means algorithm.

Statistical analysis. Using this approach, a statistical model of the process is constructed, which is then compared with the actual behavior. If the actual behavior differs from the model by more than a certain threshold, it is concluded that there are anomalies. Statistical analysis methods are divided into two groups [7]:

- parametric methods. Normal data is assumed to have a probability density function $\rho(x,\theta)$, where θ is the parameter vector, x is the data instance (observation);
- nonparametric method. The model structure is not defined a priori, but is determined from the data provided.

Nearest neighbor method. When using this technique, a metric is introduced (a measure of similarity between objects). Then two approaches are possible [10]:

- distance to the k^{th}-nearest neighbor. Abnormal data is the most distant from all other data;
- using relative density. Samples in areas with low relative density are evaluated as abnormal. (e.g. local emission level method).

Spectral methods. Based on the spectral (frequency) characteristics of the data, a model is constructed that is designed to take into account most of the variability in the data.

Let's compare the existing methods and present the results as a table.

Table 1

Comparison of anomaly recognition methods

Method	Result	Strategy	Classification of anomalies
Classification	Tag	Supervised learning, partially supervised learning	Yes
Clustering	Tag	Supervised learning, partially supervised learning	No
Statistical analysis	Degree	Partially supervised learning	No
Nearest neighbor	Degree	Unsupervised learning	No
Spectral methods	Tag	Unsupervised learning, partially supervised learning	No

In general, from the comparison of methods, we can conclude that there is an insufficient qualitative level of existing methods, because most methods are not able to use all strategies (approaches) to machine learning, and also from the results they do not provide an unambiguous answer about the assessment of detected anomalies. Moreover, most papers note an insufficient level of accuracy when using cool methods with newer model topologies. That is why the task of developing new approaches and methods for detecting anomalies in time series remains an urgent task.

2. Related Works

Point anomalies are most easily recognized-these are individual points where the behavior of the process differs dramatically from other points. For example, you can observe a sharp deviation of parameter values at a particular point [7-10].

Such values are called “outliers”, they strongly influence the statistical indicators of the process and are easily detected by setting thresholds for the observed value.

It is more difficult to detect an anomaly in a situation where the process behaves “normally” at each point, but collectively the values at several points behave “strangely”. Such abnormal behavior can include, for example, a change in the waveform, a change in statistical indicators (mean, mode, median, variance), the appearance of a mutual correlation between two parameters, small or short-term abnormal changes in the amplitude, and so on. And in this case, the task is to recognize abnormal behavior of parameters that cannot be detected by conventional statistical methods [7-10].

Finding anomalies is very important. In one situation, data should be cleared of anomalies in order to get a more realistic picture, while in another situation, anomalies should be carefully examined, as they may indicate a possible rapid transition of the device to emergency mode.

Finding anomalies in time series is not easy (unclear definition of anomalies, lack of markup, non-obvious correlation). So far, the Self Organizing Tree Algorithm (SOTA-algorithms) for searching for anomalies in time series has a high level of False Positive [7-10].

Only a small number of anomalies, mostly point-based, can be detected manually if good data visualization is available. Group anomalies are harder to detect manually, especially when it comes to large amounts of data and analyzing information about multiple devices. Also difficult to detect is the case of an "anomaly in time", when a normal signal appears at the "wrong" time. Therefore, when searching for anomalies in time series, it is advisable to use automation methods [7-10].

A big problem in finding anomalies on real data is that the data is usually not marked up, so initially it is not strictly defined what an anomaly is, there are no rules for searching. In such situations, it is necessary to apply methods of teaching without a teacher (unsupported learning), while models independently determine the relationships and characteristic laws in the data.

The methods used to search for anomalies in time series are usually divided into groups [7-10]:

- proximity-based: anomaly detection based on proximity parameter information or a sequence of fixed-length parameters, suitable for detecting point anomalies and outliers, but will not detect changes in the waveform;
- prediction-based: build a forecast model and compare the forecast and actual value, best applied to time series with pronounced periods, cycles, or seasonality;
- reconstruction-based: methods based on Data Fragment reconstruction use data fragment recovery (reconstruction), so it can detect both point anomalies and group anomalies, including changes in the waveform.

Proximity-based methods are focused on finding values that significantly deviate from the behavior of all other points. The simplest and most obvious example of implementing this method is monitoring whether a given threshold of values is exceeded [8].

In prediction-based methods, the main task is to build a qualitative process model in order to simulate the signal and compare the obtained simulated values with the original (true) ones. If the predicted and true signal are close, then the behavior is considered "normal", and if the values in the model are very different from the true ones, then the behavior of the system in this area is declared abnormal [9].

The most common time series modeling methods are SARIMA [11] and periodic neural networks.

The original approach is used in reconstruction-based models - first, the model is taught to encode and decode signals from an existing sample, while the encoded signal has a much smaller dimension than the original one, so the model has to learn to "compress" information. An example of such compression for 32-by-32-pixel images is given in [12].

After training, the models give input signals that are segments of the time series under study, and if encoding and decoding is successful, then the behavior of the process is considered "normal", otherwise the behavior is declared abnormal.

One of the newly developed reconstruction-based methods that show good results in detecting anomalies is TadGAN [13-15], developed by MIT researchers in late 2020. The TadGAN Method Architecture contains elements of an auto-encoder and Generative Adversarial Networks.

The network ε acts as an encoder that translates time series segments x into hidden space vectors z, and ς is a decoder that recovers time series segments from the hidden z representation. C_x is the critic who evaluates the recovery quality of $\varsigma(\varepsilon(x))$, and C_z is the critic who evaluates the similarity of the hidden representation of $z = \varepsilon(x)$ to white noise. In addition, there is a control of the "similarity" of the original and restored samples using the L2 measure according to the "Cycle consistency loss" ideology (which ensures the overall similarity of the generated samples to the original samples in GAN) [13]. The final objective function is a combination of all metrics for evaluating the quality of work of critics C_x, C_z and a measure of similarity between the original and restored signal.

$$\min_{\{\varepsilon,\varsigma\}} \max_{\{C_x \in C_x, C_z \in C_z\}} V_X(C_x,\varsigma) + V_Z(C_z,\varepsilon) + V_{L2}(\varepsilon,\varsigma). \quad (1)$$

To create and train a neural network, you can use various standard packages (for example, TensorFlow or PyTorch) that have a high-level API. An example of implementing a TadGAN-like architecture using the TensorFlow package for weight training can be found in the repository [14, 15]. When training this model, five metrics were optimized:

- aeLoss is the root-mean-square deviation between the original and restored time series, i.e. the difference between x and $\varsigma(\varepsilon(x))$;
- cxLoss-binary cross-entropy critique of C_x, which determines the difference between a true Time Series segment and an artificially generated one,
- cx_g_loss-binary cross-entropy, oscillator error $\varsigma(\varepsilon(x))$, which characterizes its inability to "deceive" the critic C_x,
- czloss-binary cross-entropy critique of C_z, which determines the difference between the hidden vector generated by the encoder and white noise, provides similarity of the hidden vector ‡ (x) to a random vector, preventing the model from "memorizing" individual patterns in the source data,
- cz_g_Loss is a binary cross-entropy, an error of the oscillator $\varepsilon(x)$, which characterizes its inability to create hidden vectors similar to random ones, and thus "deceive" the critic C_z.

After training the model, individual segments of the time series under study are reconstructed and the original and reconstructed series are compared, which can be performed using one of the following methods [13]:

- streaming comparison;
- comparison of curve areas in a given area around each sample (area length is a hyperparameter);
- Dynamic Time Warping.

Quality is evaluated using the f1 metric for the binary classification problem, "positive" (null hypothesis): there is an anomaly, "negative" (alternative hypothesis): no anomaly.

Table 2
Quality is evaluated using the F1 metric for the binary classification problem

	The anomaly is predicted by the model	The model predicted the absence of an anomaly
There is an anomaly	TP correctly predicted anomaly	FN there is an anomaly, but it was not found
There is no anomaly	FP predicted an anomaly where it doesn't exist	TN there is no anomaly and the model does not see it

To demonstrate the operation of the method, we use a synthetic (artificially generated) series without anomalies, which is the sum of two sinusoids, the values of which vary in the range from -1 to 1: $y(x)=\frac{1}{2}\sin(x)+\frac{1}{2}\sin(0.8x)$.

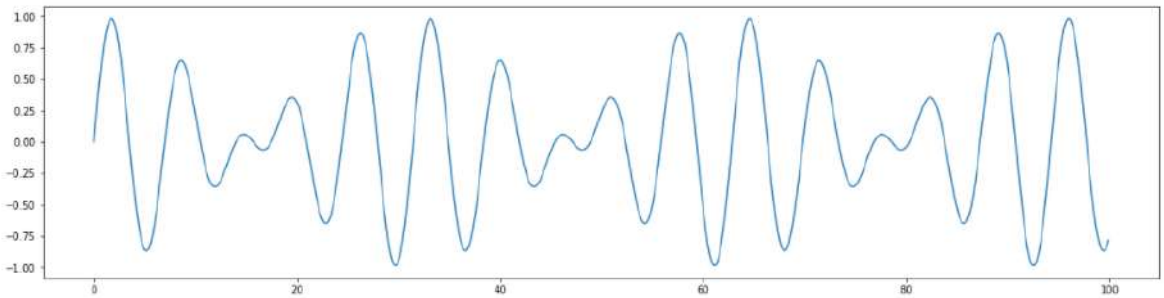


Figure 4: Graph of such a series

It can be seen that the model has quite accurately learned to predict the main patterns in the data. Let's try adding various anomalies to the data and then detect them using the tadgan model. First, let's add a few point anomalies [13].

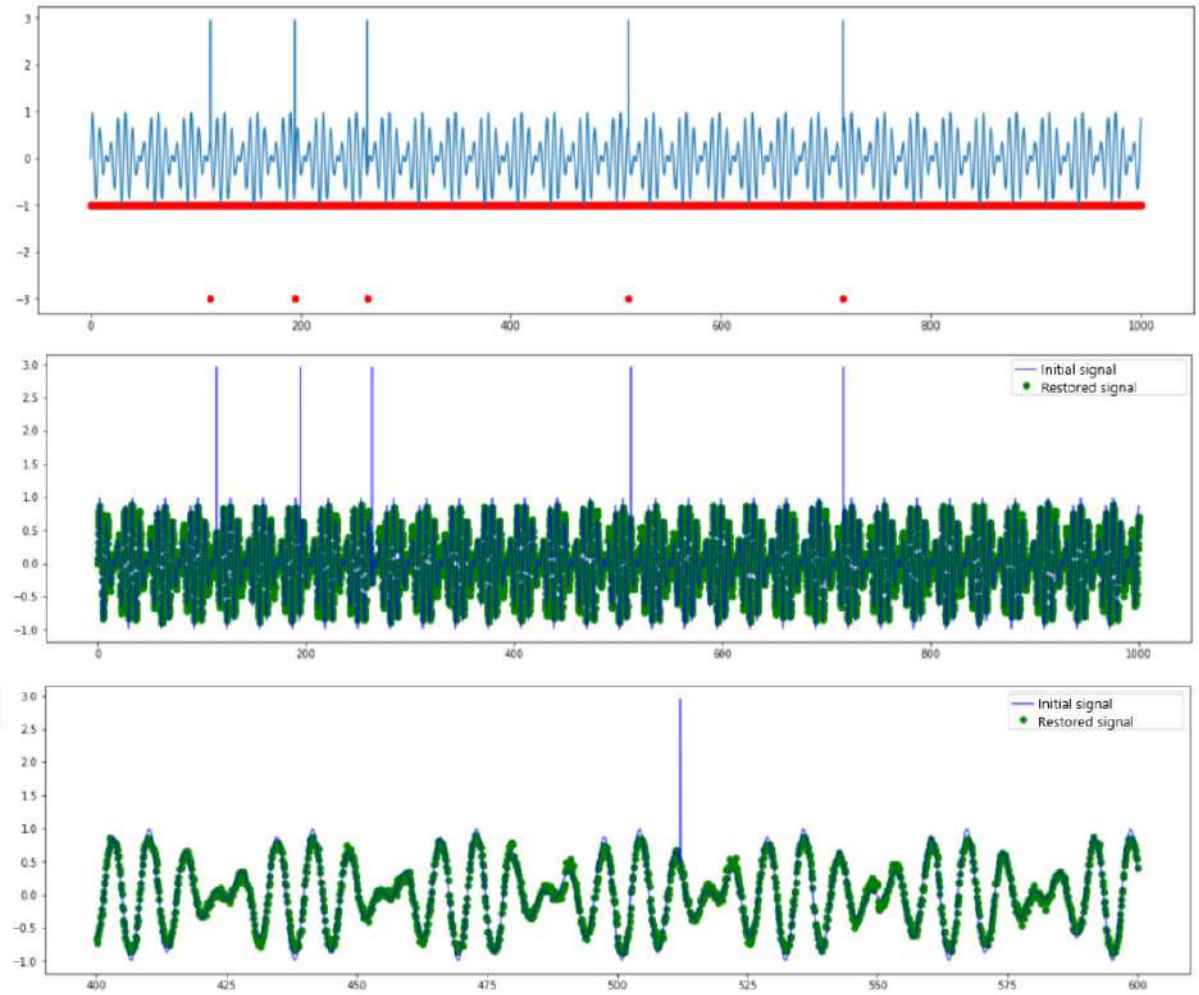


Figure 5: Spot Anomaly Detection using TadGAN

The graph of the original and predicted signals shows that the model cannot restore the “peaks” of anomalous values, which can be used with high accuracy to determine point anomalies. However, in such a situation, the crust of the complex TadGAN model is not obvious-such anomalies can also be detected by estimating the excess of thresholds [13].

Now consider a signal with a different type of anomaly: a periodic signal with an abnormal frequency change. In this case, there is no excess of the threshold: from the point of view of amplitude, all elements of the series are “normal” values, and the anomaly is detected only in the group behavior of several points. In this case, TadGAN also cannot restore the signal (as can be seen in the figure) and can be used as a sign of the presence of a group anomaly [13].

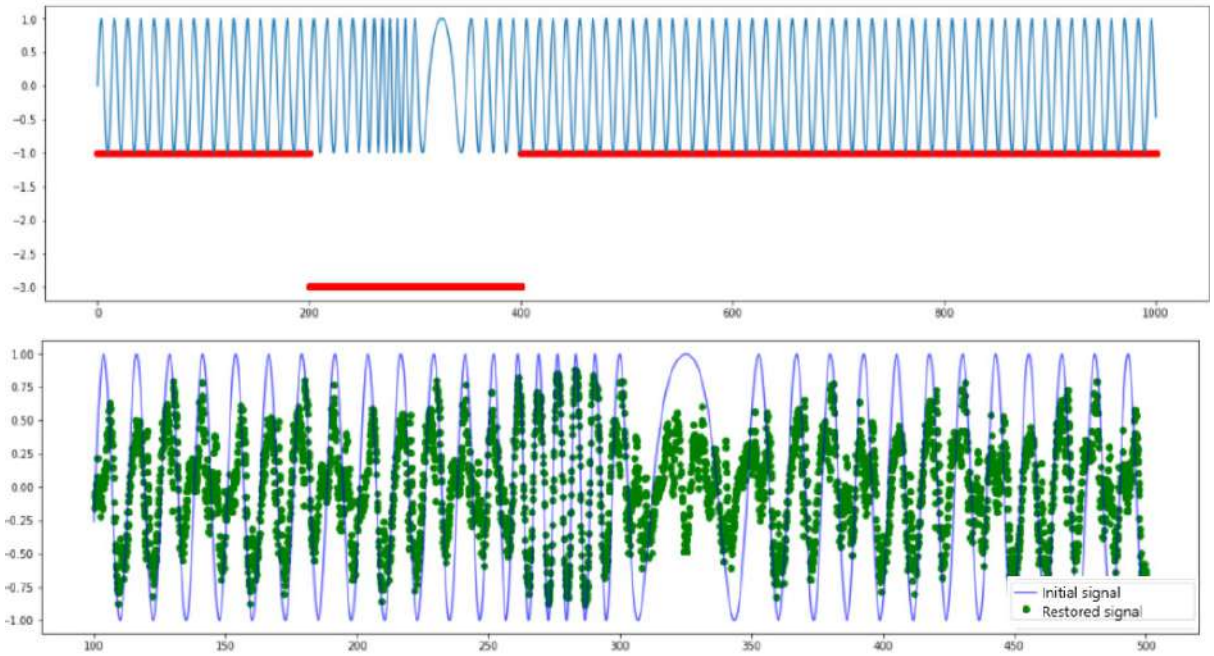


Figure 6: The result of the TadGAN operation on a dataset with an abnormal frequency change.

These two examples illustrate the work of the method. The reader can also try to create their own data sets and test the model's capabilities in various situations.

More complex examples of datasets can be found in the article by the authors of the TadGAN Method [13]. There is also a link to the Orion library, which is developed by MIT specialists, which uses machine learning to recognize rare anomalies in time series, using the unsupported learning approach [14].

However, most scientists agree that each case requires its own method of signal reconstruction and a conductive method of training the model, which significantly slows down the practical implementation.

3. Proposed method

The prototype of our solution is shown in Fig. 7, consists of two separate groups of the same population. The first is a set of models, and the second is a set of data individuals.

Each model is a sequence of encoder and decoder levels that describe the input multidimensional signal [16]. The framework allows you to create an ensemble model using an approach similar to the method based on packaging in packages. A whole ensemble model is a set of sub-models that work with subgroups of input signals. Ensemble models form a set.

The work of the solution begins with generating initial groups of input signals using correlation; the system then updates the models and groups using a genetic algorithm [17-19]. In parallel, genetic operators optimize individual models in each subgroup by making changes to the topology of neural models (for example, model length and layer parameters) [17-19]. The end result of these actions is an ensemble model optimized for anomaly detection. The ensemble model is defined as follows.

A number of papers note that despite different models, it is noticeable that almost all models show a similar set of anomalies, despite changes in their hyperparameters [16]. Of course, the results differed depending on the hyperparameters, but none of the changes significantly affected detection. Because of this, an ensemble model is proposed, based on the simultaneous division of the search space into subgroups and the evolution of models for such subgroups. As a result, the models were able to identify more specific dependencies and relationships between signals [20-22].

Models within each subgroup are optimized by changing their internal structure. The evolution of a single model is carried out in five main stages:

1. clustering the search space [23-25];

2. crossing, mutating, and selecting the best models for individual clusters;
3. syncing solutions;
4. multi-parent crossover [17-19];
5. evaluation of the ensemble solution.

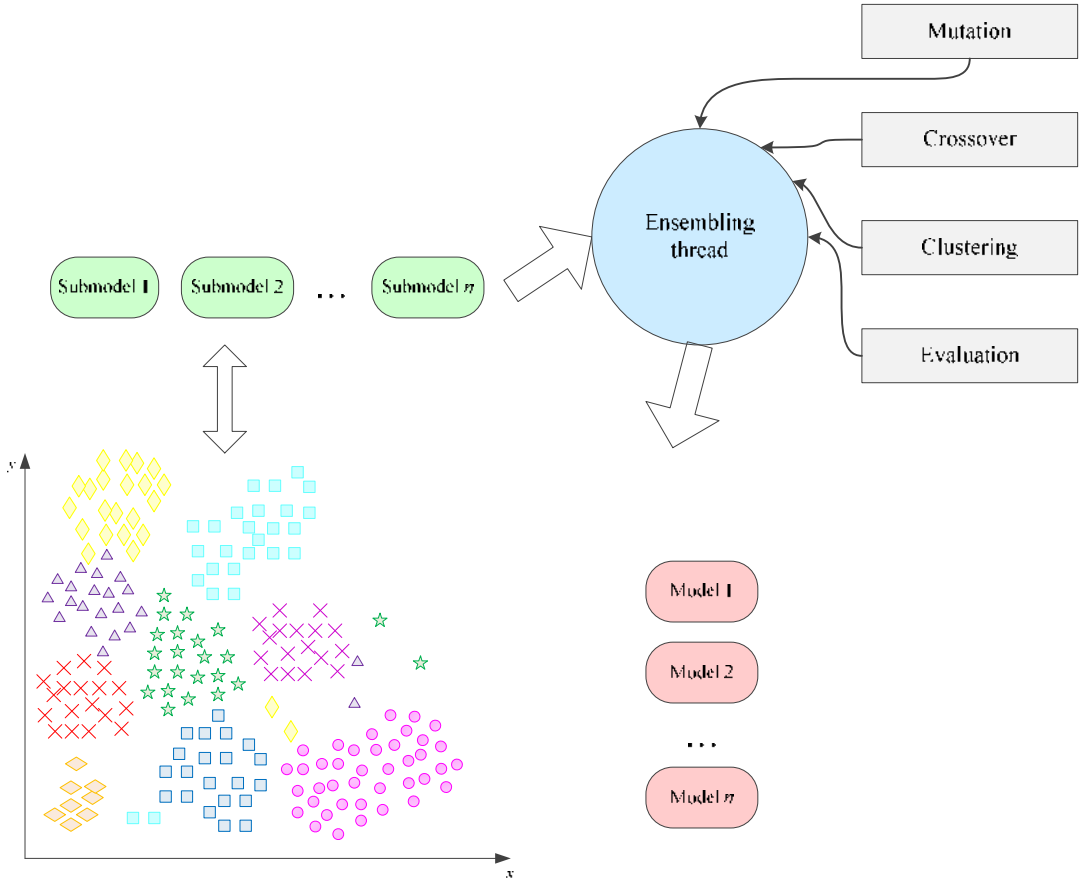


Figure 7: The proposed method

4. Experimental research

For the experimental research of proposed method was be used the following as the training and testing data:

- The Secure Water Treatment (SWaT) Dataset [26], [16]. This dataset contains data gathered from a scaled-down version of a real water treatment plant. The data were collected for 11 d in two modes – 7 d of normal operation of the plant and 4 d during which there were cyber and physical attacks executed;
- The Water Distribution (WADI) Dataset [27], [16]. This dataset contains data from a scaled-down version of a water distribution network in a city. The collected data contain 14 d of normal operation and 2 d during which there were 15 attacks executed.

The general information about datasets present in Table 3.

Table 3
General information about datasets

Datasets	Number of Input Signals	Number of Trainings	Number of Tests	Number of Anomalies
SWAT	51	49,668	44,981	11.97%
WADI-2017	123	1,048,571	172,801	5.99%
WADI-2019	123	784,571	172,801	5.77%

The meta-parameters for neuroevolution synthesis of models demonstrate at Table 4.

Table 4
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 changing the activation function
Clustering method	*k*-nearest neighbors
Neighbors number	7

The results of the work present at Table 5.

Table 5
The work results of proposed method

Datasets	Precision, %	Recall, %	f1, %
SWAT	94.41	55.35	0.74
WADI-2017	90.28	70.64	0.82
WADI-2019	89.53	71.47	0.83

5. Discussions of results

The experimental results demonstrate that parallel clustering of data and synthesis of a model based on the processed data using an ensemble system can significantly increase the efficiency of the anomaly detection process. The process of neuroevolution helps to synthesize models and develop them in stages based on updated information about input data, without separating the process of data preprocessing. Tests based on samples of WADI and SWAT data. In both cases, the results of anomaly detection demonstrated satisfactory values, raising the qualitative level of detection. The improvements in the WADI dataset were more significant than in the SWAT dataset because there are more sensors and samples in the WADI dataset than in the SWAT dataset.

The article proved that the neuroevolution approach can have a positive impact on the results. Our future work will focus on further improvements to the method. A combination of different solution topologies can significantly improve the results of the work.

6. Conclusion

At the paper a research and comparative analysis of existing strategies and methods solving the problem of detecting and classifying anomalies were carried out, and a detection method based on neuroevolutionary methods was also proposed. As can be seen from the results of the study, the method of detecting anomalies based on neuroevolution has shown much greater effectiveness. The proposed method has shown itself to be viable and can be improved.

The result of the work is not only a comprehensive study and theoretical justification of the theory associated with the analysis of time series, but also the proposed solution. His work consists of two

stages: the separation of the search space and the synthesis of models. At the training stage, the method processes and separates data about the behavior of the system. In the synthesis mode, the method gradually adjusts the models in order to get the final solution from them in the future. The resulting model is synthesized using uniform crossing, which makes it possible to increase the size of the parent pool from two individuals to a much larger number.

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" “Development of methods and tools for analysis and prediction of dynamic behavior of nonlinear objects” (state registration number 0121U107499) and “Intelligent methods and tools for diagnosing and predicting the state of complex objects” (state registration number 0122U000972).

8. References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Computing Surveys, 41(3) (2009) 1-58. doi: 10.1145/1541880.1541882.
[2] What Is Data Analytics?, 2020. URL: https://www.intel.com/content/www/us/en/artificial-intelligence/what-is-data-analytics.html#:~:text=Data%20analytics%20is%20the%20process,data%20for%20practically%20any%20purpose
[3] Cycle Consistency Loss, 2017. URL: https://paperswithcode.com/method/cycle-consistency-loss
[4] Search for anomalies in time series based on the estimation of their parameters, 2021. URL: https://openarchive.nure.ua/items/7c8e2e76-10fa-4044-907b-e51d05bd7cbf [In Ukrainian]
[5] J. Ma, S. Perkins, Time-series novelty detection using one-class support vector machines, in: Proceedings of the International Joint Conference on Neural Networks, Portland, OR, IEEE, 2003, pp. 1741-1745. doi: 10.1109/IJCNN.2003.1223670.
[6] MIT – Data to AI Lab, Time series anomaly detection — in the era of deep learning, 2020. URL: https://medium.com/mit-data-to-ai-lab/time-series-anomaly-detection-in-the-era-of-deep-learning-dccb2fb58fd
[7] Anomaly Detection in Time Series, 2023. URL: https://neptune.ai/blog/anomaly-detection-in-time-series
[8] A. Bhattacharya, Effective Approaches for Time Series Anomaly Detection, 2020. URL: https://towardsdatascience.com/effective-approaches-for-time-series-anomaly-detection-9485b40077f1
[9] S. Schmidl, P. Wenig, T. Papenbrock, Anomaly detection in time series: a comprehensive evaluation, in: Proceedings of the Proceedings of the VLDB Endowment, Vol. 15 (9), Sydney, ACM, 2022, pp. 1779–1797. doi: 10.14778/3538598.3538602.
[10] Anomaly detection and forecasting in Azure Data Explorer, 2023. URL: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/anomaly-detection
[11] B. Artley, Time Series Forecasting with ARIMA , SARIMA and SARIMAX, 2022. URL: https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6
[12] M. K. Mandal, Implementing PCA, Feedforward and Convolutional Autoencoders and using it for Image Reconstruction, Retrieval & Compression, 2018. URL: https://blog.manash.io/implementing-pca-feedforward-and-convolutional-autoencoders-and-using-it-for-image-reconstruction-8ee44198ea55
[13] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, K. Veeramachaneni, TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks, in: Proceedings of the IEEE International Conference on Big Data (Big Data), IEEE, 2020. doi: 10.1109/BigData50022.2020.9378139.
[14] TadGAN, 2021. URL: https://github.com/gusty1g/TadGAN

[15] Examples. TadGAN, 2022. URL: https://github.com/CyberLympha/Examples/tree/main/%D0%A0%D0%B0%D0%B7%D0%B1%D0%BE%D1%80%20%D1%81%D1%82%D0%B0%D1%82%D0%B5%D0%B9/TadGAN
[16] K. Faber, M. Pietron, D. Zurek, Ensemble Neuroevolution-Based Approach for Multivariate Time Series Anomaly Detection, Entropy, 23 (2021). doi: 10.3390/e23111466.
[17] S. Leoshchenko, S. Subbotin, A. Oliinyk, V. Lytvyn, M. Ilyashenko, Smart crossover mechanism for parallel neuroevolution method of medical diagnostic models synthesis, in: Proceedings of the Third International Workshop on Computer Modeling and Intelligent Systems, CMIS-2020, CEUR-WS, Zaporizhzhia, 2020, pp. 57-69.
[18] S. Leoshchenko, A. Oliinyk, S. Subbotin, Adaptive Mechanisms for Parallelization of the Genetic Method of Neural Network Synthesis, in: Proceedings of the 10th International Conference on Advanced Computer Information Technologies, ACIT 2020, Deggendorf, Ternopil, IEEE, 2020, oo. 446-450. doi: 10.1109/ACIT49673.2020.9208905.
[19] S. Leoshchenko, A. Oliinyk, S. Subbotin, T. Zaiko, S. Shylo, V. Lytvyn, Sequencing for encoding in neuroevolutionary synthesis of neural network models for medical diagnosis , in: Proceedings of the 3rd International Conference on Informatics & Data-Driven Medicine, IDDM 2020, Växjö, Lviv, CEUR-WS, 2020, pp. 62-71.
[20] J.A.J. Alsayaydeh, Irianto, M. Zainon, H. Baskaran, S. G. Herawan, Intelligent Interfaces for Assisting Blind People using Object Recognition Methods, International Journal of Advanced Computer Science and Applications(IJACSA), 13(5) (2022) 734-741. doi: 10.14569/IJACSA.2022.0130584.
[21] J.A.J. Alsayaydeh, Irianto, A. Aziz, C.K. Xin, A. K. M. Zakir Hossain, S.G. Herawan, Face Recognition System Design and Implementation using Neural Networks, International Journal of Advanced Computer Science and Applications(IJACSA), 13(6) (2022) 519-526. doi: 10.14569/IJACSA.2022.0130663.
[22] V. Shkarupylo, I. Blinov, A. Chemeris, J.A.J. Alsayaydeh, A. Oliinyk, Iterative approach to TLC model checker application, in: Proceedings of the 2nd KhPI Week on Advanced Technology, KhPI Week 2021, IEEE, Kyiv, 2021, pp. 283-287. doi: 10.1109/KhPIWeek53812.2021.9570055.
[23] R. Pawar, k-NN based Time Series Classification, 2021. URL: https://towardsdatascience.com/k-nn-based-time-series-classification-e5d761d01ea2
[24] S. Tajmouati, B. Wahbi, A. Bedoui, A. Abarda, M. Dakkon, Applying k-nearest neighbors to time series forecasting : two new approaches, 2021. URL: https://arxiv.org/abs/2103.14200
[25] F. Martínez, M.P. Frías, M. Pérez, A. J. Rivera Rivas, A methodology for applying k-nearest neighbor to time series forecasting, Artificial Intelligence Review, 52 (2019) 2019–2037. doi: 10.1007/s10462-017-9593-z.
[26] A. P. Mathur, N. O. Tippenhauer, SWaT: a water treatment testbed for research and training on ICS security, in: Proceedings of the International Workshop on Cyber-physical Systems for Smart Water Networks, CySWater, Vienna, IEEE, 2016, pp. 31-36, doi: 10.1109/CySWater.2016.7469060.
[27] C. M. Ahmed, V. R. Palleti, A. P. Mathur, WADI: a water distribution testbed for research in the design of secure cyber physical systems, in: Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER '17, Pittsburgh, PA, ACM, 2017, pp. 25-28. doi: 10.1145/3055366.3055375.

Multiclass Image Classification Based on Quantum-Inspired Convolutional Neural Network

Hamza Kamel Ahmed[a,b], Baraa Tantawi[a] and Gehad Ismail Sayed [a]

[a] *School of Computer Science, Canadian International College (CIC), New Cairo, Egypt*
[b] *Computer Science Department, School of Science and Technology, Troy University, USA*

Abstract

Multiclass image classification is considered a challenging task in computer vision that requires correctly classifying an image into one of the multiple distinct groups. In recent years, quantum machine learning has emerged as a topic of significant interest among researchers. Using quantum concepts such as superposition and entanglement, quantum machine learning algorithms provide a more efficient method of processing and classifying high-dimensional image data. This paper proposes a new image classification model using quantum-inspired convolutional neural network architecture or, shortly, QCNN. The proposed model consists of two main phases; pre-processing and classification based on the QCNN phase. Seven benchmark datasets with different characteristics are adopted to evaluate the performance of the proposed model. The experimental results revealed that the proposed QCNN outperformed its classical version. Additionally, the results demonstrated the effectiveness of the proposed model compared with the state-of-the-art models.

Keywords [1]

Quantum Computing, Convolutional Neural Networks, Image Classification, Quantum Machine Learning

1. Introduction

Convolutional neural networks (CNNs) have experienced rapid expansion in the image classification and multiclass image classification fields. Which computer vision task requires categorizing an image into one of many categories? As the number of classes increases, the task's difficulty increases, and it requires identifying fine-grained details that differentiate between distinct objects or classes. In recent years, convolutional neural networks (CNNs) have become a popular solution to this issue. The CNN architecture is designed to mimic the structure of the visual cortex in humans and animals. It consists of multiple layers of interconnected nodes trained to extract and identify various image features. The input image is fed to the CNN's first layer, and each successive layer extracts increasingly complex features. The class with the highest probability is selected as the predicted class based on the probabilities generated by CNN's output layer.

Although CNNs have been demonstrated to be an effective tool for image classification and other computer vision tasks, they may not always perform at their best with small datasets [12]. Quantum computing comes into play at this point. Using quantum concepts such as superposition and entanglement, quantum computing can potentially improve traditional machine-learning techniques [13]. Notably, it has been demonstrated that quantum machine learning algorithms provide superior feature selection capabilities than classical methods, enabling more efficient processing and classification of high-dimensional data [14]. Consequently, a growing interest has been in developing machine-learning models inspired by quantum mechanics, such as quantum-inspired CNNs [15].

[1] The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: hamzakamel.a@gmail.com (H. Ahmed); baraa.tantawi@gmail.com (B. Tantawi); gehad_sayed@cic-cairo.com (G. Sayed)
ORCID: 0000-0003-0895-8873 (H. Ahmed); 0000-0003-1045-579X (B. Tantawi); 0000-0001-9007-916X (G. Sayed)

These models aim to combine the power of convolutional neural networks (CNNs) with the benefits of quantum computing to improve performance on complex image classification tasks [16]. This paper proposes a new image classification model based on quantum computing principles, including entanglement and superposition, in conjunction with the CNN layer. The proposed model comprises two principal phases: pre-processing and the classification-based quantum convolutional neural network phase. The original image undergoes image resizing and data normalization during the pre-processing phase. The processed images are then used as input for the proposed QCNN architecture. This paper's primary contribution is summarized in the following points:

1. A new QCNN architecture is proposed.
2. The proposed QCNN is applied to the multiclass image classification problem.
3. Seven benchmark datasets are used to evaluate the proposed model.
4. A comparative analysis between the proposed model and state-of-the-art models is considered.

The remainder of the paper's format is as follows: Section 2 presents the related work; Section 3 introduces the overall proposed image classification model based on QCNN in detail; and Section 4 presents and discusses the results of experiments conducted on various datasets using the proposed QCNN architecture. Section 5 concludes with a summary of the paper's key findings, a discussion of the limitations of the proposed model, and suggestions for future research directions.

2. Literature review

CNNs have demonstrated efficacy in multi-class image classification by extracting higher-level image information and outperforming conventional image processing [1]. CNNs' ability to capture complex image characteristics has led to significant advances in computer vision tasks such as object recognition [4], semantic segmentation [6], and target detection [8]. Recent studies have proposed altering CNN's architecture to enhance its performance in multi-class image classification tasks [2]. These alterations are intended to improve the network's precision, reduce computational complexity, and accelerate the training procedure. These modifications include the addition of skip connections [3], the use of various activation functions [4], and the incorporation of attention mechanisms [5]. In numerous studies involving computer vision, CNNs have produced outstanding results. For instance, some papers [5, 6] focused on target detection. The authors of [5] created a CNN-based infrared dim small target detection algorithm that proved effective and precise for detecting small targets in infrared images with low SCR and complex scenes. They incorporated spatially finer, target-oriented shallow features and semantically more robust deep features.

In contrast, the authors of [6] presented a depth recognition algorithm for robot vision systems used for apple picking. Using deep learning techniques, the algorithm achieved greater recognition accuracy and robustness than conventional methods, demonstrating the potential of deep learning in agricultural robot applications. Other studies [7, 8] have investigated semantic segmentation. The authors of [7] proposed a multi-receptive-field convolutional neural network (MRFNet) to extract rich and useful context information from complex and dynamic medical images. On three public medical image datasets, including SISS, 3DIRCADb, and SPES, the MRFNet demonstrated exceptional performance. In the study [8], the authors introduced a novel network architecture for thermal image semantic segmentation called edge-conditioned convolutional neural network (EC-CNN). The end-to-end-trained EC-CNN generated high-quality segmentation results by incorporating prior edge knowledge. In addition, they presented a new benchmark dataset, Segmenting Objects in Day and Night (SODA), to facilitate exhaustive evaluations in thermal image semantic segmentation. In addition, CNN architectures have demonstrated their effectiveness in image classification, as demonstrated in [9, 10–11]. Regarding image classification, the authors of [9] proposed the dilated CNN and the HDC models, demonstrating improved training efficiency and accuracy on the MNIST dataset and a wide-band remote sensing image dataset. The research in [10] centered on classifying biological images using inverted residual blocks to replace some CNN modules to address the increased computational time. The method demonstrated promising performance online on five benchmark datasets, including two biological IM performances. Lastly, [11] sought to identify an appropriate architecture for transfer learning and identified Inception-v3 as such. The retrained

Inception-v3 model performed significantly better than previous state-of-the-art works on the CIFAR-10 dataset. This study's findings demonstrated the utility of transfer learning and paved the way for future developments in deep neural networks.

Researchers are investigating the potential benefits of combining CNNs with quantum computing techniques in light of the development of quantum computing. The training and application of quantum CNNs (QCNNs) have the potential to accelerate significantly. Quantum mechanics enables QCNNs to perform specific calculations significantly faster than their classical counterparts. As full-fledged quantum computation is still a distant prospect, the research in [13] discusses the potential of quantum-inspired (QI) algorithms implemented on classical computers to improve existing machine-learning techniques. Based on the theory of Quantum State Discrimination (QSD), they developed a QI algorithm for direct multi-class classification that provides a systematic method for locating sub-optimal solutions. This strategy enabled them to extend the capabilities of previous QI classifiers, which were restricted to binary classification, and address general multi-class datasets. In [14], the authors proposed a novel algorithm for feature selection based on a generalized embedding of Quadratic Unconstrained Binary Optimization (QUBO), which is executable on both classical and quantum hardware. They used mutual information as the basis for measures of importance and redundancy, with an interpolation factor serving as a counterbalance. Experiments comparing standard feature selection methods and their performance on various machine learning models demonstrated the effectiveness of the researchers' framework. In addition, they successfully executed one of their experiments on actual quantum hardware, demonstrating the algorithm's viability and compatibility with NISQ. Although their experiments were conducted on low-dimensional problems due to hardware constraints, the authors anticipate that the algorithm will scale with future quantum computing advancements.

3. The Proposed image classification based on QCNN model

The two primary phases of the proposed image classification model are the pre-processing phase and the classification based on the QCNN phase. The overall design of the proposed image classification model is depicted in Figure 1. From this figure, it can be seen that the original images are first normalized, after which the normalized images are fed into the quantum convolutional layer, after which a series of classical convolutional layers are applied, followed by a fully connected layer to obtain the final class. The following sections will go over each part's full description.

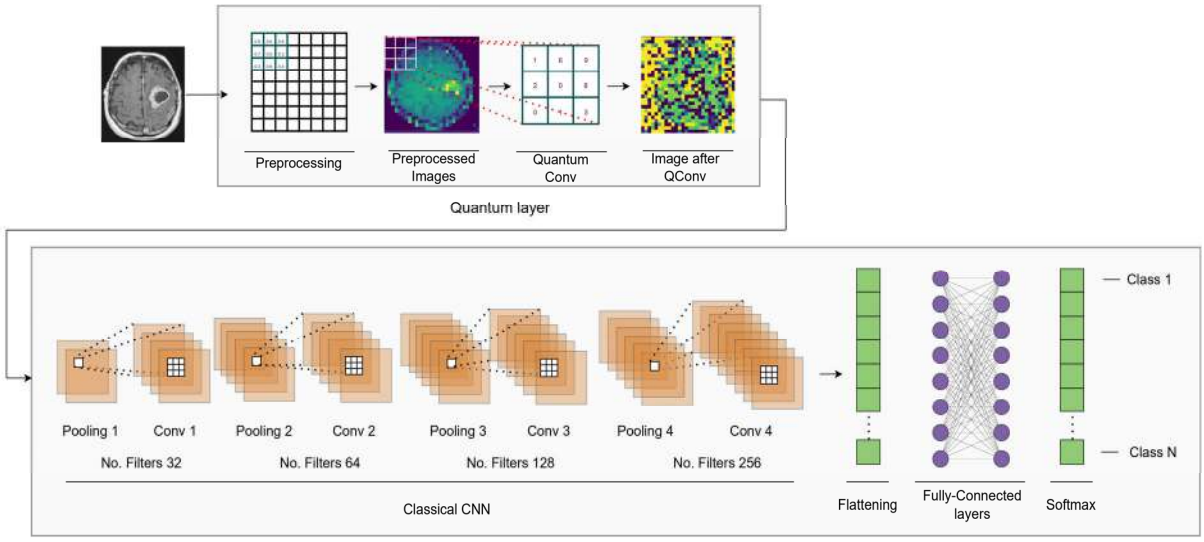


Figure 1: The architecture of the proposed image classification model

3.1. Pre-processing phase

Preparing and transforming the raw dataset into a more acceptable format for subsequent analysis and modeling is the primary goal of the pre-processing phase. The original images in this study are downsized to 28x28, and after that, data normalization is applied to all of the images in the dataset. Data normalization is crucial to improve the effectiveness and interpretability of machine learning algorithms. Normalization is a method that rescales the input features to a uniform range, usually between 0 and 1. This standardized range aids in reducing the influence of data distribution and scale differences, which could otherwise result in biased and less-than-ideal model performance. Each pixel in the image is normalized in this study by multiplying it by the highest number, 255. Each data point in the image is effectively scaled down to a value between 0 and 1 when this procedure is done to the entire dataset's image. The model is made to be less sensitive to the input characteristics' absolute values and more focused on the underlying patterns and relationships in the data by doing this normalization.

3.2. Classification based on QCNN phase

This phase feeds the processed image into the quantum convolutional neural network (QCNN). The quantum and classical components of the proposed QCNN make up its two main components. The proposed creates a powerful and effective classification system by combining the advantages of both quantum and conventional methodologies. Next, a detailed description of each part is presented.

3.2.1. Quantum part

The quantum part transforms the input data using the capabilities of quantum computing as a pre-processing layer. Quantum encoding, convolution, and measurement comprise its three main components. The block diagram for the quantum part of the proposed QCNN architecture is shown in Figure 2.

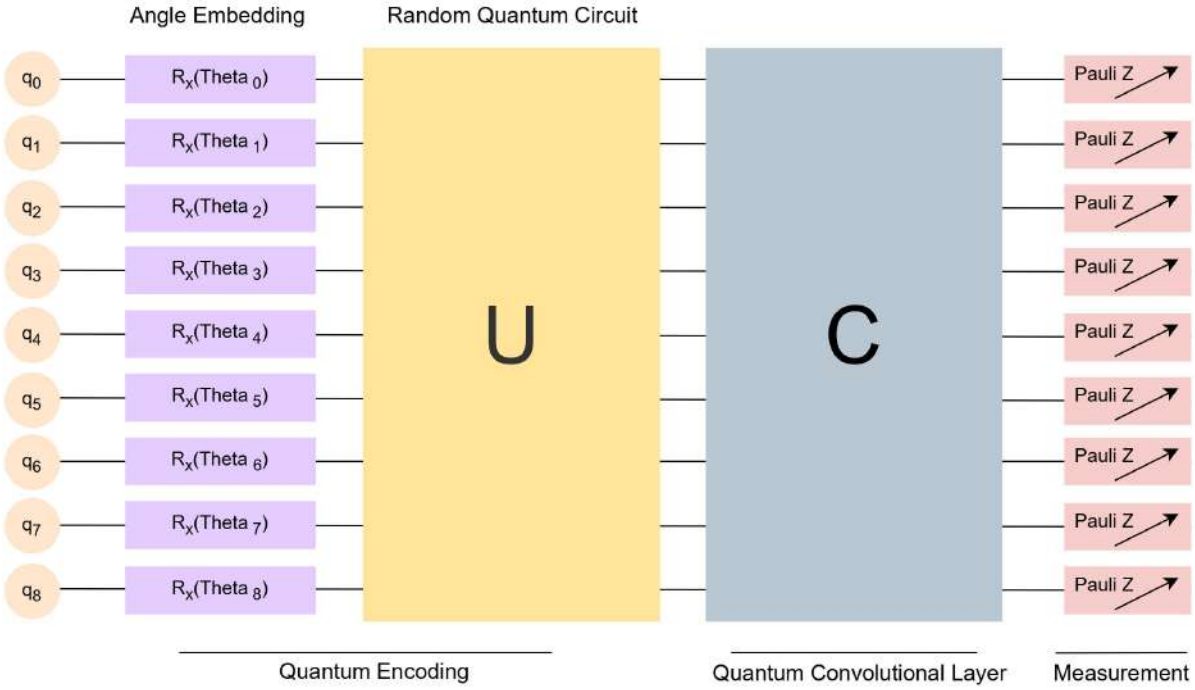


Figure 2: The block diagram of the quantum part of the proposed QCNN

The input data is first embedded into a quantum state during the quantum encoding stage. Nine qubits total are used in this study. The following equation is commonly used to represent a quantum state.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} \quad (1)$$

where $|\psi\rangle$ is the quantum state, α and β are complex coefficients or amplitudes, and $|0\rangle$ and $|1\rangle$ represent the basis states. In quantum computation, the amplitude contains the probability information of finding qubits in a specific state, and the normalized $|\alpha|^2 + |\beta|^2 = 1$ expresses this probability.

The input data can be embedded using various methods, depending on the embedding method. These embeddings each have unique traits and use cases. For instance, one method prepares a quantum state using binary data, another utilizes that data to encode data into the amplitudes of a quantum state, and yet another encodes data into the rotation angles of quantum gates. An essential step in quantum encoding is to embed the input data into a quantum state, enabling the model to use quantum computing. The type of data being used and the problem being solved determine the embedding approach to be used; each method has unique properties and applications.

The embedding approach used in this paper is called AngleEmbedding. AngleEmbedding modifies the behavior of quantum gates based on the input data values by encoding classical information into the rotation angles of the gates. In our case, the rotation gate employed is the gate denoted by the equation below.

$$R_X(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \quad (2)$$

This method is especially effective for continuous or real-valued data since it enables a concise representation of the input data within the quantum system. AngleEmbedding allows the model to benefit from the unique features of quantum computing, which may improve speed and make it possible to identify intricate patterns in the data that conventional methods could miss. The application of a random quantum circuit comes after the input data has been integrated into a quantum state. Many single-qubit gates, including the RX and controlled-NOT gates with the same angle, were randomly selected to make up this circuit. The random circuit operates as a non-linear transformation of the input data, which might aid in revealing intricate patterns in the data that traditional methods might find challenging to identify.

The image is then processed by a quantum convolutional step, which works similarly to a conventional convolutional layer but uses quantum circuits. This step entails processing the image through many quantum filters, each of which is a quantum circuit that only processes a small portion of the image. The output tensor, which depends on the quantum kernel size and the type of padding employed, will take a particular shape depending on the stride and padding used. After that, a selected activation function, such as ReLU, sigmoid, or tanh, is applied to the output tensor to create nonlinearity. The Pauli-Z operator is then used for each qubit to measure it on a computational basis. The expectation values are employed as features and fed into the traditional CNN layers in the classical part.

3.2.2. Classical part

Convolutional and fully connected blocks are combined to create a deep learning architecture in the classical part. The primary classifier for the processed dataset is represented by this architecture, which uses the predictive capabilities of traditional deep learning architectures. Max-pooling layers are put first, then convolutional layers, to create convolutional blocks. To lessen overfitting, they also have dropout layers. These blocks are in charge of identifying regional patterns and features in the input data, assisting the model in recognizing crucial data details for the classification task.

On the other hand, fully connected blocks act as the model's final steps, merging the information that the convolutional blocks have learned and deciding how to interpret the data. The final dense layer, which has a SoftMax activation function, comes after the fully connected layers and generates the class probabilities for each potential class. Classical architecture may learn intricate patterns in the data and produce precise predictions based on the cleaned dataset by combining these building blocks with information derived from the quantum layer.

4. Experimental results and discussion

Four main experiments are conducted to evaluate the general effectiveness of the proposed image classification model based on QCNN. Seven benchmark datasets are used to assess how well the proposed model based on QCNN performs. These datasets include BreastMNIST, OrganMNIST, ChestMNIST, PneumoniaMNIST, FashionMNIST, CIFAR-10, and Brain MRI Images. The adopted datasets are described in Table 1 according to the number of samples and classes. The adopted benchmark datasets have a varying number of classes, as can be shown.

It should be noted that the proposed model was evaluated on 200 training samples and 50 testing samples due to the existing constraints of quantum hardware. These samples were chosen at random. The first experiment aims to identify the best kernel size by experimenting with different kernel sizes. The model's performance in the second experiment is contrasted before and after adding the Quantum convolutional layer. The third experiment evaluates the proposed model using several benchmark datasets. The fourth experiment examines the proposed model with state-of-the-art models. These experiments used accuracy, sensitivity, precision, and f1-score as evaluation criteria. The CoLab notebook platform and the Pennylane Quantum Library were used for all experiments.

Table 1
Datasets description

	Number of Samples	Number of Classes
Brain MRI	253	2
PneumoniaMNIST	5,856	2
BreastMNIST	780	2
OrganMNIST	23,660	11
FashionMNIST	70,000	10
CIFAR-10	60,000	10
ChestMNIST	112,120	2

Finding the ideal kernel size for the proposed model is the goal of the first experiment in Figure 3. It should be noted that the Brain MRI Images dataset, one of the adopted datasets, is used to test the ideal kernel size. As can be seen, the accuracy for the kernel with a size of 1x1 is 73.75%; for the kernel with a size of 5x5 it is 77.50%; for the kernel with a size of 10x10 it is 97.95%; and finally, for the kernel with a size of 20x20, it is 98.65%. From this chart figure, it can be demonstrated that a 20x20 kernel size is the most efficient kernel size. The following experiments will continue to use this kernel size in the upcoming experiments. Also, 700 epochs will be used as well.

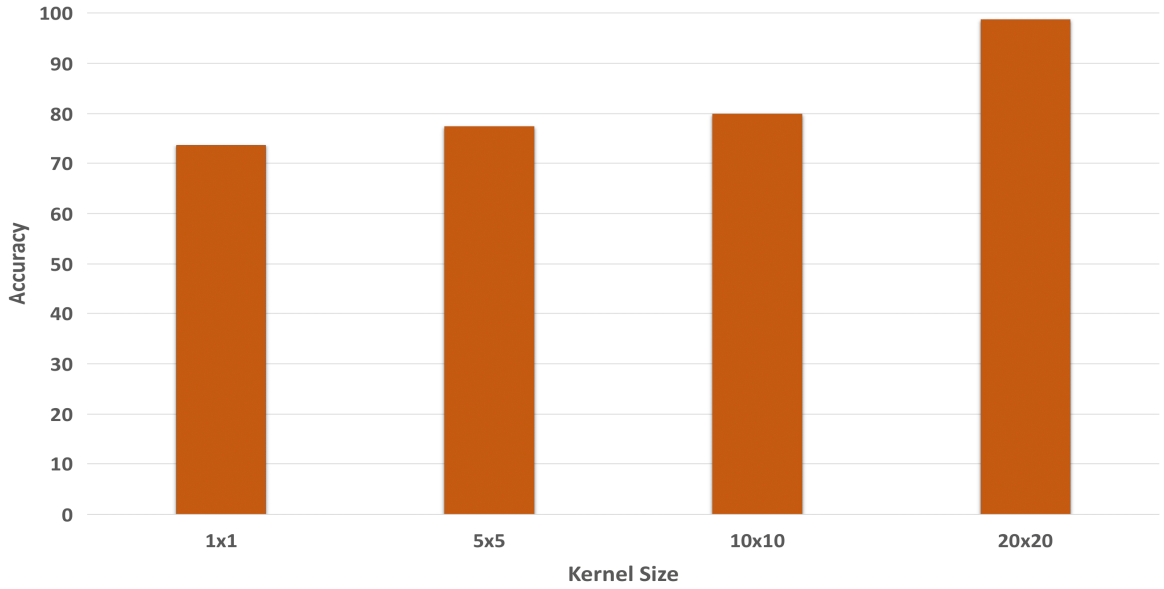


Figure 3: Comparison of using different kernel sizes in terms of accuracy

Results are obtained before and after applying the quantum convolutional layer to the dataset of Brain MRI images to demonstrate the significance of using the layer in the proposed QCNN. The experiment makes use of some evaluation metrics. Accuracy, precision, recall, and f1-score are the measures used for evaluation. Using the quantum convolutional layer increased the proposed model's accuracy to 98.65%, while doing without it reduced it to 92.68%, as shown in Table 2. Thus, it can be seen that applying the fundamentals of quantum computing to one of the convolutional layers can considerably boost the functionality of CNN architecture.

Table 2
The results before and after applying the quantum convolutional layer

	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Before	0.92	0.80	0.76	0.76
After	0.98	0.98	0.98	0.98

The proposed model was subsequently assessed and tested on seven benchmark datasets in the third experiment in Table 3, including the Brain MRI dataset and six other benchmark datasets: PneumoniaMNIST, BreastMNIST, OrganMNIST, FashionMNIST, CIFAR-10, and ChestMNIST. Table 3 compares the proposed model's performance on several datasets, presenting the findings regarding accuracy, precision, recall, and f-score. As demonstrated, the proposed model exhibits superior learning and picture classification abilities across all datasets. Furthermore, the proposed model performs well in multi-class and binary classification issues, demonstrating its adaptability and versatility in handling different classification tasks. This emphasizes the proposed model's adaptability and robustness in various image classification cases. The training and validation accuracies are presented for 700 epochs in Figure 4 to assess the proposed model's performance further. The proposed model is up-and-coming, as this graph shows. Both the validation and training datasets showed high classification accuracy. These outcomes match those of the experiments shown in Table 3.

Table 3
The performance of the proposed model using seven benchmark datasets in terms of accuracy, precision, recall, and an f-score.

	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Brain MRI	0.98	0.98	0.98	0.98
PneumoniaMNIST	0.99	0.99	0.98	0.98
BreastMNIST	0.96	0.96	0.96	0.95
OrganMNIST	0.98	0.98	0.97	0.97
FashionMNIST	0.98	0.98	0.98	0.98
CIFAR-10	0.90	0.90	0.89	0.87
ChestMNIST	0.99	0.99	0.91	0.95

The performance of the proposed model in comparison to the state-of-the-art models, namely hybrid quantum-classical convolutional neural network (HQC-CNN) [17], attentive octave convolutional capsule network (AOC-Caps) [18], Feature Pyramid Vision Transformer (FPViT) [19], cross-stitch Affine Network (CANet) [20], and convolution neural networks with 3x3 filter size (ConvNet) [21] is shown in Table 4 for further assessment of the robustness of the model. This table demonstrates how competitive the proposed model is. On most of the adopted datasets, it achieves the highest accuracy.

Table 4
The performance of the proposed model vs. the state-of-the-art models in terms of accuracy

	The Proposed Model	HQC-CNN [17]	AOC-Caps [18]	FPViT [19]	CANet [20]	ConvNet [21]
Brain MRI	98.69	97.8	-	-	-	-
PneumoniaMNIST	99.56	-	93.1	-	-	-
BreastMNIST	96.00	-	-	88.46	-	-
OrganMNIST	98.22	-	93.1	-	-	-
ChestMNIST	99.59	-	-	-	98.8	-
FashionMNIST	98.00	-	-	-	-	93.68
CIFAR-10	90.00	-	-	-	-	73.04

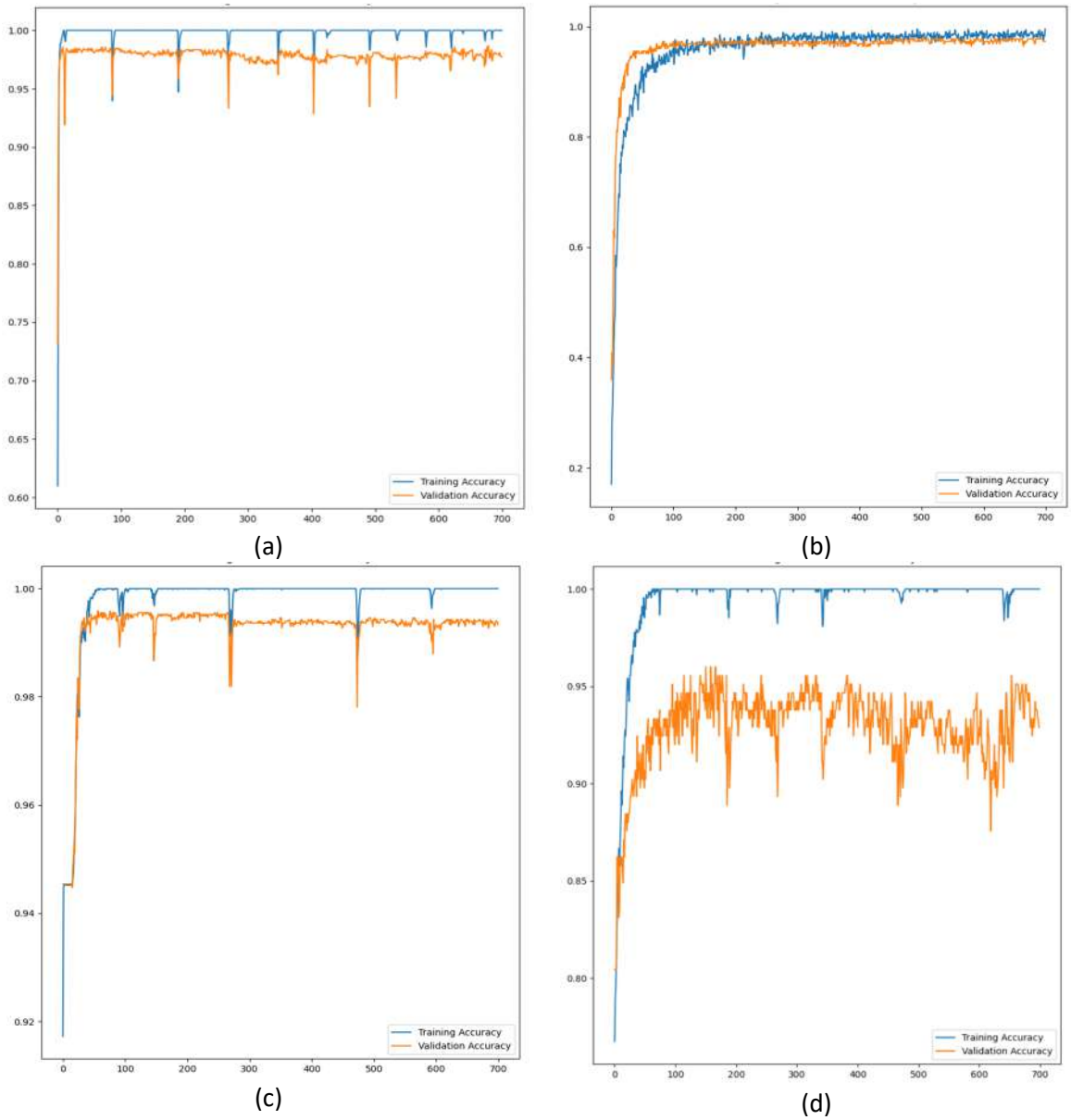


Figure 4: Training and validation accuracy through 700 epochs, (a) Brain MRI Images, (b) OrganMNIST, (c) ChestMNIST, and (d) BreastMNIST

5. Conclusion

This study proposed a new quantum convolutional neural network architecture. The proposed QCNN is applied to the image classification task. PneumoniaMNIST, FashionMNIST, CIFAR-10, BreastMNIST, OrganMNIST, ChestMNIST, and Brain MRI Images benchmark datasets are adopted. The experimental results revealed that exploiting quantum principles such as superposition and entanglement in the classical CNN can positively boost its performance. The results showed that increasing the kernel size of the quantum layer can significantly boost the performance of the proposed QCNN. Moreover, the results revealed that the proposed QCNN obtained the best results compared to several well-known landmark models. The overall proposed image classification model achieves an accuracy of 98% for Brain MRI Images, 99% for ChestMNIST, 98% for OrganMNIST, 96% for BreastMNIST, 90% for CIFAR-10, 98% for FashionMNIST, and 99% for PneumoniaMNIST. The main challenge in implementing the proposed QCNNs to large-scale image classification tasks is the restricted capacity of quantum hardware, which limits the size of input images and the number of layers employed in the network. As a result, in the future, the proposed QCNN may be implemented using simple quantum circuits.

6. Acknowledgements

We extend our sincere gratitude to Artivay.INC's Quantum R&D department for their generous sponsorship and invaluable support. Their contribution has been instrumental in the success of our project, and we are truly grateful for their partnership.

7. References

[1] Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. Pattern Recognition Letters, 141, 61-67. https://doi.org/10.1016/j.patrec.2020.07.042

[2] A. Zharmagambetov, M. Gabidolla, and M. Á. Carreira-Perpiñán, "Improved Multiclass Adaboost For Image Classification: The Role Of Tree Optimization," 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 2021, pp. 424-428, doi: https://doi.org/10.1109/ICIP42928.2021.9506569

[3] Z. Lin, J. Jia, W. Gao, and F. Huang, "Fine-grained visual categorization of butterfly specimens at sub-species level via a convolutional neural network with skip-connections," in Neurocomputing, vol. 384, pp. 295-313, Apr. 2020, doi: https://doi.org/10.1016/j.neucom.2019.11.033.

[4] L. Liu et al., "Deep Learning Based Automatic Multiclass Wild Pest Monitoring Approach Using Hybrid Global and Local Activated Features," in IEEE Transactions on Industrial Informatics, vol. 17, no. 11, pp. 7589-7598, Nov. 2021, doi: https://doi.org/10.1109/TII.2020.2995208

[5] H. Yao, X. Zhang, X. Zhou, and S. Liu, “Parallel Structure Deep Neural Network Using CNN and RNN with an Attention Mechanism for Breast Cancer Histology Image Classification,” Cancers, vol. 11, no. 12, p. 1901, Nov. 2019, https://doi.org/10.3390/cancers11121901

[6] Y. Guo, L. Du, and G. Lyu, “SAR Target Detection Based on Domain Adaptive Faster R-CNN with Small Training Data Size,” Remote Sensing, vol. 13, no. 21, p. 4202, Oct. 2021, https://doi.org/10.3390/rs13214202

[7] L. Liu, F. -X. Wu, Y. -P. Wang and J. Wang, "Multi-Receptive-Field CNN for Semantic Segmentation of Medical Images," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 11, pp. 3215-3225, Nov. 2020. https://doi.org/10.1109/JBHI.2020.3016306.

[8] C. Li, W. Xia, Y. Yan, B. Luo and J. Tang, "Segmenting Objects in Day and Night: Edge-Conditioned CNN for Thermal Image Semantic Segmentation," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 7, pp. 3069-3082, July 2021, https://doi.org/10.1109/TNNLS.2020.3009373

[9] X. Lei, H. Pan and X. Huang, "A Dilated CNN Model for Image Classification," in IEEE Access, vol. 7, pp. 124087-124095, 2019, https://doi.org/10.1109/ACCESS.2019.2927169

[10] Qin, J., Pan, W., Xiang, X., Tan, Y., & Hou, G. (2020). A biological image classification method based on improved CNN. Ecological Informatics, 58, 101093. Doi: https://doi.org/10.1016/j.ecoinf.2020.101093

[11] Hussain, M., Bird, J.J., Faria, D.R. (2019). A Study on CNN Transfer Learning for Image Classification. In: Lotfi, A., Bouchachia, H., Gegov, A., Langensiepen, C., McGinnity, M. (eds) Advances in Computational Intelligence Systems. UKCI 2018. Advances in Intelligent Systems and Computing, vol 840. Springer, Cham. https://doi.org/10.1007/978-3-319-97982-3_16

[12] K. Pasupa and W. Sunhem, "A comparison between shallow and deep architecture classifiers on small dataset," 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 2016, pp. 1-6, https://doi.org/10.1109/ICITEED.2016.7863293.

[13] Giuntini, R., Holik, F., Park, D. K., Freytes, H., Blank, C., & Sergioli, G. (2023). Quantum-inspired algorithm for direct multi-class classification. Applied Soft Computing, 134, 109956. https://doi.org/10.1016/j.asoc.2022.109956

[14] Mücke, S., Heese, R., Müller, S. et al. Feature selection on quantum computers. Quantum Mach. Intell. 5, 11 (2023). https://doi.org/10.1007/s42484-023-00099-z
[15] V. Rajesh, U. P. Naik, and Mohana, "Quantum Convolutional Neural Networks (QCNN) Using Deep Learning for Computer Vision Applications," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2021, pp. 728-734, doi: https://doi.org/10.1109/RTEICT52294.2021.9574030.
[16] Wei, S., Chen, Y., Zhou, Z. et al. A quantum convolutional neural network on NISQ devices. AAPPS Bull. 32, 2 (2022). https://doi.org/10.1007/s43673-021-00030-3
[17] Y. Dong, Y. Fu, H. Liu, X. Che, L. Sun, and Y. Luo, "An improved hybrid quantum-classical convolutional neural network for multi-class brain tumor MRI classification," J. Appl. Phys., vol. 133, no. 6, pp. 064401, 2023. https://doi.org/10.1063/5.0138021
[18] H. Zhang, Z. Li, H. Zhao, Z. Li, and Y. Zhang, "Attentive Octave Convolutional Capsule Network for Medical Image Classification," Applied Sciences, vol. 12, no. 5, p. 2634, Mar. 2022, https://doi.org/10.3390/app12052634
[19] J. Liu, Y. Li, G. Cao, Y. Liu and W. Cao, "Feature Pyramid Vision Transformer for MedMNIST Classification Decathlon," 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 1-8, https://doi.org/10.1109/IJCNN55064.2022.9892282
[20] Chen, X., Meng, Y., Zhao, Y., Williams, R., Vallabhaneni, S.R., Zheng, Y. (2021). Learning Unsupervised Parameter-Specific Affine Transformation for Medical Images Registration. In, et al. Medical Image Computing and Computer Assisted Intervention – MICCAI 2021. MICCAI 2021. Lecture Notes in Computer Science, vol 12904. Springer, Cham. https://doi.org/10.1007/978-3-030-87202-1_3
[21] Khanday, O. M., Dadvandipour, S., & Lone, M. A. (2021). Effect of filter sizes on image classification in CNN: a case study on CFIR10 and Fashion-MNIST datasets. IAES International Journal of Artificial Intelligence (IJ-AI), 10(4), 872-878. ISSN: 2252-8938. https://doi.org/10.11591/ijai.v10.i4.pp872-878.

Analysis of Discrete Wavelet Spectra of Broadband Signals

Olha Oliinyk[a], Yuri Taranenko[b] and Valerii Lopatin[c]

[a] *College of Radio Electronics, st. Stepan Bandera, 18, Dnipro, 49000, Ukraine*
[b] *Company «Likopak» st.Kachalova, 1, Dnipro, 49005, Ukraine*
[c] *Poljakov Institute of Geotechnical Mechanics of the National Academy of Sciences of Ukraine st. Simferopolska., 2a, Dnipro, 49005,Ukraine*

Abstract

The article is devoted to the development of a mathematical model for autoregressive and autocoherent analysis of discrete wavelet spectra with approbation on known experimental data. The model is used to detect anomalies or emergency states of the object after wavelet filtering of noise and zero crossing. The proposed method for processing noisy signals, combining noise filtering and Mahalanobis proximity analysis, shows better machine learning results than neural networks and other methods for detecting classification and recognition anomalies. The wavelet spectra were processed using autoregressive and autocoherent analyses. The possibility of using these functions as classification features for identifying possible anomalous states of devices is shown. Before forming the wavelet spectra, the procedure of direct with filtering and inverse transformation of the already filtered signal is carried out. As a result, it seems possible to identify the characteristic features of the signal - anomalies or emergency conditions. The main factor in the classification of device status signals is the moment when changes in the signal spectrum begin to develop, therefore, all analysis, both in the time and frequency domain, is tied either to the number of samples by the time the signal was received, and in some cases even to the extended date format - month number.

Keywords 1

Wavelet spectrum, autocoherence, autoregression, state anomaly, machine learning

1. Introduction

Computer simulation, processing, identification, and analysis of signals are among the most urgent tasks facing intelligent systems today. The task of analyzing the state signals of complex objects attracts the attention of specialists, since there is no single approach and method for identifying signals in computerized control and monitoring systems. This problem has not been solved for broadband signals, for which the frequency band used for signal transmission is much wider than the minimum required for information transmission.

The most common approach to signal analysis is the search and comparison with a signal database containing signal samples with which the obtained data are compared [1]. Comparison of signal records with a reference base can be carried out in different ways: cross-correlation; distance comparisons; cluster analysis; application of the likelihood ratio; hybrid expert methods [2]. These methods are widely covered in the literature.

A well-known approach to solving this problem of analyzing and classifying signals is to find the optimal feature space in which objects (signals) can most easily be separated using classical classification algorithms [3, 4]. If the diagnostic analysis is carried out for a long period of operation and is characterized by the appearance and development of increasing non-informative noise, the known methods of signal analysis cannot be applied, since the comparison of signals will be incorrect

CMIS-2023 (The Sixth International Workshop on Computer Modeling and Intelligent Systems), 3 of May 2023, Zaporizhzhia, Ukraine
EMAIL: oleinik_o@ukr.net (O. Oliinyk); tatanen@ukr.net (Yu. Taranenko); vlop@ukr.net (V. Lopatin)
ORCID:0000-0003-2666-3825 (O. Oliinyk); ORCID: 0000-0003-2209-2244 (Yu. Taranenko);ORCID: 0000-0003-2448-0857 (V. Lopatin)

[5]. The problem can be solved by analyzing the state signals of devices and identifying the characteristic features of the signal that cause anomalous or emergency states of objects.

The implementation of this approach makes it possible to obtain correct data for the formation of a comparative base of signals for known methods of signal identification, and to become an independent analysis tool. Since in real conditions of monitoring and control it is not uncommon to work with signals about which there are no a priori data [2], the use of classification of status signals significantly expands the scope of diagnostic analysis of signals.

2. Analysis of the literature data and a formulation of the problem

The analysis shows that the most common types of signal errors are: stop of data transmission from the measuring device, atypical data outliers, failures of measuring equipment, accumulation of errors due to interference, data breaks, etc. [6]. In addition, the need for data processing in comparison with the specified threshold signal values also precludes a qualitative analysis of information [6].

The use of filtering the data of the diagnostic analysis of the control object makes it possible to eliminate noise, however, in the future, it is not possible to isolate from the data the signs of loss of device operability. Numerous publications on this problem are devoted to the search for an algorithm or a combination of data processing methods for selective filtering of information.

Under conditions of growth of non-informative noise, diagnostic signals of complex objects are non-stationary complex signals, which are characterized by the presence of complex time dependences of amplitude, frequency, and phase. Therefore, it seems logical to apply filtering using the wavelet transform. In the general case, the wavelet transform is the decomposition of a signal into a wavelet spectrum (signal decomposition) with subsequent processing (detailing) and reconstruction [7-10]. The height of the signal peak and its location in the frequency spectrum are ideal characteristics for classifiers (random forest, gradient boosting, logistic regression, etc.) used in machine learning [11]. However, studies in [8-12] confirm that filtering using only the continuous wavelet transform does not solve the problem of filtering high-frequency and low-frequency noise.

Signal classification using discrete wavelet transform (DWT) is as follows: DWT is used to separate the signal into different frequency subbands [13]. If there are different frequency characteristics in the signals, the characteristic features appear in one of the frequency subranges. Thus, when generating features from each subrange and using a set of features as input data for a classifier in machine learning (random forest, gradient boosting, logistic regression, etc.), the task of identifying state signals for various types of signals is realized [14]. Therefore, classifier machine learning using signal processing methods has wide application prospects.

The literature widely describes the application of various criteria that are used in the application of classical classification algorithms. The work [15] considers the implementation of the block of mathematical signal processing, which is implemented by deconstructing the signal into its frequency subbands, from each subbands areas are generated that can be used as input data for the comparative proximity of the series. Authors [16. 17] use the autocoherence function to determine the local in time non-stationarity of a random process [18-20]. The above model of autocoherence is used both in the spectral and in the correlation representation. This makes it possible to determine signal anomalies in local frequency and time intervals.

Analysis of publications shows that a large amount of experimental data arrays makes it impossible to use a single function for all arrays. For real-time data processing, other approaches should be used, for example, the initial filtering of incoming data by filtering out obviously incorrect measurements [6].Thus, research related to the development and improvement of signal analysis methods for the subsequent application of machine learning methods is an urgent scientific task. The purpose of this work is to develop a mathematical model for autoregressive and autocoherent analysis of discrete wavelet spectra for the classification of noisy signals by machine learning.

3. Mathematical model of discrete wavelet spectra

This paper uses empirical data collected during experimental studies of reliability characteristics using four bearings (B1-B4) on a loaded shaft (6000 pounds) rotating at a constant speed of 2000 rpm

[21]. The 2nd_test set containing one accelerometer was used for the analysis. The dataset is formatted in separate files, each containing a 1-second snapshot of the vibration waveform recorded at specific time intervals. Each file consists of 20480 points with a sampling frequency of 20 kHz [21]. The file name indicates the date of the data capture which occurred every 10 minutes. The known data were taken in order to compare the obtained results on the processing of vibration monitoring data performed by other authors [21].

3.1 Autocoherence for Stationarity Analysis of Discrete Spectra

If there is noise in the signal, then the series can become non-stationary and it is impossible to extract the necessary information from it, which is usually contained in the low-frequency part of the spectrum. Such a characteristic for the series of wavelet coefficients is autocoherence, which can be determined from the well-known relation for the continuous wavelet spectrum:

$$W_X = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} I(t) \varphi \left(\frac{t-b}{a} \right) dt, \quad (1)$$

where $\varphi \left(\frac{t-b}{a} \right)$– wavelet function; W_X– the scale-time spectrum of the signal I(t); a – the scale factor, b – the signal shift along the time axis.

To determine autocoherence, one should use the relation for wavelet coherence:

$$R^2(a,b) = \frac{|s(a^{-1} W_{XY}(a,b))|^2}{s(a^{-1}|W_X(a,b)|^2) s(a^{-1}|W_Y(a,b)|^2)}, \quad (2)$$

where s – smoothing operator [10]. Coherence is interpreted as the square of the correlation coefficient, and its values range from 0 to 1.

Autocoherence is determined by a simple substitution in (2) of relations (1) and (3):

$$W_Y = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} \frac{dI(t)}{dt} \varphi \left(\frac{t-b}{a} \right) dt, \quad (3)$$

It should be noted that the relation for autocoherence is used to estimate the stationarity of a number of discrete wavelet coefficients.

3.2 Proximity of time series for the analysis of autoregression and autocoherence

Let us designate the next two time series of the wavelet coefficients of the spectra as:

$$X_i = \{\tilde{a}_{j,k}\}, \{\tilde{d}_{2,k}\}, \dots, \{\tilde{d}_{j,k}\} \dots \{\widetilde{d_{J,k}}\}, k = 1,2, \dots \frac{N}{2^j}, j = 1 \dots J,$$
$$Y_i = \{\tilde{a}_{j,k}\}, \{\tilde{d}_{2,k}\}, \dots, \{\tilde{d}_{j,k}\} \dots \{\tilde{d}_{j,k}\}, k = 1,2, \dots \frac{N}{2^j}, j = 1 \dots J, \quad (4)$$

where X_i – series of wavelet coefficients at time t_i; Y_i – series of wavelet coefficients at time t_{i+dt}; dt – determined by the time for which a change in the spectrum will not lead to equipment failure.

The correlation closeness of series (4) is determined from the relationship:

$$proximity = 1 - \frac{\sum_{i=1}^{N} X_i Y_i}{\sqrt{\sum_{i=1}^{N} X_i^2 \sum_{i=1}^{N} Y_i^2}}. \quad (5)$$

When the spectra series are close and completely identical proximity=0 when there is no difference -proximity=1. On practice $0 \le \text{proximity} \le 1$.

3.3. Decomposition of the Time Series

The wavelet coefficients in the decomposition of the measuring signal f(t) are generally determined from the following system of equations [22]:

$$\left.\begin{aligned} a_{j,k} &= \int_R f(t)\varphi_{j,k}(t)dt \\ d_{j,k} &= \int_R f(t)\psi_{j,k}(t)dt \end{aligned}\right\}, \quad (6)$$

where R – the interval for determining the function f(t) of the signal; $a_{j,k}$, $d_{j,k}$ – the coefficients of approximation and detailing of the discrete wavelet decomposition of the signal, respectively; $\varphi_{j,k}$, $\psi_{j,k}$ – father and mother wavelets, respectively; j, k – he current level of the wavelet decomposition and the ordinal number of the wavelet coefficient in the wavelet decomposition of the signal, respectively.

In calculations, instead of integrating (6), Mull's pyramidal algorithm is used to eliminate methodological errors associated with quadrature formulas [23]. The approximating coefficients are set at the conditionally zero level j_o, $j_{0,k}$, k=1,…N, where $N=2^m$, m>1. Accuracy was assessed using the predictive regression metrics method and an uncertainty matrix.

Next, approximating $a_{j_{0+1},k}$ and detailing $d_{j_{0+1},k}$ coefficients are calculated, and from $a_{j_{0+1},k}$ вычисляют $a_{j_{0+1,2}\,k}, d_{j_{0+1,2}\,k}$ and so on. In the presence of noise with zero mathematical expectation and standard deviation, the set of coefficients enclosed in arrays array ({}) for the maximum decomposition level J will take the form of the formula:

$$\{\tilde{a}_{j,k}\}, \{\tilde{d}_{2,k}\}, \dots, \{\tilde{d}_{j,k}\} \dots \{\tilde{d}_{j,k}\}, k = 1,2, \dots \frac{N}{2^j}, j = 1 \dots J, \quad (7)$$

The set of nested arrays (7) is transformed into a 1D list using the special flatten function [24]:

$$\left[\tilde{a}_{j,k}, \tilde{d}_{2,k}, \dots \tilde{d}_{j,k} \dots \tilde{d}_{j,k}\right], k = 1,2, \dots \frac{N}{2^j}, j = 1 \dots J, \quad (8)$$

3.4 Noise Reduction by Limiting Detail Factors

The filtered signal is determined from the relation [25]:

$$\hat{f}(x) = \sum \bar{a}_{j+j_{o,k}} \varphi_{j+j_{o,k}}(t) + \sum_{j=1}^{j} \sum_{k} \left[F(\lambda_i)\bar{d}_{j+j_{o,k}} \psi_{j+j_{o,k}}(t)\right], \quad (9)$$

where $F(\lambda_j)$ – threshold function from the list garotte, garrote, greater, hard, less, soft, for example, hard(λ_j), λ_j – threshold at the j decomposition level; $\varphi_{j+j_{o,k}}(t), \psi_{j+j_{o,k}}(t)$ – the "father" and "mother" wavelets, respectively [25]. The universal threshold λ_j is determined from the relation [25]:

$$\left.\begin{aligned} \sigma &= \frac{\text{median}(|d_{1,k}|)}{0{,}6742}, \\ \lambda_j^{\text{univ}} &= \sigma\sqrt{2\ln N_j}, \end{aligned}\right\} \quad (10)$$

where λ_j^{univ} – universal threshold; N_j – he number of detail coefficients $d_{1,k}$ at the j decomposition level; *median*($|d_{1,k}|$)– median from the array of detail coefficients.

A common threshold can be used as the average of λ_j^{univ} or both series (4):

$$\lambda_X = \lambda_Y = \frac{1}{n}\sum_{i=1}^{N} \lambda_i, \quad (11)$$

The universal threshold $\lambda_j^{\text{univ}} = \sigma\sqrt{2\ln N_j}$ of limiting the wavelet coefficients of the spectrum depends on the dispersion and the number of wavelet coefficients of detail, therefore, when analyzing discrete spectra, it is an essential characteristic of high-frequency noise.

Depending on the nature of the signal, additional mathematical processing can be applied to discrete wavelet coefficients [25].

4. Regression Analysis of Real Data Based

Applying the proposed model of the method of regression analysis of discrete spectra, it is possible

to clearly identify the date 2004-02-19 06:12:39 of a possible defect for the B3 device (Figure 1). Identification occurs both by the number of accumulated maxima - 10 exceeding the threshold of 0.5 correlation proximity, and by the maximum value.

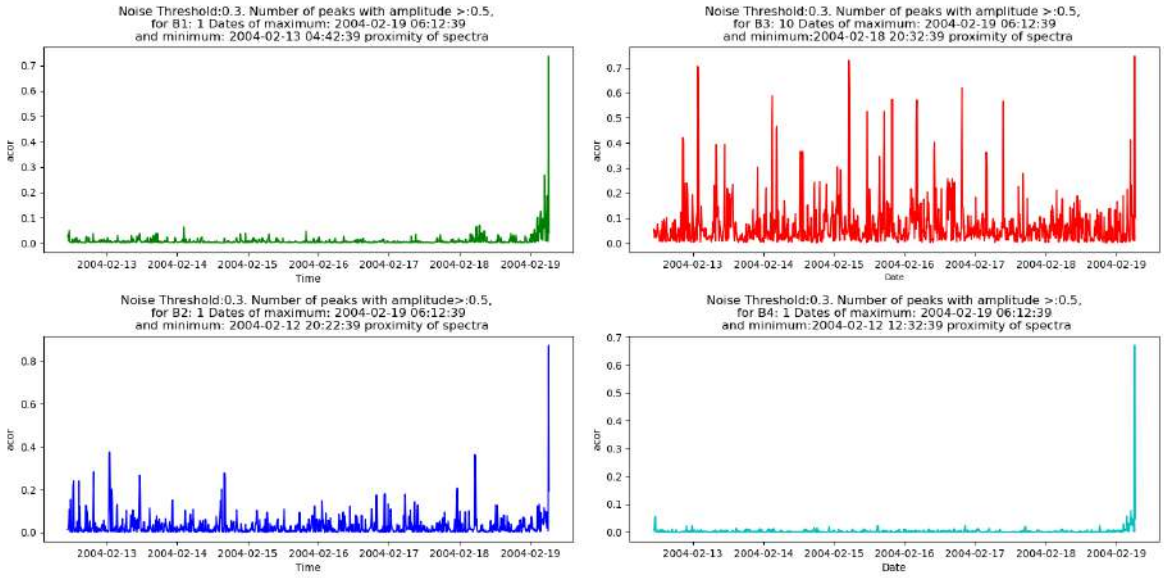


Figure 1: Autoregression of discrete wavelet spectra obtained from relations (4), (5) with filtering by a common threshold (11)

5. Coherent analysis of real data based on the developed mathematical model

The analysis shows that the proximity of the autocoherence of the spectra is the highest - unity for the B3 device and a minimum of autocoherence. Thus, the loss of stationarity is achieved on 2004-02-19 06:12:39 (Figure 2). Not despite the fact that the number of peaks -91 is less than that of the device B2.

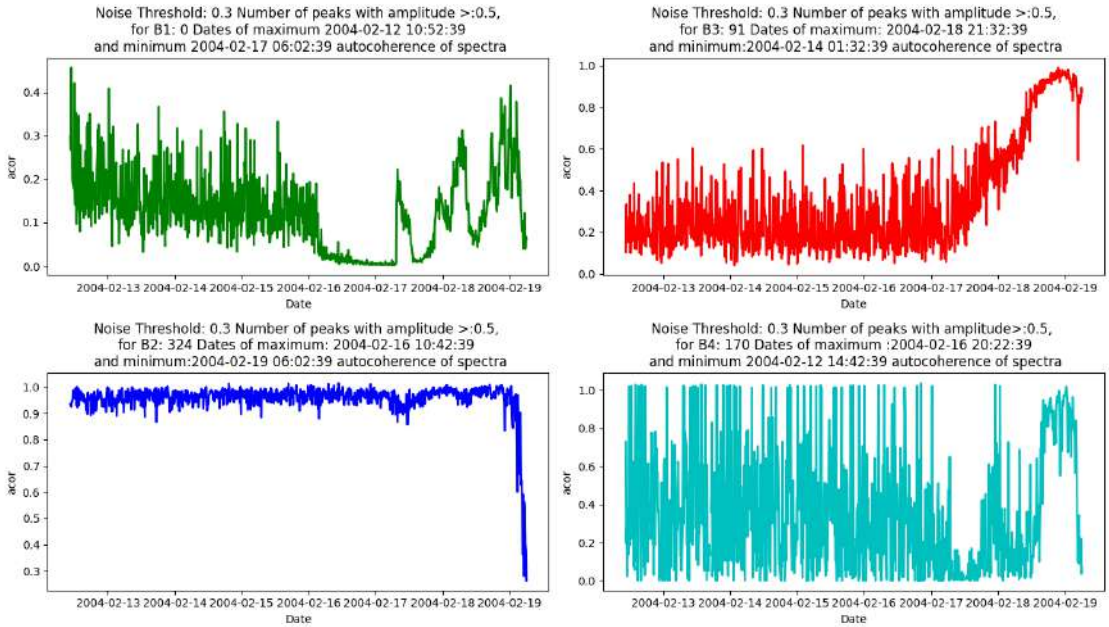


Figure 2: Autocoherence of discrete wavelet spectra obtained by relations (1), (2), (3), (4), (5) with common threshold filtering (11)

6. Using the Mahalanobis algorithm to analyze discrete wavelet spectra

To detect signal anomalies, we use zero-crossing analysis, which has practically not been used in relation to discrete wavelet spectra. This approach is designed to detect chirp and non-linear chirp crossover signals. These include the average value of the derivative, the zero crossing frequency, and the average transition rate of a given signal amplitude [25]. Combined with noise filtering and Mahalanobis proximity analysis, zero-crossing analysis shows better machine learning results than neural networks and other classification and recognition anomaly detection methods.

Features of the Mahalanobis algorithm consist in the use of a covariance matrix according to the relation [26]:

$$D_M(\bar{x}) = \sqrt{(\bar{x} - \bar{\mu}_x)^T S^{-1} (\bar{y} - \bar{\mu}_y)}, \quad (12)$$

where $\bar{x}, \bar{y}$ – multidimensional series of wavelet coefficients; $\overline{\mu_X}, \overline{\mu_Y}$ – mathematical expectations; S– he covariance matrix.

The covariance matrix S and its inverse matrix are symmetric and positive definite. Therefore, the transformation of the spectral series of wavelet coefficients from multidimensional to one-dimensional is not required. At the same time, wavelet decomposition into levels during filtering and zero-crossing dramatically increase the number of identification features. The increase in the number of features only due to zero-crossing and filtering can be seen by comparing the graphs of the training sets (Figure 3).

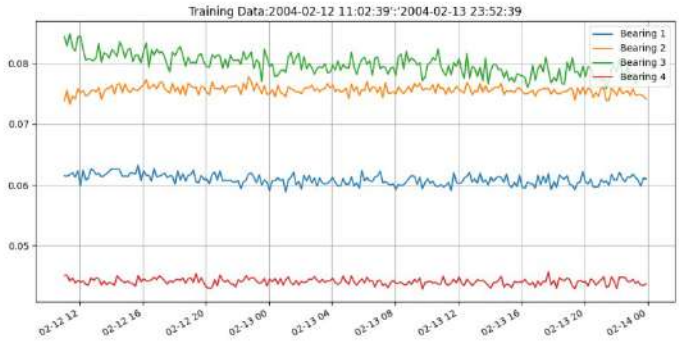


Figure 3: Graph of the training set signals in the trouble-free period of operation without the use of wavelet noise filtering and zero-crossing

From the graph (Figure 4) it follows that the method of regression analysis of discrete spectra clearly identifies the date 2004-02-19 06:12:39 of a possible defect for device B3 both by the number of accumulated maxima– 10 exceeding the threshold of 0.5 correlation closeness, and by the maximum value.

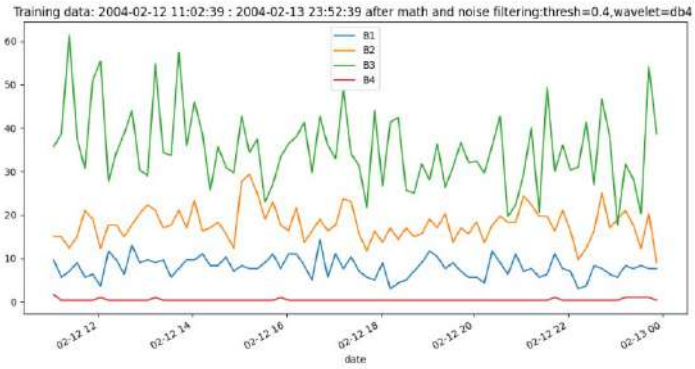


Figure 4: Graph of the training set signals in the trouble-free period of operation using wavelet noise filtering and zero-crossing

The graphs of test set signals also have a significant difference (Figure 5, Figure 6).

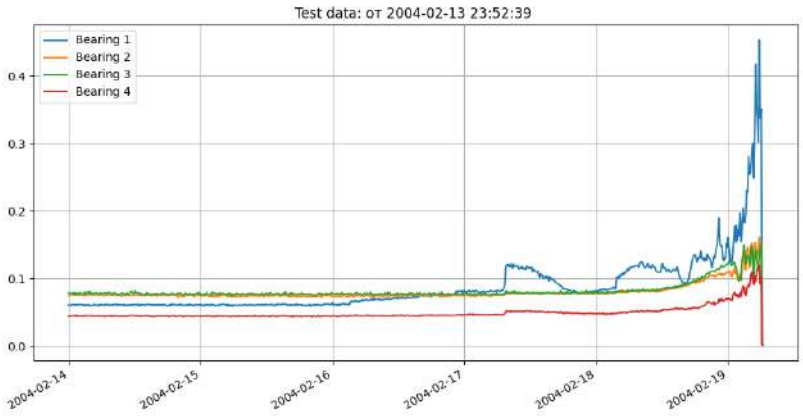


Figure 5: Graph of test set signals during detection of anomalies or emergency conditions without wavelet noise filtering and zero crossing

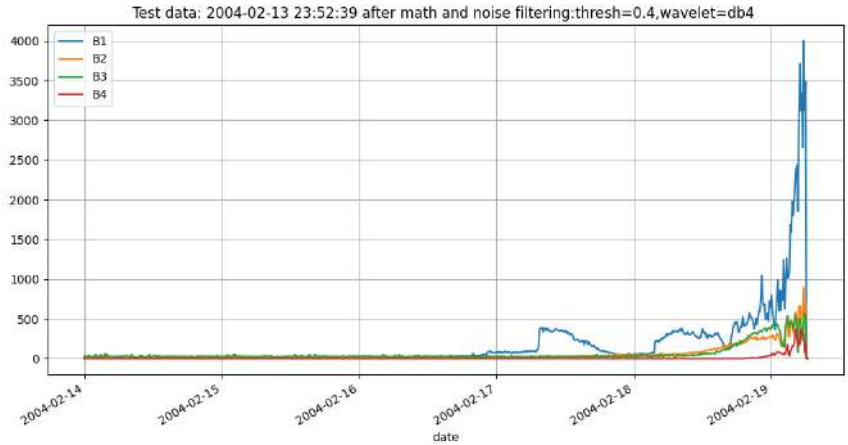


Figure 6: Graph of test set signals in the period of detection of anomalies or emergency conditions after wavelet filtering of noise and zero crossing

When determining anomalies using the described approach, a clear identification of signs of anomalies and emergency conditions of the object is observed, as evidenced by a strong oscillation of the Mahalanobis distance in the threshold zone (Figure 7, Figure 8).

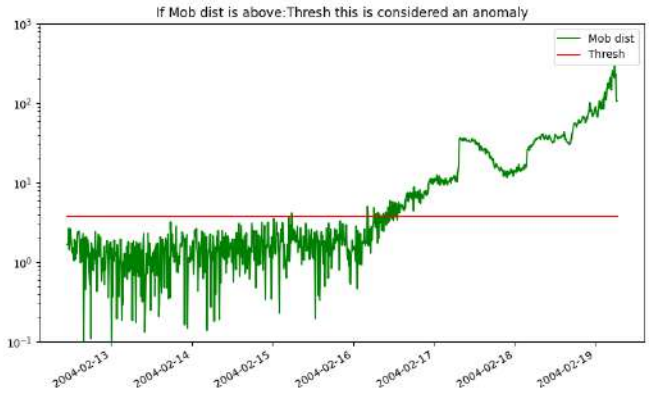


Figure 7: Changing the Mahalanobis distance in the threshold zone without signal processing

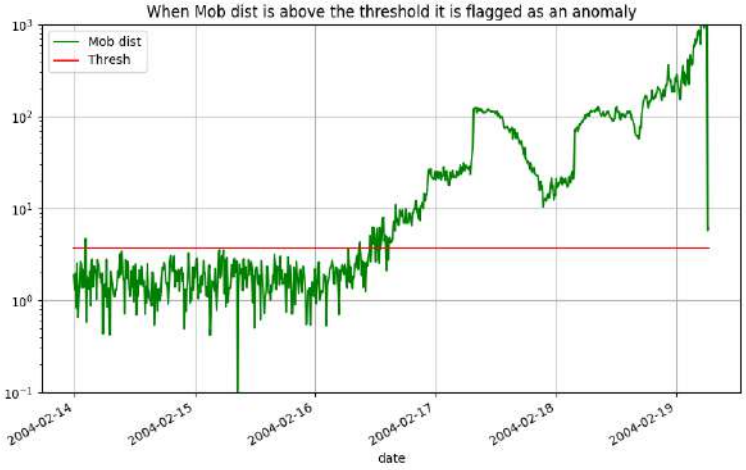


Figure 8: Changing the Mahalanobis distance in the threshold zone using Filtering Wavelet and Zero Crossing

7. The Results of the Classifier Using the Machine Learning Method

To test the proposed method for classifying state signals using the obtained features, the classifiers described by the following algorithms were studied: gradient method, linear (SVM), logistic regression, nearest neighbors, decision tree (Table 1) on the resulting computer model.

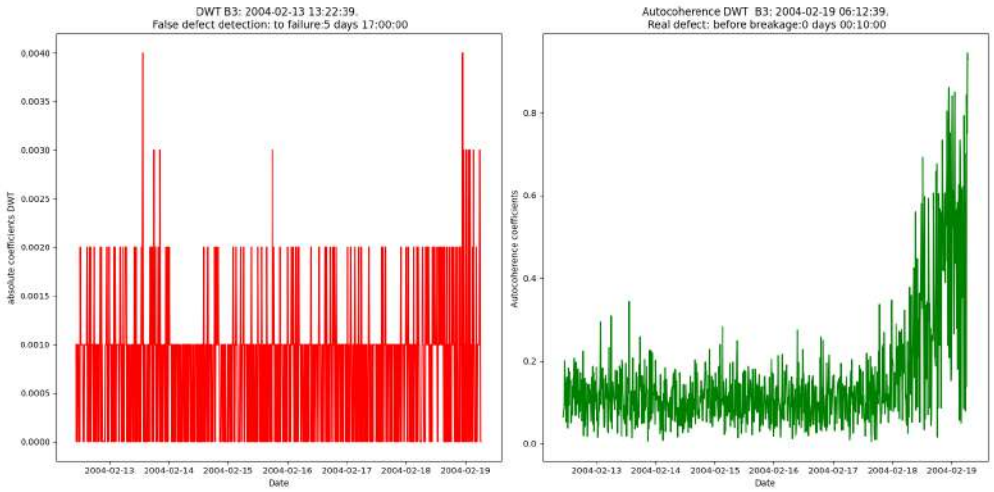
The previously obtained wavelet data analysis model was used for machine learning with the gradient method signal classifier (as the most commonly used). Time series of wavelet coefficients db38 of cosine proximity (autocoherent proximity) were taken as features. The results of machine learning of the model confirm the effectiveness of the proposed algorithm (Table 1).

To confirm the effectiveness of the developed method for classifying noisy signals using the machine learning method, we will process the same signal without using the established signs of state classification (Figure 9, a) and using the developed classifier (Figure 9, b).

Table 1

Efficiency of application of algorithms for machine learning of classifiers

№	Algorithm name	Accuracy on the training sample at the level of detail			Accuracy on the test sample at the level of detail			Studying time with varying degrees of detail		
		0	0,3	0,4	0	0,3	0,4	0	0,3	0,4
1	logistic regression	0,949	0,934	0,916	0,912	0,907	0,920	0,148	0,016	0,023
2	linear (SVM)	1,000	1,000	0,996	0,910	0,904	0,926	0,083	0,057	0,041
3	nearest neighbors	0,931	0,938	0,916	0,899	0,900	0,916	0,045	0,003	0,006
4	gradient method	1.000	1,000	1,000	0,891	0,959	0,905	1,907	1,091	1,964
5	decision tree	1,000	1,000	1,000	0,838	0,950	0,855	0,039	0,024	0,019



a b

Figure 9: Comparative analysis of the results of signal processing without (a) and with (b) the use of the developed method for classifying state signals

The use of noise filtering with its own time-dependent universal threshold makes it possible to increase the accuracy of predicting the occurrence of object defects.

8. Conclusions

1. A new method for classifying noisy signals using machine learning has been developed. Proposed method for processing noisy signals that combines noise filtering and Mahalanobis proximity analysis. The essence of the method lies in the following: the data obtained after mathematical processing for all frequency subranges of the signal and a given moment of time constitute time series, filtering each series from noise with its own, time-dependent, universal threshold. The time series filtered in this way are used as input signals to the machine learning classifier and characterize the change in the state of the control object over time. The use of this classifier implements pre-processing of the signal in order to eliminate noise and allows you to highlight the signs of inoperable states of devices in dynamics.

2. The list of used functions for processing wavelet spectra has been extended with two more ones: by cosine proximity of the autocoherence spectra of discrete wavelet spectra and by the cosine proximity function of wavelet spectra of signals following one after another in time. The possibility of using these functions as classification features for identifying possible anomalous states of devices has been confirmed.

3. A new criterion for classifying the loss of uptime is based on the presence of local minima depending on the sum of squares of the wavelet detailing coefficients at the level of the wavelet decomposition.

4. The developed classification method was tested using a computational experiment on a known data set obtained using an accelerometer, all stages of the classification of an object's inoperable state signal are shown. The obtained results of machine learning using the signal classifier gradient method confirm the effectiveness of the application of new signs of signal classification in wavelet analysis (the accuracy on the test set was 0.96).

9. References

[1] M. Bayer, J. Fábio Alice, An iterative wavelet threshold for signal denoising, Signal Processing 162 (2019) 10–20. URL:https://www.sciencedirect.com/science/article/abs/pii/S0165168-419301240.

[2] G. J. Mendis, J. Wei-Kocsis, A. Madanayake, "Deep learning based radio-signal identification with hardware design." IEEE Transactions on Aerospace and Electronic Systems 55.5 (2019): 2516–2531. doi:10.1109/TAES.2019.2891155.
[3] O. Oliynyk, Y. Taranenko, D. Losikhin, A. Shvachka, Investigation of the Kalman filter in the noise field with an excellent Gaussian distribution, Eastern-European journal of enterprise technologies 94 (2018) 36–42. doi:10.15587/1729-4061.2018.140649.
[4] W. Deng, "A novel fault diagnosis method based on integrating empirical wavelet transform and fuzzy entropy for motor bearing" IEEE Access 6 (2018): 35042–35056. doi:10.1109/ACCESS.2018.2834540.
[5] D. Yoon, Deep learning-based electrocardiogram signal noise detection and screening model, Healthcare informatics research 25 (2019) 201–211. doi:10.4258/hir.2019.25.3.201.
[6] D. V. Efanov, V. N. Myachin, G. V. Osadchiy, M. V. Zueva, Choice of method for filtering diagnostic data in systems of continuous monitoring of transport infrastructure facilities, Transport of the Russian Federation 2 (2020) 35–40.
[7] L. Wu, B. Yao, Z. Peng, Y. Gua, Fault Diagnosis of Roller Bearings Based on a Wavelet Neural Network and Manifold Learning, Applied Sciences 7 (2017) 158. doi:10.3390/app7020158. URL:www.mdpi.com/journal/applsci.
[8] Y. Xi, Z. Li, X. Zeng, X. Tang, X. Zhang, H. Xiao, Fault location based on travelling wave identification using an adaptive extended Kalman filter, IET Generation, Transmission & Distribution 12.6 (2018) 1314–1322.
[9] S. Zolfaghari, Broken rotor bar fault detection and classification using wavelet packet signature analysis based on fourier transform and multi-layer perceptron neural network, Applied Sciences 8 (2018) 25. URL:https://doi.org/10.3390/app8010025.
[10] H. Choi, J. Jeong, Speckle noise reduction technique for SAR images using statistical characteristics of speckle noise and discrete wavelet transform, Remote Sensing 11 (2019) 1184. URL: https://ieeexplore.ieee.org/abstract/document/8262663.
[11] E. Darnila, M. Ula, K. Tarigan, T. Limbong, M. Sinambela, "Waveform analysis of broadband seismic station using machine learning Python based on Morlet wavelet", IOP Conference Series: Materials Science and Engineering, IOP Publishing (2018): 012048. doi:10.1088/1757-899X/420/1/012048.
[12] K.G. Polovenko, Large-scale analysis of electroencephalograms based on wavelet transforms with Daubechies basis function, Prikl. Radioelektron 10(1) (2011) 15–21. URL:https://openarchive.nure.ua/server/api/core/bitstreams/faa076b7-4360-4189-a421-71aef249e438/content.
[13]Yu. K. Taranenko, Efficiency of using wavelet transforms when filtering noise in the signals of measuring transducers, Measuring technology 2 (2021) 16–21. URL:https://link.springer.com/article/10.1007/s11018-021-01902-8.
[14] W. Chen, J. Li, Q. Wang, K. Han, Fault feature extraction and diagnosis of rolling bearings based on wavelet thresholding denoising with CEEMDAN energy entropy and PSO-LSSVM, Measurement, 172 (2021) 108901. URL:https://www.sciencedirect.com/science/article/abs/pii/S0263224120313865.
[15] O. Yu. Oliynik, Yu. K. Taranenko, Analysis of the coherence of signals by the wavelet-reversal method, Metrology and Appliances 6 (2020) 48–52.
[16] P. F. Shchapov, R. P. Migushchenko, O. Yu. Kropachek, I. M. Korzhov, Further correlation models of spectral nonstationarity of dip signals , Metrology and Appliances 5 (2018) 11–14.
[17] I. M. Korzhov, R. P. Mygushchenko, P. V. Shchapov, O. Y. Kropachek, Studying the influence of training sample volume on the average risk of technical diagnostics, International Journal of Engineering Research and Applications (IJERA) 9 (2019) URL:https://www.ijera.com/ papers/vol9no2/S1/L0902016466.pdf.
[18] I. Korzhov, Analysis of models of the coherence function of the spectral non-stationarity of dip signals, Bulletin of the National Technical University, Series: "Hydraulic machines and hydraulic units 46 (2018) 30–34.
[19] D. Matteo, A. Longo, M. Rizzi, Noisy ECG signal analysis for automatic peak detection, Information 10.2 (2019) 35. URL:https://www.mdpi.com/2078-2489/10/2/35.

[20] M. J. Hinich, A statistical theory of signal coherence, IEEE J, Oceanic Engineering 25 (2000) 256–261. doi: http://dx.doi.org/10.1109/48.838988.
[21] Prognostics Center of Excellence - Data Repository, 2004. URL:https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-datarepository/#turbofan.
[22] Yu. E. Voskoboinikov, Wavelet Filtering with Two-Parameter Threshold Functions: Function Selection and Optimal Parameter Estimation, Automation and Software Engineering 1.15 (2016) 69–78. URL:http://https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository.
[23] G. He, S. Ma, A study on the short-term prediction of traffic volume based on wavelet analysis.Proceedings, The IEEE 5th International Conference on Intelligent Transportation Systems, IEEE, 2002. URL:https://ieeexplore.ieee.org/abstract/document/1041309.
[24] NumPy Array manipulation: ndarray.flatten() function. URL:https://www.w3resource.com/numpy/manipulation/ndarray-flatten.php.
[25] Yu. K. Taranenko, N. O. Rizun, Wavelet filtering of signals without using model functions. Radioelectronics 65 (2022) 110–125. URL:https://doi.org/10.20535/S0021347022020042.
[26] N. Rehman, B. Khan, K. Naveed, Data-driven multivariate signal denoising using Mahalanobis distance, IEEE Signal Processing Letters 26.9 (2019) 1408–1412. URL:https://ieeexplore.ieee.org/abstract/document/8784188.

Control, diagnostics and error correction in the modular number system

Victor Krasnobayev[a], Alina Yanko[b] and Dmytro Kovalchuk[a]

[a] N. Karazin Kharkiv National University, Svobody sq., 4, Kharkiv, 61022, Ukraine
[b] National University «Yuri Kondratyuk Poltava Polytechnic», Poltava,36011, Ukraine

Abstract

This article discusses methods that allow controlling, diagnosing and correcting errors that occur in a computer system (CS) operating in a modular number system (MNS). A feature of using the MNS is the possibility, in some cases, of correcting errors even if there is only one control base, using the concept of an alternative set of numbers. The article deals with the control, diagnostics and error correction in the dynamics of the data processing process. Currently, to correct errors in the dynamics of the computational process, the CS uses the projection method and its modifications. The projection method requires the calculation of all projections $\tilde{A}_i$ of the distorted number $\tilde{A}$, which leads to a large number of additional operations for each correction of an individual result. The article discusses a method for correcting errors in the CS, based on the use of a conditional alternative set of numbers in the MNS. Two methods for determining an alternative set $W(\tilde{A})$ are considered. The disadvantage of the first method is the large hardware and time costs for determining an alternative set of numbers. As a working method, the second method is proposed, in which the initial number $\tilde{A}$ is first reduced to the form $\tilde{A}^{(Z)} = (0,0,...0,\gamma_{n+1})$, i.e. the operation of zeroing the initial number $\tilde{A}$ is performed. Examples of specific execution of modular operations for a given MNS are considered.

Keywords1

Alternative set, corrective modular code, error correction, information processing cycle, modular number system, non-positional code structure, projection method, zeroing process.

1. Introduction

One of the promising areas for ensuring the fault-tolerance of CSs is the widespread use of corrective codes that can detect and correct errors that occur in the dynamics of the data processing process. A characteristic feature of such codes is the presence in the construction structure of corrective codes of interdependent parts: informational and control. The analysis of known positional codes showed that these parts of the code are not equal in relation to arithmetic operations. The non-arithmetic nature of the procedures for obtaining the check digits of the corrective code doesn't allow controlling the results of performing arithmetic operations [1-4]. Thus, it is obvious that the use of positional corrective codes when implementing arithmetic operations in a CS operating in a positional number system (PSN) is impossible.

Non-positional codes, in particular, codes in the modular number system (MNS), are deprived of this drawback. A number of works, both domestic and foreign, are devoted to the construction of corrective modular codes [5-7]. The equality of residues in the structure of the correcting code in the MNS is the basis for constructing codes capable of detecting and correcting errors in the process of

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: v.a.krasnobaev@gmail.com (V. Krasnobayev); al9_yanko@ukr.net (A. Yanko); kovalchuk.d.n@ukr.net (D. Kovalchuk)
ORCID: 0000-0001-5192-9918 (V. Krasnobayev); 0000-0003-2876-9316 (A. Yanko); 0000-0002-8229-836X (D. Kovalchuk)

implementing modular operations. In addition, this property of modular codes serves as the basis for the implementation of exchange operations between the accuracy, fault tolerance and speed of the implementation of arithmetic operations in the dynamics of the computational process of the CSs. This is due, first of all, to the fact that each residue of the modular code carries information about the entire original number [8]. Then, by varying the number of information and control bases of the MNS, it is possible to achieve the required values of the main indicators of the quality of the functioning of the CSs. One of the promising directions for ensuring the fault tolerance of the CSs is the widespread use of codes capable of detecting and correcting errors that occur in the dynamics of the data processing process. A characteristic feature of such codes is the presence in the structure of the corrective code of two interdependent parts: informational and control [9].

The specificity of the representation of numbers in the MNS allows [10], in a number of cases, not only to detect an error, but also to find the place of its occurrence, using only a single control base, which is impossible with existing methods for monitoring and correcting errors in the MNS. In some cases, it is possible to carry out error correction with a minimum code distance $d_{\min} = 2$ either by the projection method or by using the concept of an alternative set (AS) of numbers.

The projection method requires the calculation of all projections $\tilde{A}_i$ of the distorted number $\tilde{A}$, which leads to a large number of additional operations for each correction of an individual result. Hardware and especially software implementation of the projection method leads to a large expenditure of time. In addition, this method fundamentally doesn't allow unambiguous detection of the place of occurrence of any single errors, i.e. errors in one of the residues of a non-positional code structure in the MNS.

Much more effective is the method developed and researched further for correcting errors in the CSs, based on the use of a conditional alternative set (CAS) of numbers in the MNS.

2. Development and research of methods for monitoring, diagnosing and correcting errors in CS based on the use of an alternative and conditional alternative set of numbers

A distinctive feature of the MNS is the possibility, in some cases, of correcting errors even if there is only one control base, using the concept of an alternative set of numbers [5].

The set of bases: $m_{i_1}, m_{i_2}, ...m_{i_k}$ on which the numbers $A_1, A_2, ..., A_k$ differ from the wrong (distorted) number $\tilde{A}$, this is called an alternate number set of the number $A_1, A_2, ..., A_k$ and denote it as $W(\tilde{A}) = \left\{ m_{i_1}, m_{i_2}, ..., m_{i_k} \right\}$. The basic principle of determining the erroneous residue a_i is that for the set of incorrect (distorted) numbers $\tilde{A}_1, \tilde{A}_2, ..., \tilde{A}_\rho$ obtained as a result of operations, during the execution of the program, CASs are determined sequentially in time:

$$W_{\wedge}(\tilde{A}) = W(\tilde{A}_1) \wedge W(\tilde{A}_2) \wedge ... \wedge W(\tilde{A}_\rho) \quad (1)$$

where $W(\tilde{A}_l) = \left\{ m_{l_1}, m_{l_2}, ..., m_{l_\rho} \right\}$ – alternative set of *l*-th wrong (distorted) number.

Let's consider the error correction time in the dynamics of the information processing process of the CS. It is known that the correction time is determined by the following expression:

$$T_{cor} = T_{det} + T_{fix} \quad (2)$$

where T_{det} – errors detection time; T_{fix} – errors fix time.

For MNS, the error detection time in the dynamics of the data processing process is determined by the following expression:

$$T_{det} = k_1 \cdot T_{det_{\gamma_{n+1}}} + k_2 \cdot T_{det_{AS}} + k_3 \cdot T_{det_{CAS}} \quad (3)$$

where $T_{det_{\gamma_{n+1}}}$ – time to determine and check the value of γ_{n+1};

γ_{n+1} – the value of the residue of the number $\tilde{A}$ to the base m_{n+1} during the zeroing process;

$T_{det_{AS}}$ – time of determination and verification of AS $W(\tilde{A})$;

$T_{det_{CAS}}$ – time of determination and verification of CAS $W_{\wedge}(\tilde{A})$;

k_1, k_2, k_3 – multiplicity factors for determining, respectively, the processes of zeroing numbers ($\tilde{A}^{(Z)}$), the number of operations to determine the AS of numbers ($W(\tilde{A})$) and the number of operations to determine the CAS of numbers ($W_{\wedge}(\tilde{A})$).

Thus, the error correction time in the dynamics of the information processing process in the MNS is determined by the final expression:

$$T_{cor} = k_1 \cdot T_{det_{\gamma_{n+1}}} + k_2 \cdot T_{det_{AS}} + k_3 \cdot T_{det_{CAS}} + T_{fix} \quad (4)$$

Taking into account the fact that the operation of determining AS, CAS and fix (correction) errors in the MNS can be performed in a tabular version, i.e. in one cycle, get that:

$$T_{cor} \approx k \cdot T_Z \quad (5)$$

where k – number of stages of AS determination;

T_Z – zeroing time, which is necessary to convert the original number $A = (a_1, a_2, ..., a_n, a_{n+1})$ into a number of the form $\tilde{A}^{(Z)} = (0, 0, ...0, \gamma_{n+1})$.

It can be seen from the last expression that there are two main ways to reduce the error correction time T_{cor}. The first way is to reduce the number of stages k in determining AS. This is achieved through the use of the developed error correction methods in the MNS [9]. The second way is to reduce the zeroing time T_Z. This can be achieved by using the method of pairwise zeroing of numbers with a preliminary sample of digits [11].

Let's consider the necessary and sufficient condition for error correction in the dynamics of the computational data processing process.

On Figure 1 schematically shows the contraction (pull together) process of the AS of numbers in the process of processing CS information, where:

Δt_i – the duration of the implementation of the *i*-th operation of the information processing cycle;

S – the number of the operations in the considered information processing cycle;

S' – the number of the operation in the CS information processing cycle, at which the presence of errors is recorded;

t_0 – start time of the information processing cycle;

t_k – end time of the information processing cycle;

Δt_{AS_i} – the duration of the determination of the AS of the *i*-th number to be checked;

$\Delta t_{\wedge_i}$ – the duration of the determination of the *i*-th CAS;

t' – the moment of time of the error detection;

Δt_{Δ_i} – the duration of time from the end of finding the *i*-th CAS to the start of determining the next AS;

Δt_{det} – the error detection time;

k – the number of numbers to be checked (the number of stages in determining the AS).

Let in the process of monitoring the processing of information by the CS at time t', a distortion of the number $\tilde{A}$ is detected. In this case, the information processing process is not interrupted, and for the distorted number for the period Δt_{AS_1}, AS $W(\tilde{A}_1)$ is determined. After determining $W(\tilde{A}_1)$, after a period of time Δt_{Δ_1}, AS $W(\tilde{A}_2)$ is determined, where: $\tilde{A}_2$ – the result of the operation of the next cycle of information processing of the CS. After that, the CAS $W_{\wedge}(\tilde{A}) = W(\tilde{A}_1) \wedge W(\tilde{A}_2)$ is determined. If $W(\tilde{A}) \le 2$, then finding the AS stops. When $W(\tilde{A}) > 2$ the process of finding CAS continues. At the end of the information processing cycle ($t = t_k$), on the basis of the received AS $W(\tilde{A}) \le 2$, the result is corrected.

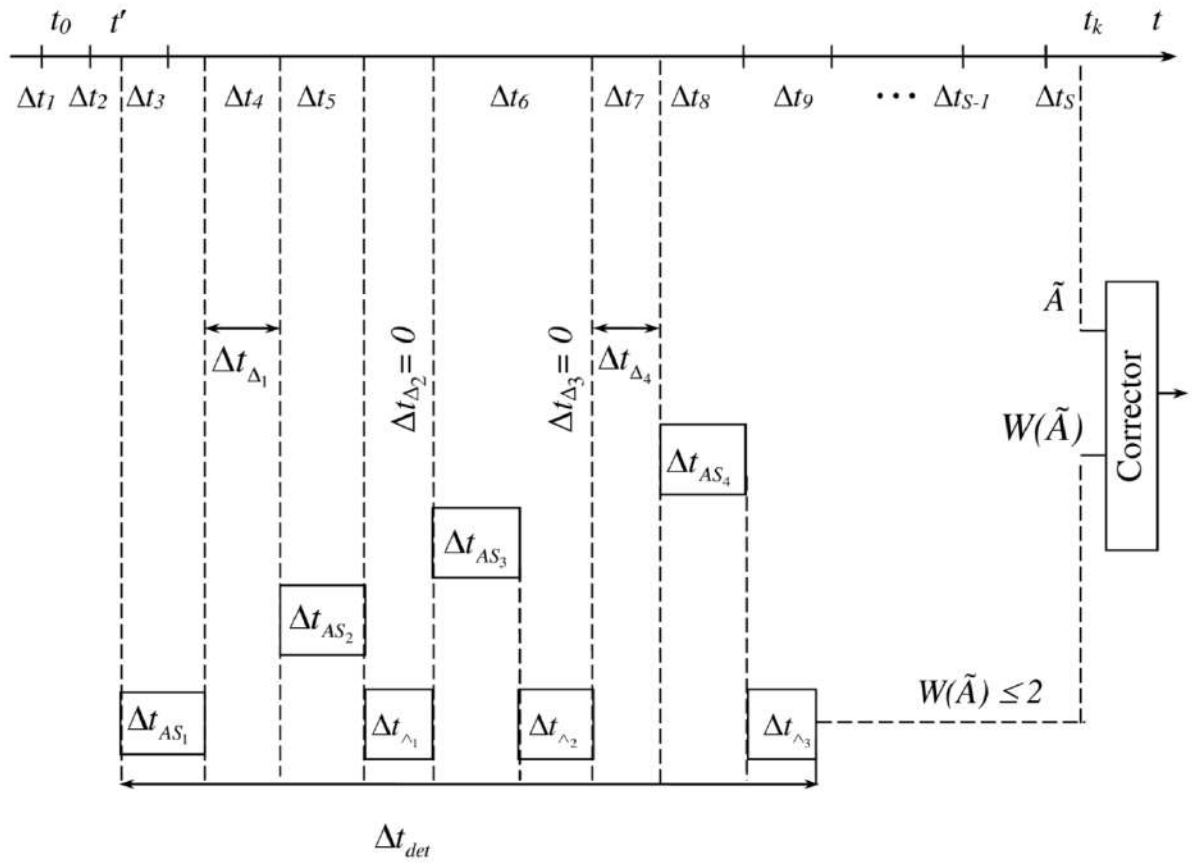


Figure 1: Error control and correction scheme in the MNS

In order to perform AS $W(\tilde{A})$ contraction to an erroneous base, it is necessary that the error detection time Δt_{det} be no more than the time from the moment the error was detected to the end of the AS information processing cycle, i.e.

$$\Delta t_{det} \le \Delta t'_c \tag{6}$$

where $\Delta t'_c$ – time from the moment the error was detected to the end of the AS information processing cycle.

The error detection time is determined according to the expression:

$$\Delta t_{det} = \sum_{i=1}^{k} \Delta t_{AS_i} + \sum_{i=1}^{k-1} \Delta t_{\wedge_i} + \sum_{i=1}^{k-1} \Delta t_{\Delta_i} \tag{7}$$

The time from the moment of error detection to the end of the AS information processing cycle is determined according to the expression:

$$\Delta t'_c = t_k - t' = \sum_{i=S'}^{S} \Delta t_i \tag{8}$$

Condition (6) is necessary and sufficient for $W(\tilde{A}) \le 2$.

Thus, when correcting errors in the dynamics of the information processing process, it is assumed that the implemented chain of operations has a sufficient length to allow the CAS to be pulled to one erroneous base.

There are three options for error correction.

1. In the course of information processing, the CAS are sequentially determined and, over time Δt_{det}, are pulled together to an erroneous base.

2. For the first AS, a certain hypothesis about the number of the distorted residue is accepted. The result is corrected until the fact of the erroneousness of the accepted hypothesis is discovered. In this case, it is necessary to move on to another hypothesis, and so on before the discovery of a distorted residue.

3. The third option consists of a synthesis of the first and second. The CAS of the numbers obtained as the information processing program is implemented up to their contraction (pull together) in the number $\tilde{A}$ to two bases m_i and m_{n+1} are determined. Further, the hypothesis of the error of the residue $\tilde{a}_i$ is accepted and its correction is carried out. If the hypothesis turns out to be untenable, then the residue $\tilde{a}_{n+1}$ will be wrong.

Thus, with any existing version of error correction in the dynamics of the information processing process, it becomes necessary to determine the AS, i.e. the effectiveness of any of the possible options for correcting errors depends on the method of determining the alternative set of numbers [12].

As a rule, the duration of the information processing cycle, for solving this algorithm, is a constant value $t_k - t_0 = const$. In this regard, to strengthen the fulfillment of condition (6), it is necessary to strive to reduce the time Δt_{det}. This can be achieved by performing the following operations.

1. Creation at the beginning of the information processing cycle of complicated modes of operation of the CS [13], which can lead to a shift of the time axis (error detection time) to the left to t_0. In other words, the error is detected almost immediately after the start of the information processing cycle. However, this path does not guarantee high reliability of error detection at the beginning of the data processing cycle.

2. Reducing the time $\sum_{i=1}^{k-1} \Delta t_{\Delta_i}$ as a result of splitting the operations of the information processing cycle into shorter ones. However, in most cases, splitting operations is either impossible or impractical.

3. Reducing the time $\sum_{i=1}^{k-1} \Delta t_{\wedge_i}$ as a result of the acceleration of the process of logical multiplication $W(\tilde{A}_i) \wedge W(\tilde{A}_{i+1})$. As a rule, the CAS definition block is built according to the tabular principle. The result of the operation is determined in one clock cycle of the CS.

4. Reducing the time $\sum_{i=1}^{k} \Delta t_{AS_i}$ as a result of increasing the performance of the zeroing operation [14].

5. A decrease in the number k of the numbers to be checked as a result of an increase in the information content $W(\tilde{A})$, i.e. reduction in the AS of the number of bases on which an error is possible.

Thus, studies are necessary and relevant for the development of effective methods for determining AS, which will increase the information content of $W(\tilde{A})$, which reduces the error correction time in the MNS.

3. Methods for determining an alternative set of numbers

Let's consider two methods for determining AS $W(\tilde{A})$.

The first method is that AS $W(\tilde{A})$ is established by checking each of the bases $m_i (i = \overline{1,\ n+1})$ as follows. A sequence of numbers is determined that has the same digits in all bases as the number $\tilde{A}$, except for the base m_ρ, differing only in digits in this base, i.e. numbers of the form:

$$A_{\rho_s} = (a_1, a_2, ...a_{\rho-1}, s, a_{\rho+1}, a_{n+1}) \qquad (9)$$

where $s = (\overline{1,\ m_\rho - 1})$.

Among the numbers of the form (9) there may not be a single correct number, or there may be only one correct number. In the latter case, m_ρ enters the AS of the number $\tilde{A}$. Having carried out similar checks for each of the bases of the MNS, should determine $W(\tilde{A}) = \left\{ m_{i_1}, m_{i_2}, ..., m_{i_k} \right\}$.

The disadvantage of the first method is the large hardware and time costs for determining the AS.

In the second method, the number $\tilde{A}$ is reduced to the form $\tilde{A}^{(Z)} = (0,0,...0,\gamma_{n+1})$ i.e. the so-called operation of zeroing the number $\tilde{A}$ is performed. In accordance with the error distribution theorem, the number of the interval $(j+1)$, in which the number $\tilde{A}$ falls, is determined by the expression:

$$j = \left[\frac{\Delta a_i \cdot \overline{m}_i \cdot m_{n+1}}{m_i} \right] \bmod m_{n+1} + \Delta^* \qquad (10)$$

where Δa_i – the value of a possible error in base m_i;

$\overline{m}_i$ – weight of the orthogonal basis B_i;

$[x]$ – the integer part of the number x, not exceeding the value of x.

Δ^* – takes the value 0 or 1.

In accordance with expression (10), a table of values of the correspondence of the number γ_{n+1} to possible errors Δa_i is compiled, where $j = (\gamma_{n+1} \cdot \overline{m}_{n+1}) \bmod m_{n+1}$. The desired AS is determined from the correspondence table.

4. Method of time numerical sections

The considered second method for determining an alternative set of numbers is called the method of time numerical sections, and in comparison with the first method, it allows to reduce hardware and time costs in determining the AS, however, the disadvantage of the second method is that the AS may contain redundant bases. This is due to the fact that the values of γ_{n+1} correspond to errors Δa_i related not only to the distorted number $\tilde{A}$, but also to the group of numbers $\tilde{A}_k$ lying in the interval $\left[j\frac{M_1}{m_{n+1}}, (j+1)\frac{M_1}{m_{n+1}} \right]$, where $M_1 = \prod_{i=1}^{n+1} m_i$ – the full range (including control base m_{n+1}) of numbers represented in the MNS. Excessive bases contained in AS reduce the information content of $W(\tilde{A})$.

Indeed, with an increase in the number of bases in the AS, the entropy of error detection in one of the bases of the MNS increases. An increase in the entropy of determining the erroneous base m1 increases the number k of the numbers to be checked (check cycles), and this, in turn, increases the time of contraction (pull together) of the AS to the erroneous base. This method cannot be effectively applied in a short chain of CS data processing. Thus, there is a need to develop procedures for determining the AS, with the help of which it is possible to effectively correct errors in a fairly short chain of the information processing process in the CS. Let us consider the procedure for increasing the information content of the AS in the MNS, based on obtaining additional information about the possible distorted residues of the wrong number $\tilde{A}$. This information is contained in all possible ASs of number $\tilde{A}$.

Let MNS be given by ordered bases $m_1,...,m_{n+1}$ and let the wrong number $\tilde{A}$ be determined in the process of information processing. To increase the information content about the location and magnitude of the error [15], it is proposed to additionally determine the AS of number of the form $W_{k_{\rho_i}}(\tilde{A}) = \left\{ m_{k_1}, m_{k_2}, ..., m_{k_{\rho_i}} \right\}$, i.e. set of AS of the form:

$$\begin{gathered} W_{1_{\rho_1}}(\tilde{A}) = \left\{ m_{1_1}, m_{1_2}, ..., m_{1_{\rho_1}} \right\} \\ W_{2_{\rho_2}}(\tilde{A}) = \left\{ m_{2_1}, m_{2_2}, ..., m_{2_{\rho_2}} \right\} \\ ... \\ W_{n+1_{\rho_{n+1}}}(\tilde{A}) = \left\{ m_{n+1_1}, m_{n+1_2}, ..., m_{n+1_{\rho_{n+1}}} \right\} \end{gathered} \qquad (11)$$

To determine the set of values (11), first should calculate the number of the interval j_k $(k = \overline{1,\ n+1})$ where the number $\tilde{A}$ hits:

$$j_k = (\gamma_k \cdot \overline{m}_k) \bmod m_k \quad (12)$$

Note that for $k = n+1$ the equality $W_{n+1_{\rho_{n+1}}}(\tilde{A}) = W(\tilde{A})$ is satisfied. In accordance with expression (12), let's compile k tables, where the values of γ_k are compared to the value of Δa_i. After ASs $W_{k_{\rho_i}}(\tilde{A})$, which called primary, are determined, define secondary ASs in the form of vectors, the components of which are possible error values Δa_i of the form: $W_1^1(\tilde{A}) = \{\Delta a_1^{(1)}, \Delta a_2^{(1)}, ..., \Delta a_{n+1}^{(1)}\}$, …, $W_1^{(\varphi_1)}(\tilde{A}) = \{\Delta a_1^{(\varphi_1)}, \Delta a_2^{(\varphi_1)}, ..., \Delta a_{n+1}^{(\varphi_1)}\}$, $W_2^2(\tilde{A}) = \{\Delta a_1^{(2)}, \Delta a_2^{(2)}, ..., \Delta a_{n+1}^{(2)}\}$, …, $W_2^{(\varphi_2)}(\tilde{A}) = \{\Delta a_1^{(\varphi_2)}, \Delta a_2^{(\varphi_2)}, ..., \Delta a_{n+1}^{(\varphi 2)}\}$ and so on up to the value of vectors of the form: $W_n^{(\varphi_n)}(\tilde{A}) = \{\Delta a_1^{(\varphi_n)}, \Delta a_2^{(\varphi_n)}, ..., \Delta a_{n+1}^{(\varphi_n)}\}$ and vector $W_{n+1}^{(\varphi_{n+1})}(\tilde{A}) = \{\Delta a_1^{(\varphi_n+1)}, \Delta a_2^{(\varphi_{n+1})}, ..., \Delta a_{n+1}^{(\varphi_n+1)}\}$.

The components of vector $W_{n+1}^{(\varphi_{n+1})}(\tilde{A})$ are compared with the corresponding components of all vectors $W_i^{(\varphi_i)}(\tilde{A})$ for $i = \overline{1, n}$. In the components of the vectors coinciding in magnitude, the MNS bases are determined, the set of which will determine the desired (resulting) AS of the form: $W'(\tilde{A}) = \left\{m_{z_1}, m_{z_2}, ..., m_{z_\rho}\right\}$. The AS $W_{k_{\rho_i}}(\tilde{A})$ always contains the base m_i, on which the error Δa_i occurred, and this base can only be among the bases common to the set (11), i.e.:

$$W(\tilde{A}) \geq W'(\tilde{A}) \quad (13)$$

where $W'(\tilde{A})$ – desired (resulting) AS.

If Δa_i is such that the number $\tilde{A} = A + \Delta A$ (where $\Delta A = (0, 0, ..., \Delta a_i, 0, ..., 0)$ is a single error) lies in the interval $\left[(m_{n+1} - 1)M, M_1\right]$, where $M = \prod_{i=1}^{n} m_i$ – the operating range of numbers represented in the MNS, then:

$$W(\tilde{A}) = W'(\tilde{A}) \quad (14)$$

Thus, the essence of the proposed procedure lies in the fact that all possible AS are determined on each of the intervals where the numbers A hit. After that, the common bases $m_{z_1}, ..., m_{z_\rho}$ for these intervals are determined, on which an error is possible. This set of bases determines the desired AS. Reducing the number of bases in AS increases the information content of AS $W(\tilde{A})$ about the place and magnitude of the error. This reduces the time of AS contraction to an erroneous base (reduces the number of stages for determining the CAS), which increases the efficiency of corrective codes in the MNS. The block diagram of the process of contraction of the AS to the erroneous base is shown in Figure 2. It is advisable to consider the block diagram of the process of contraction of the AS. Determining the number of the hit interval $(j+1)$ (under the influence of the error Δa_i) of the distorted number $\tilde{A}$ is equivalent to shifting this number in the interval $\left[j\frac{M_1}{m_i}, (j+1)\frac{M_1}{m_i} \right]$ to the left to the value $j\frac{M_1}{m_i}$. Let's divide the numerical segment $\left[0, M_1\right]$ into the corresponding intervals with duration: $\frac{M_1}{m_1}, \frac{M_1}{m_2}, ... \frac{M_1}{m_{n+1}}$. Let's determine the numbers of intervals $(j+1)$ in which the number $\tilde{A}$ is located on each of the numerical segments:

$$T_{j_1} = \left[j_1 \frac{M_1}{m_1}, (j_1 + 1)\frac{M_1}{m_1} \right]$$

$$\ldots \quad (15)$$

$$T_{j_{n+1}} = \left[j_{n+1} \frac{M_1}{m_{n+1}}, (j_{n+1} + 1)\frac{M_1}{m_{n+1}} \right]$$

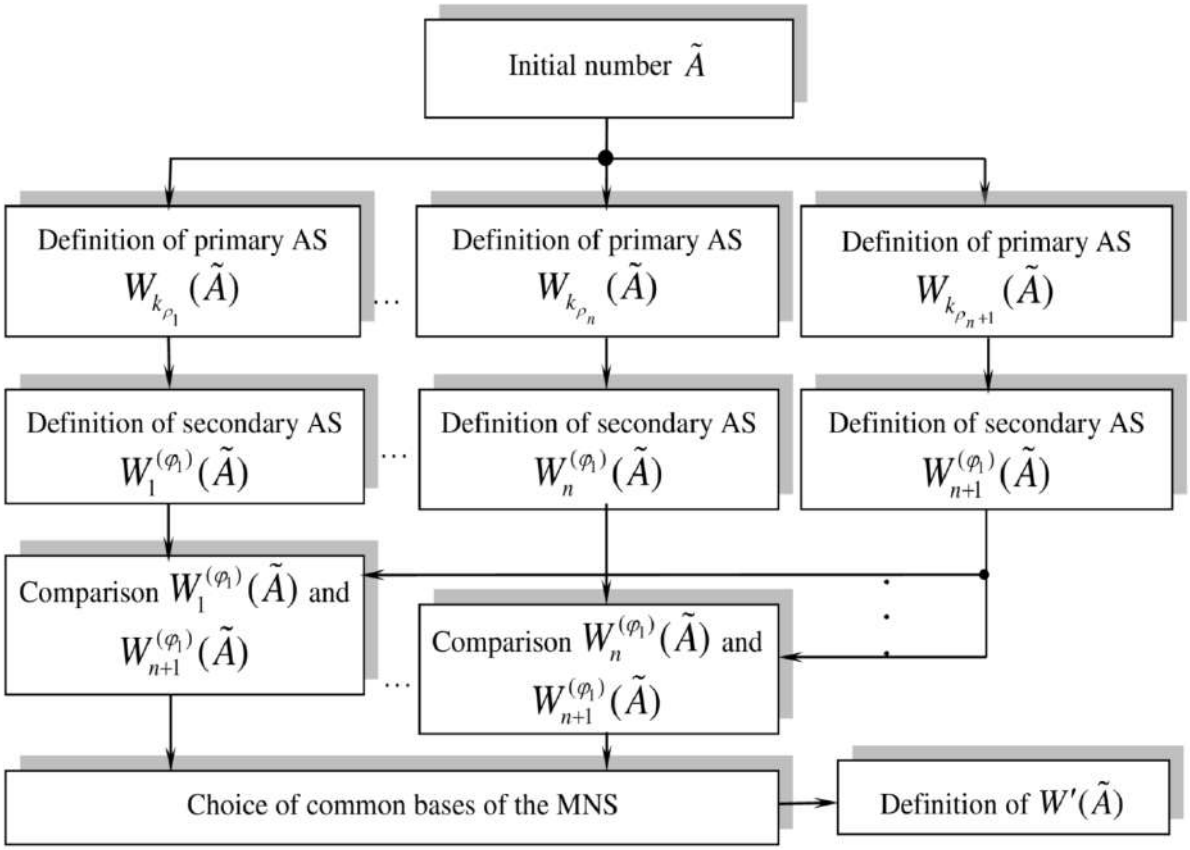


Figure 2: Structural diagram of the AS contraction (pull together) process

The definition of primary AS (11) corresponds to the definition the numbers of intervals (15). The definition of the secondary AS, geometrically corresponds to the definition of the interval $[z_1, z_2)$, where $z_1 = \max \in j_i \frac{M_1}{m_i}$. And $z_2 = \min \in (j_i + 1)\frac{M_1}{m_i}$, i.e. the desired interval is defined as the intersection of the sets of intervals (15) $T_{W'(\tilde{A})} = T_{j_1} \wedge T_{j_2} \wedge ... \wedge T_{j_{n+1}}$. It's obvious that:

$$z_2 - z_1 = \frac{M_1}{M_{n+1}} \tag{16}$$

Condition (16) is equivalent to condition (13). If the error translates the number $\tilde{A}$ into the interval $[(m_{n+1} - 1)M, M_1)$, then:

$$z_2 - z_1 = \frac{M_1}{M_{n+1}} = M \tag{17}$$

Condition (17) is equivalent to condition (14).

The presented structural diagram (Figure 2) of the AS contraction (pull together) process confirms the correctness of the mathematical description and more clearly demonstrates the essence of the procedure for increasing the information content of the AS – narrowing the interval for getting the distorted number $\tilde{A}$.

Consider an example of determining the AS of number $\tilde{A}$ in accordance with the developed procedure. Let the MNS be given by the bases $m_1 = 2,\ m_2 = 3,\ m_3 = 5$. Table 1 shows the code words of this MNS. Thus $M = 2 \cdot 3 = 6,\ M_1 = M \cdot m_3 = M \cdot 5 = 6 \cdot 5 = 30,\ m_{n+1} = m_3 = 5,\ A = (0,2,2)$ according to Table 1 this number in the positional number system is 2, let equals $\Delta A = (0,2,0)$ according to Table 1 this number in the positional number system is 20. Let, under the influence of a single error

$\Delta A = (0,0,...,\Delta a_i,...,0)$, based on the i-th base $(\Delta a_i = \Delta a_2 = 2)$ received number $\tilde{A} = A + \Delta A = (0,1,2)$.

To determine the set of primary AS, first determine the values of γ_k. To do this, let's zeroing of the number $\tilde{A}$ in accordance with the tables of zeroing constants (see Tables 2-4), obtain the values $\gamma_1 = 1,\ \gamma_2 = 1,\ \gamma_3 = 2$.

The set of primary AS is defined as: $W_{1_{\rho_1}}(\tilde{A}) = \{m_2, m_3\}$, $W_{2_{\rho_2}}(\tilde{A}) = \{m_1, m_3\}$, $W_{3_{\rho_3}}(\tilde{A}) = \{m_1, m_2, m_3\}$.

From Table 5-7, compiled according to the values γ_k, determine the set of secondary AS: for $\gamma_3 = 2$ has that $W_3^{(1)}(\tilde{A}) = \{1,1,2\}$; for $\gamma_2 = 1$ has that $W_2^{(1)}(\tilde{A}) = \{1,0,2\}$, $W_2^{(2)}(\tilde{A}) = \{0,0,3\}$; for $\gamma_1 = 1$ $W_1^{(1)}(\tilde{A}) = \{0,2,3\}$, $W_1^{(2)}(\tilde{A}) = \{0,0,4\}$.

Table 1
Table of code words in the MNS

A	m_1	m_2	m_3	A	m_1	m_2	m_3
0	0	0	0	15	1	0	0
1	1	1	1	16	0	1	1
2	0	2	2	17	1	2	2
3	1	0	3	18	0	0	3
4	0	1	4	19	1	1	4
5	1	2	0	20	0	2	0
6	0	0	1	21	1	0	1
7	1	1	2	22	0	1	2
8	0	2	3	23	1	2	3
9	1	0	4	24	0	0	4
10	0	1	0	25	1	1	0
11	1	2	1	26	0	2	1
12	0	0	2	27	1	0	2
13	1	1	3	28	0	1	3
14	0	2	4	29	1	2	4

Table 2
Table of zeroing constants

m_1	m_2
(1, 1, 1)	(0, 1, 4)
	(0, 2, 2)

Table 3
Table of zeroing constants

m_1	m_3
(1, 1, 1)	(0, 0, 1)
	(0, 2, 2)
	(0, 0, 3)
	(0, 1, 4)

Table 4
Table of zeroing constants

m_2	m_3
(0, 1, 0)	(1, 0, 3)
(1, 2, 0)	(0, 2, 2)
	(1, 1, 1)

Table 5
Table of zeroing constants of the secondary AS

γ_3	Errors	$W_i^{(\varphi_i)}$
0	no	–
1	$\square a_2 = 1,$ $\square a_3 = 1$	$W_3^{(1)}(\tilde{A}) = 0,1,1$
2	$\square a_1 = 1,$ $\square a_2 = 1,$ $\square a_3 = 2$	$W_3^{(1)}(\tilde{A}) = 1,1,2$
3	$\square a_1 = 1,$ $\square a_2 = 2,$ $\square a_3 = 3$	$W_3^{(1)}(\tilde{A}) = 1,2,3$
4	$\square a_2 = 2,$ $\square a_3 = 4$	$W_3^{(1)}(\tilde{A}) = 0,2,4$

Table 6
Table of zeroing constants of the secondary AS

γ_2	Errors	$W_i^{(\varphi_i)}$
0	$\square a_3 = 1$	$W_2^{(1)}(\tilde{A}) = \{0,0,1\}$
1	$\square a_1 = 1,$ $\square a_2 = 1,$ $\square a_3 = 1$	$W_2^{(1)}(\tilde{A}) = \{1,0,2\},$ $W_2^{(2)}(\tilde{A}) = \{0,0,3\}$
2	$\square a_3 = 4$	$W_2^{(1)}(\tilde{A}) = \{0,0,4\}$

Table 7
Table of zeroing constants of the secondary AS

γ_1	Errors	$W_i^{(\varphi_i)}$
0	$\square a_1 = 1,$ $\square a_2 = 2,$ $\square a_3 = 1$	$W_1^{(1)}(\tilde{A}) = \{0,0,1\},$ $W_1^{(2)}(\tilde{A}) = \{0,0,2\}$
1	$\square a_2 = 2,$ $\square a_3 = 3$	$W_1^{(1)}(\tilde{A}) = \{0,2,3\},$ $W_1^{(2)}(\tilde{A}) = \{0,0,4\}$

It is convenient to implement the choice of common MNS bases in the form of tables (see Tables 8-11), where the "+" sign indicates the coincidence of the components of the secondary AS, and the "-" sign indicates a mismatch. From Tables 8-11 it can be seen that the components of the vectors coincide in the bases m_1, m_3, i.e. the desired AS has the form $W'(\tilde{A}) = \{m_1, m_3\}$. Thus,

$W(\tilde{A}) \geq W'(\tilde{A})$. Therefore, the described procedure is guaranteed to increase information about the location of the error in number $\tilde{A}$.

In geometric interpretation, this procedure, for a given MNS, will be implemented as follows (Figure 3). Let's divide the segment $[0,30)$ into the corresponding numerical intervals $[15,30)$, $[10,20)$ and $[12,18)$. Determine the numbers of intervals in which the number $\tilde{A} = (0,1,2)$ is located: $T_{j_1} = [15,30)$, $T_{j_2} = [10,20)$, $T_{j_3} = [12,18)$. The desired interval is determined as follows $T_{W'(\tilde{A})} = [15,18)$.

The interval $T_{W'(\tilde{A})}$ is reduced compared to T_{j_3} by three units (by 50%), which leads to a reduction in the number of possible error options. This procedure is most effectively used in a chain of calculations that doesn't allow all the planned procedures to be carried out before the AS is contracted (pulled together) to an erroneous base, i.e. in a long chain of data processing CS [16-18].

Consider the procedure for probabilistic evaluation of the choice of the working hypothesis about the fallacy of the residue on an arbitrary *i*-th base. To do this, it is advisable to determine the relationship between the reduced error distribution coefficient $\xi_{k_i} = F(\gamma_{n+1})$ and the value of γ_{n+1}.

The reduced distribution coefficient of errors will be the ratio of the number of possible errors on the basis of m_k in the *i*-th interval to the number of possible errors of the number $\tilde{A}$ in the entire range $[0, M_1)$. It is numerically equal to the share of errors in the *i*-th interval according to the *k*-th base of the MNS.

Table 8

Table of zeroing constants

m_1	m_2	m_3
1	1	2
1	0	2
+	-	+

Table 9

Table of zeroing constants

m_1	m_2	m_3
1	1	2
0	0	3
-	-	-

Table 10

Table of zeroing constants

m_1	m_2	m_3
1	1	2
0	2	3
-	-	-

Table 11

Table of zeroing constants

m_1	m_2	m_3
1	1	2
0	0	4
-	-	-

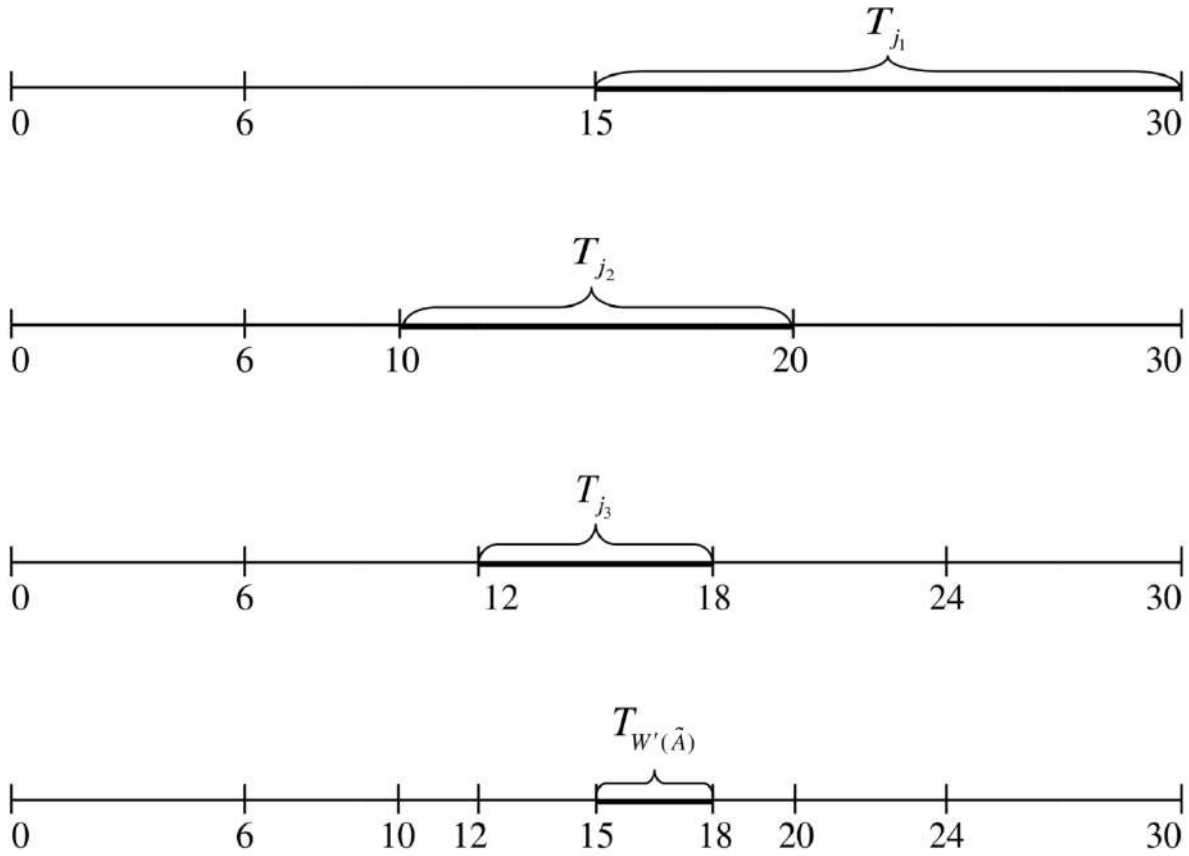


Figure 3: Selection scheme of time interval

When calculating ξ_{k_i}, it is necessary to consider the distribution of errors on the intervals $[jM,(j+1)M)$ for $j=1,2,...,m_{n+1}-1$. Based on the distribution, a table of values is compiled, according to which histograms n are built $\xi_{k_i}=F(\gamma_{n+1})$. The criterion for choosing a working hypothesis about the error of the residue based on m_z is the maximum value of the reduced error distribution coefficient, i.e.:

$$\xi_{z_i}=\max,\ \text{for}\ i=const \tag{18}$$

Let's compose an algorithm for determining an erroneous base when implementing the procedure for probabilistic evaluation of the choice of a working hypothesis.

1. The distorted number $\tilde{A}$ zeroable out. Getting the value of $\tilde{A}^{(Z)}=(0,0,...0,\gamma_{n+1})$.

2. According to the value of γ_{n+1}, determine AS $W(\tilde{A})=\left\{m_{l_1},m_{l_2},...m_{l_\rho}\right\}$.

3. By the value of γ_{n+1}, let's turn to the tables (histograms) $\xi_{k_i}=F(\gamma_{n+1})$. In the *i*-th interval $(i=(\gamma_{n+1}\cdot m_{n+1})\bmod m_{n+1})$, the largest of the values ξ_{z_i} is determined. Base m_z, for which the value of $\xi_{z_i}=\max$ at $i=const$ is the desired one. Thus, the working hypothesis is that the error is assumed in the residue to the base $m_z\in W(\tilde{A})$. If it turns out that the hypothesis is erroneous, then as the second working hypothesis should choose the base $m_i\in W(\tilde{A})$, for which $\xi_{k_i}<\xi_{z_i}$, and so on. As a rule, the choice of the primary working hypothesis according to the criterion (18) gives a reliable result about the location of the error.

As an example of choosing a working hypothesis, let's determine the number of the erroneous residue for the number $\tilde{A}=(0,1,2)$ specified in the MNS with bases $m_1=2$, $m_2=3$, $m_3=5$ $(m_{n+1}=m_3=5)$.

1. Zeroing of the number $\tilde{A} = (0,1,2)$ is performed, get $\tilde{A}^{(Z)} = (0,0,3)$.

2. According to the value of $\gamma_{n+1} = \gamma_3 = 3$, an appeal is made to Table 5, where AS $W_3^{(1)}(\tilde{A}) = \{1,2,3\}$ is defined.

3. Table 12 is compiled for $\xi_{k_i} = F(\gamma_3)$, in accordance with the distribution of errors over the intervals of the range $(0, M_1]$ (Figure 4). On Figure 4, the sign $\uparrow k$ shows that the error in the base m_k translates the correct number A into the wrong (distorted) number $\tilde{A}$. For the value $\gamma_3 = 3$ (fourth interval), according to the criterion (18), determine $\xi_{2_4} = 0,3$.

Table 12

Table of values $\xi_{k_i} = F(\gamma_3)$

γ_3	i	ξ_{1_i}	ξ_{2_i}	ξ_{3_i}	Working hypothesis of fallibility
0	1	0	0	0	-
1	2	0	0,25	0,75	m_2, m_3
2	3	0,23	0,3	0,47	m_2, m_3
3	4	0,23	0,3	0,47	m_2, m_3
4	5	0	0,25	0,75	m_2, m_3

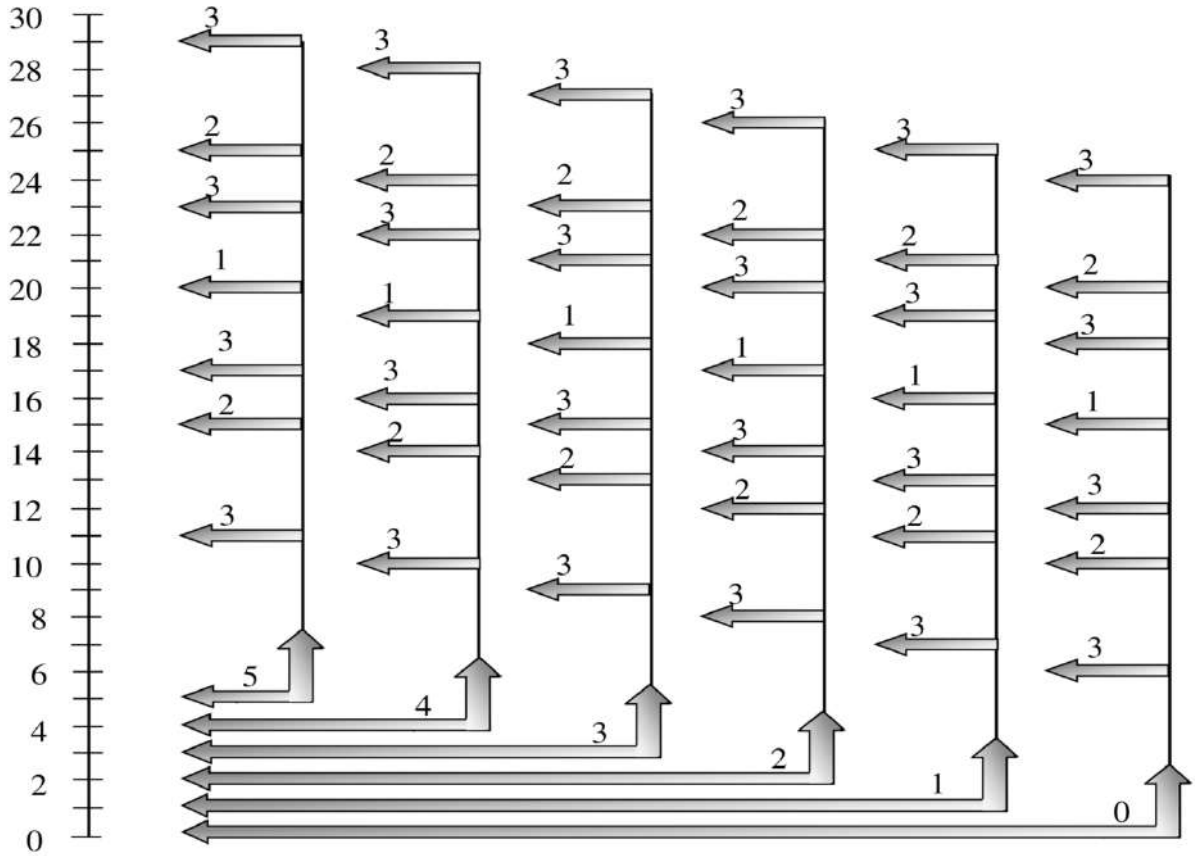


Figure 4: Scheme of distribution of errors over intervals of the numerical range $[0,30)$

Thus, in that the error is assumed in the residue of the base m_2 (an error in the residue of the control base m_3 is not taken into account). Verification confirms the correctness of the choice of hypothesis. So $A = \tilde{A} - \Delta A = (0,1,2) - (0,2,0) = (0,2,2)$, as a result it was received the original number (see Table 1, $A = (0,2,2)$ corresponds to the value of 2 positional number system, which was at the beginning).

The use of a probabilistic estimate makes it possible to reduce the number of check numbers, which in turn makes it possible to reduce the error correction time [19].

5. Conclusions

The article considers the process of monitoring, diagnosing and correcting errors in the MNS in the dynamics of the computational process of the CS. In some cases, this is possible in the presence of a minimum code distance by using the concepts of an alternative set of numbers and a conditional alternative set of numbers. A scheme for monitoring and correcting errors in the MNS has been developed. The necessary and sufficient condition for error correction in the dynamics of the data processing process is formulated and substantiated. When correcting errors in the dynamics of the information processing process, it is assumed that the implemented chain of computational operations has a sufficient length to allow the CAS to be reduced to one erroneous residue.

Three variants of error correction are defined. With any existing option for correcting errors in the dynamics of the data processing process, it becomes necessary to determine the AS, i.e. the effectiveness of any of the possible error correction options depends on the method for determining an alternative set of numbers. In this aspect, the article conducted research on the development of effective methods for determining the AS, which can increase the information content of the AS, which reduces the error correction time in the MNS. The analysis of methods for determining AS $W(\tilde{A})$ was carried out.

A procedure has been developed for monitoring and diagnosing errors in the CS based on a probabilistic assessment of the choice of a working hypothesis about the error of the residue based on the *i*-th base of the MNS. An algorithm for determining an erroneous base in the implementation of the procedure for probabilistic assessment of the choice of a working hypothesis has been implemented. This procedure allows, in some cases, to reduce the number of check numbers, which in turn reduces the error correction time.

It is shown that the proposed method for correcting errors in the MNS (the method of time numerical sections) makes it possible to simplify the implementation of the process of determining AS in the MNS and correct not only multiple errors in one residue, but also, in some cases, multiple errors in different residues. This method is most effectively used in a short data processing chain of the CS. An example of error correction for a specific MNS given by the bases $m_1 = 2,\ m_2 = 3,\ m_3 = 5$ is given. The results of the presented example confirm the main provisions of this article.

6. References

[1] Richard E. Blahut, Theory and Practice of Error Control Codes, 1st ed., Addison-Wesley, 1983.

[2] J. B. Anderson, S. Mohan, Error Control Coding, volume 150 of Source and Channel Coding, Springer-Verlag, Boston, MA, 1991, pp. 77–97. doi:10.1007/978-1-4615-3998-8_3.

[3] L. Huang, Y. Chen, H. Huang, Research of Error Control Coding and Decoding, in 2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, 2022, pp. 590–593. doi:10.1109/AEECA55500.2022.9919106.

[4] J. L. Massey, Theory and practice of error control codes, Proceedings of the IEEE 74 (1986) 1293–1294. doi:10.1109/PROC.1986.13626.

[5] I.Ya. Akushskii, D.I. Yuditskii, Machine Arithmetic in Residual Classes, Sov. Radio, Moscow, 1968.

[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] N.I. Chervyakov, A.V. Veligosha, 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.
[8] 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.
[9] V. Krasnobayev, S. Koshman, A. Yanko, A. Martynenko, Method of Error Control of the Information Presented in the Modular Number System, in: 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 2018, pp. 39–42. doi:10.1109/INFOCOMMST.2018.8632049.
[10] J.-C. Bajard, L. Imbert and T. Plantard, Arithmetic operations in the polynomial modular number system, in: 17th IEEE Symposium on Computer Arithmetic (ARITH'05), Cape Cod, MA, 2005, pp. 206–213. doi:10.1109/ARITH.2005.11.
[11] V. Krasnobayev, A. Yanko, A. Kuznetsov, O. Smirnov, T. Kuznetsova, Methods of nulling numbers in the system of residual classes, in: 1st International Workshop on Cyber Hygiene & Conflict Management in Global Information Networks, volume 2588, Kyiv, 2019. URL: http://ceur-ws.org/Vol-2588/paper9.pdf.
[12] 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.
[13] D. Schinianakis, T. Stouraitis, An RNS modular multiplication algorithm, in: 2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), Abu Dhabi, 2013, pp. 958–961. doi:10.1109/ICECS.2013.6815571.
[14] K Lee, J Chun, On the interference nulling operation of the V-BLAST under channel estimation errors, in: Proceedings IEEE 56th Vehicular Technology Conference, Vancouver, BC, 2002, pp. 2131–2135. doi: 10.1109/VETECF.2002.1040595.
[15] V. Ruzhentsev, R. Oliynykov, Properties of Linear Transformations for Symmetric Block Ciphers on the Basis of MDS-Codes, in: 2011 Conference on Network and Information Systems Security, La Rochelle, 2011, pp. 1–4. doi:10.1109/SAR-SSI.2011.5931391.
[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] E. Shiryaev, E. Bezuglova, M. Babenko, A. Tchernykh, B. Pulido-Gaytan, J. M. Cortés-Mendoza, Performance Impact of Error Correction Codes in RNS with Returning Methods and Base Extension, in: 2021 International Conference Engineering and Telecommunication (En&T), Dolgoprudny, 2021, pp. 1–5. doi:10.1109/EnT50460.2021.9681756.
[18] 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.
[19] 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.

Thermal Image Super-Resolution Methods Using Neural Networks

Andrii Didenko[a], Andrii Oliinyk[a] and Sergey Subbotin[a]

[a] *National University "Zaporizhzhia Polytechnic", Zhukovskoho street 64, Zaporizhzhia, 69063, Ukraine*

Abstract

Thermography has gained popularity in many fields. Since the human eye cannot see the thermal spectrum and in low light, thermal image analysis has become an integral part of medicine, manufacturing, construction, and other industries. Most thermal cameras produce low-resolution images when analyzing the temperature of objects, which complicates the process of analyzing original thermogram. Therefore, the problem of improving the quality of thermal images is relevant today. With the development of artificial intelligence and deep learning technologies, new super-resolution methods are emerging. Such methods can also be adapted for thermal image processing. This work examines the performance of modern super-resolution methods in the thermal vision domain.

Keywords

Machine learning, thermography, deep learning, thermal image, neural network, super-resolution, image processing, computer vision

1. Introduction

At all temperatures above absolute zero, every object emits energy from its surface in the form of a spectrum of different wavelengths and intensities. The radiation we call visible light, forms a very small part of the electromagnetic spectrum. The infrared or thermal wavelength region extends over the range 0.7 to 1000 μm [1]. Infrared radiation can carry a lot of useful information about an object that is studied. However, it is invisible to the human eye, thus it is important to have special equipment to analyze infrared data.

Infrared thermography (IRT) has a lot of applications in different fields. It is fast and non-invasive method of diagnostics that is widely used in medicine to detect abnormal body temperature, to diagnose breast cancer, diabetes neuropathy, etc. [2]. It is also used in the architectural and civil engineering fields [3]. In manufacturing, for example, a combination of IRT and deep learning helps to detect cracks in steel plates [4]. IRT also finds its application in the condition monitoring of electrical equipment using machine learning algorithms [5]. Moreover, it can be also used for field phenomics of different tree species using unmanned aerial vehicles (UAVs) [6].

Despite having a lot of applications in different fields, IRT has its own drawbacks. The main issue with thermal imaging is the resolution of the output thermogram. While modern RGB cameras are able to produce high-resolution images, the majority of thermal cameras produce images of low spatial resolution. Besides, thermal cameras that can produce high-resolution thermograms cost much more than regular ones and are not affordable to ordinary users. This creates a need for thermal image super-resolution methods.

The goal of Super-Resolution (SR) methods is to recover a high-resolution image from one or more low-resolution input images [7]. A high-resolution image provides a higher pixel density and thus more detail about the original scene. The need for high-quality images often arises in the field of computer

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: an232did@gmail.com (A. Didenko); olejnikaa@gmail.com (A. Oliinyk); subbotin@zntu.edu.ua (S. Subbotin)
ORCID: 0009-0009-9236-3936 (A. Didenko); 0000-0002-6740-6078 (A. Oliinyk); 0000-0001-5814-8268 (S. Subbotin)

vision to improve the efficiency of pattern recognition and image analysis. SR methods have been developing for decades and have come a long way from classical approaches to modern deep learning models.

The purpose of this work is the implementation and analysis of different modern SR methods in the thermal image domain.

2. Related Work

SR methods can be divided into several groups: interpolation methods, Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), and Transformers. Each of these methods has its own advantages and disadvantages but in practice, neural network based methods show better performance which usually overlaps their disadvantages.

2.1. Interpolation Methods

Interpolation methods are the simplest among SR methods. Among them are nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation [8]. In the nearest neighbor method, each interpolated output pixel is assigned the value of the nearest sample point in the input image while bilinear interpolation is used to know values at random positions from the weighted average of the four closest pixels to the specified input coordinates. Bicubic is an advancement over the above interpolation and uses polynomials, cubic, or cubic convolution algorithms [8]. These methods are available in every image processing software. Despite their ease of implementation and clarity, the main disadvantage of the interpolation methods is the quality of the output image, thus they are usually used as baselines for more advanced SR methods.

2.2. CNNs

CNN is the main type of neural network that is used for image processing. It also has found its application in SR tasks. For the first time, the idea of using CNNs instead of classical approaches was introduced in [9]. In this method, the image is first upscaled using bicubic interpolation, then processed through a network consisting of three parts: image patch extraction, nonlinear mapping and reconstruction. Patch extraction represents patches of image as a multidimensional vector of features. Nonlinear mapping transforms each multidimensional vector into another multidimensional vector. At the reconstruction stage, the image patch representations are aggregated to generate the original enlarged image [9].

Method [10] is the improvement of the previous method that speedups image super-resolution. Unlike the method [9], the authors do not enlarge the image with bicubic interpolation before passing it to the neural network but work with the original image. In addition, the size of the convolutional layer filter was reduced to 5. Then, the number of filters is reduced to reduce the training load. At the nonlinear mapping stage, several convolutional layers are used instead of one. The next stage increases the number of filters, which improves the quality of image processing. The last step is the deconvolution, which produces an enlarged image.

Method [11] uses neural network with autoencoder architecture for image restoration which includes SR task. The authors propose a deep fully convolutional auto-encoder network, which is an encoding-decoding framework with symmetric convolutional-deconvolutional layers. The network is composed of multiple layers of convolution and de-convolution operators, learning end-to-end mappings from corrupted images to the original ones [11].

The ideas of the method [9] found their usage in thermal image SR in [12]. The authors examined the use of datasets from different domains to train the model. The results show that a model trained to upscale regular RGB images is better at enhancing thermal images than a model trained on thermal images.

In [13], the authors also studied the influence of the dataset domain on the quality of training and model results. The authors conducted experiments in such color models as grayscale, HSL, HSI, and

HSV. In addition, the architecture of the model was also inspired by the method [9]. According to the results, the network trained by the gray outperformed the one that used the lightness and intensity domains, but the networks based on the brightness domain provided better performance compared to the gray-based network [13].

In [14], on contrary to the above methods, the authors showed that training a model on a dataset of thermal images gives better results than training on a dataset of RGB images.

In [15] authors use a progressive upscaling strategy with asymmetrical residual learning [15] and also compare their work to SR methods that are commonly used in RGB image SR task.

2.3. GANs

GANs [16], as its name suggests, originally were used to generate images. GANs consist of two models: a generative model *G* that captures the data distribution, and a discriminative model *D* that estimates the probability that a sample came from the training data rather than *G*. The training procedure for *G* is to maximize the probability of *D* making a mistake [16].

The architecture of GANs is widely used in SR tasks. For example, in [17] authors proposed a novel model for SR, called SRGAN. This method uses residual network called SrResNet as a generator. Besides, the authors propose a perceptual loss function which consists of an adversarial loss and a content loss.

An improvement of SRGAN is ESRGAN [18] which increases model performance and improves architecture and loss function of the previous method. In particular, they introduced a special block in generator model, called Residual-in-Residual Dense Block (RRDB). They also modified perceptual loss by using features before activation layers.

GANs found their application in thermal image SR task. For example, method [19] uses CycleGAN [20]. The generator is ResNet6 model and discriminator is PatchGAN, whole model is trained in unsupervised way. Besides, the authors also released dataset for training thermal image SR models, which consists of low-resolution, middle-resolution, and high-resolution thermal images.

2.4. Transformers

Recently, the Transformer neural network architecture has been increasingly used in deep learning solving different tasks. Transformer was first introduced for solving natural language processing (NLP) tasks [21], but then showed impressive performance in other domains. The core of Transfomer is the attention mechanism that helps neural network to memorize long sequences and focus on specific parts of the input.

In [22], the authors propose to use a Transformer, namely an encoder, to classify images, while preserving the original architecture as much as possible. This architecture is called a Vision Transformer (ViT). To do this, the image is divided into several patches (in this article, a patch is 16x16 pixels), so a sequence of patches is fed to the transformer's input. To preserve information about the location of the patches, information about the position of these patches relative to each other (positional coding) is added to the input sequence. In order for the model to learn to classify images, a representation of the image class to be learned during model training is also added to the input sequence. ViT shows better results in comparison to CNN models.

Despite having good performance, ViT has several important drawbacks. One of these drawbacks is the processing of high-resolution images, as the computational complexity increases quadratically with image size. In addition, the architecture of a ViT is poorly suited for solving other computer vision tasks, such as segmentation, since in this task it is important to distinguish image features at different scales.

To solve these problems Shifted-window Transformer (Swin Transformer) [23] was introduced. Swin Transformer has linear computation complexity as it computes self-attention not within every patch of the image but within patches in the local window. Then, to compute connections between each window, shifted window attention is used. It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks [23].

Transformers can be also used for solving SR tasks. For example, SwinIR [24] is a transformer-based image restoration model mostly inspired by Swin Transformer. SwinIR consists of three parts: shallow feature extraction, deep feature extraction, and HQ image restoration. In turn, deep feature extraction block is a stack of special residual Swin Transformer blocks (RSTB) [24]. This architecture makes it possible to achieve excellent results in different image restoration tasks, such as SR, denoising, and artifact reduction.

3. Method

As each method has its own advantages and disadvantages, it was decided to conduct experiments with classical algorithms as baseline, CNNs, GANs and Transformer-based neural networks and then compare the results of these methods.

In particular, in this paper following neural networks are examined: SRGAN, SrResNet (generator of SRGAN), ESRGAN, RRDBNet (generator of ESRGAN), SwinIR on thermal images and RGB images domains, and SwinIRGAN (combination of SwinIR as generator and SRGAN discriminator).

4. Experiments

Each of the selected models was already pretrained on its own domain. Also, models were trained and evaluated with the help of BasicSR [25] – deep learning framework for solving image restoration tasks, such as denoising and SR. Each model was trained to upscale thermal images by a factor of 2.

4.1. Dataset

Since the task of thermal image SR is quite specific, the number of open-source datasets for this task is quite low.

One of the largest and most popular is the FLIR dataset [26]. This dataset offers annotated thermal and RGB images for object detection systems. The dataset contains more than 26000 annotated frames and 15 different object categories. In total, the dataset contains almost 10,000 thermal and more than 9,000 RGB images with a size of 640 by 480 pixels [26]. Figure 1 shows an example of data from FLIR dataset.

Figure 1: Example data from FLIR [26] dataset

One of the most popular datasets for solving the RGB SR problem is DIV2K [27]. This dataset contains 1000 high-resolution images (1500-2000 pixels per image side) divided into training, test and validation sets. This dataset also contains reduced images by a factor of 2, 3, 4, and 8.

In this work, both of these datasets are used. In particular, DIV2K is used to examine the performance of model trained on grayscale images of this dataset in thermal image domain.

4.2. Data Preprocessing

To use FLIR dataset for the thermal image SR task, only thermal images (single channel) were selected and a version of the dataset with thermal images reduced by a factor of 2 (320 by 240 pixels) was created.

In order to use the DIV2K dataset for the thermal image SR task, a version of the dataset with images in grayscale format was created. Thus, images have one channel with a pixel range of 0-255.

4.3. Evaluation

To evaluate trained models SSIM [28] (structural similarity index measure) and RSNR (peak signal-to-noise ratio) metrics were used. SSIM is defined as follows (1):

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (1)$$

where x, y are windows of image of size $N \times N$;
μ_x – mean of x;
μ_y – mean of y;
σ_x – variance of x;
σ_y – variance of y;
c_1, c_2 – stabilization variables.
PSNR is defined as follows (2):

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right), \quad (2)$$

where MAX_I is maximum pixel value of image I;
MSE – mean squared error.

4.4. Experiment Setup

Before conducting experiments, a set of hyperparameters was chosen.

Adam [29] with a learning rate 0.0002 was chosen as an optimization algorithm for all models. All models were trained for 100000 iterations. The number of images in one batch is 16 for SwinIR, ESRGAN and RRDBNet and 64 for SRGAN and SrResNet.

All models were trained not on whole images, but on square patches that were randomly selected from each image. This speeds up training and forces the neural network to pay attention to high-frequency features. Thus, for all models, a patch of 128×128 pixels was chosen, except for the classic SwinIR model (96×96 pixels).

4.5. Results

The quantitative comparison is listed in Table 1.

Table 1
Results of thermal image super-resolution

Model	SSIM	PSNR
Nearest Neighbors	0,7389	30,3350
Bilinear Interpolation	0,7557	31,1910
Bicubic Interpolation	0,7672	31,5820
SRGAN	0,6617	28,3197
SwinIRGAN	0,6751	29,6528
ESRGAN	0,6853	29,8002
SwinIR (DIV2K)	0,7642	31,4355
RRDBNet	0,7829	32,3622
SRResNet	0,7828	32,3471
SwinIR (FLIR)	**0,7833**	**32,3688**

Qualitative results are shown in Figure 2, Figure 3 and Figure 4.

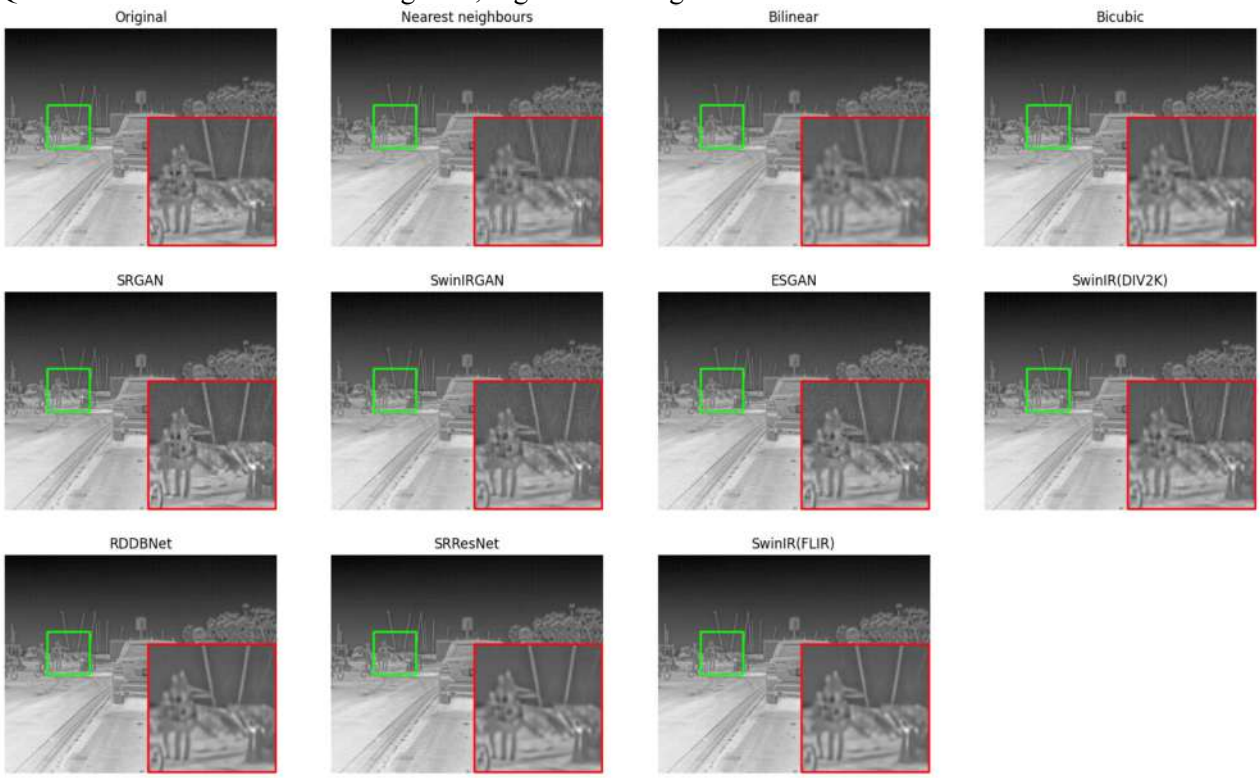


Figure 2: Qualitative comparison on FLIR sample (small objects), scaling factor ×2

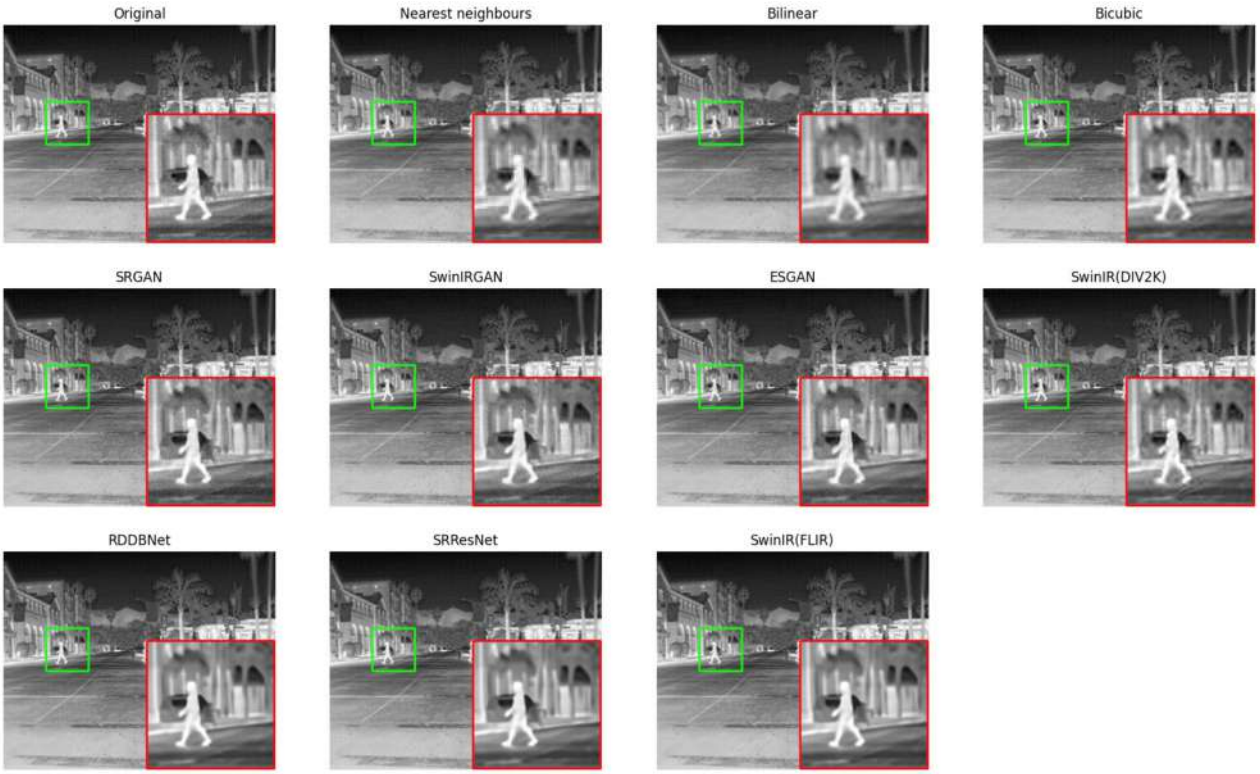


Figure 3: Qualitative comparison on FLIR sample (pedestrian), scaling factor ×2

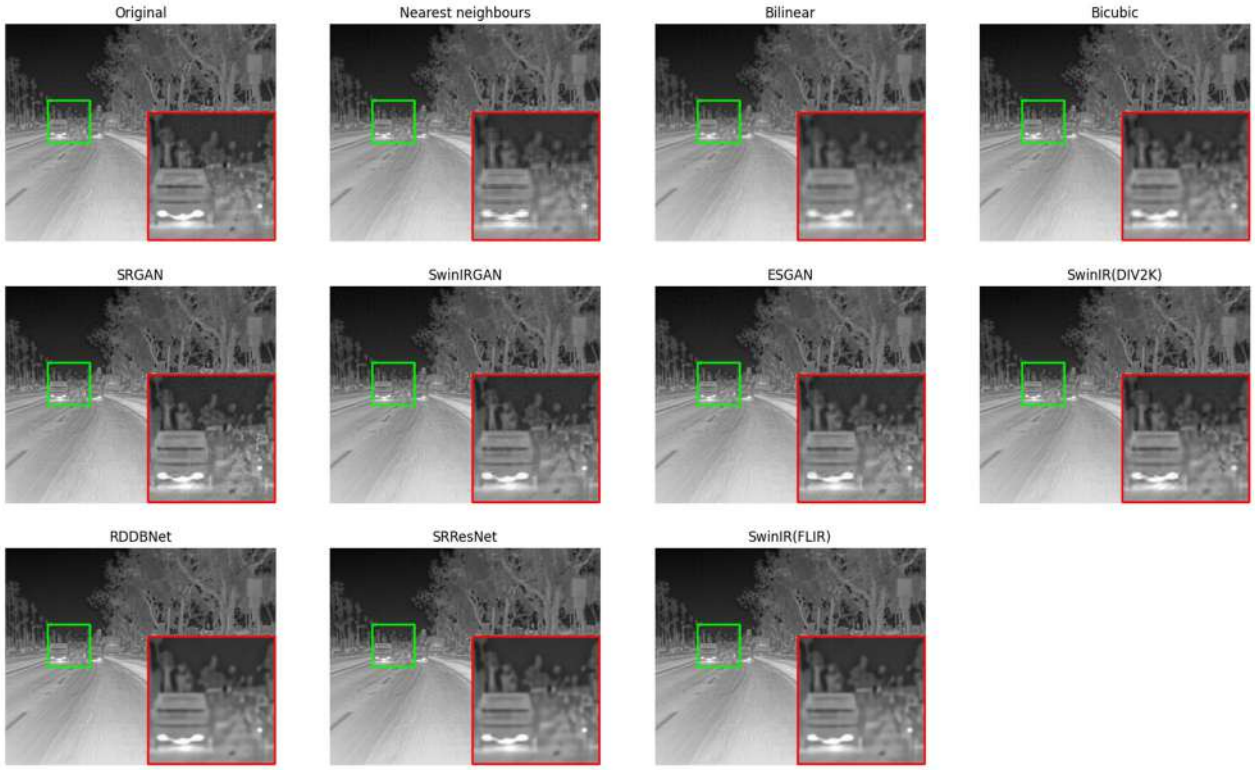


Figure 4: Qualitative comparison on FLIR sample (car), scaling factor ×2

Results of SSIM and PSNR metrics during training are shown in Figure 5 and Figure 6 respectively.

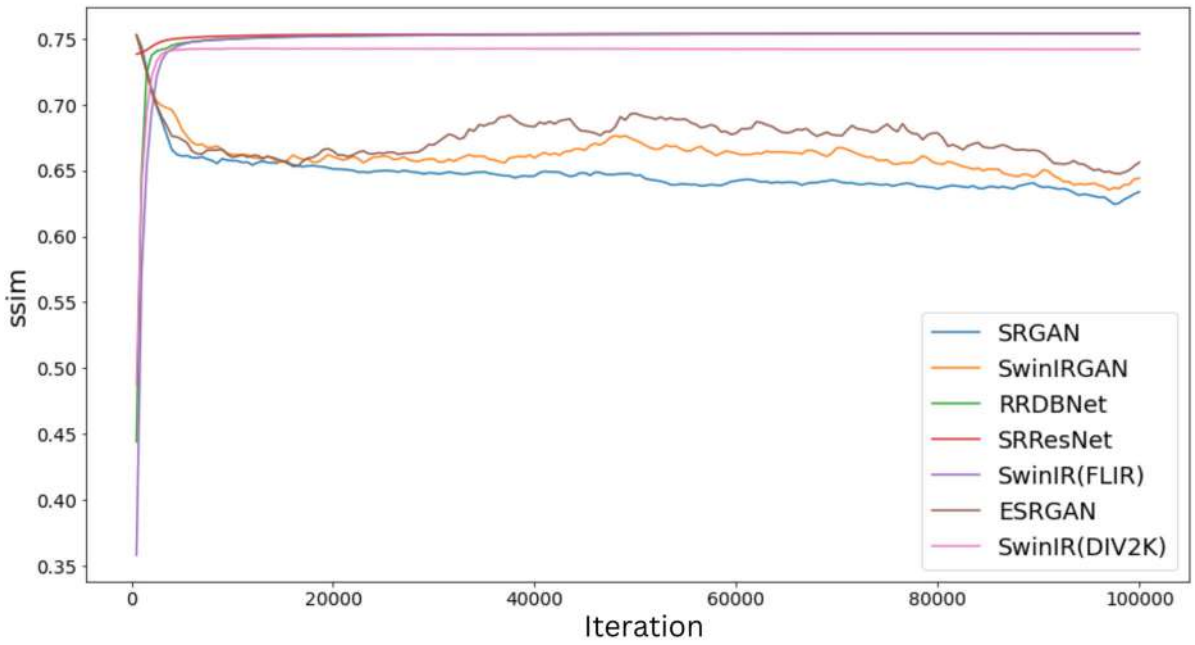


Figure 5: SSIM metric on the validation set during training

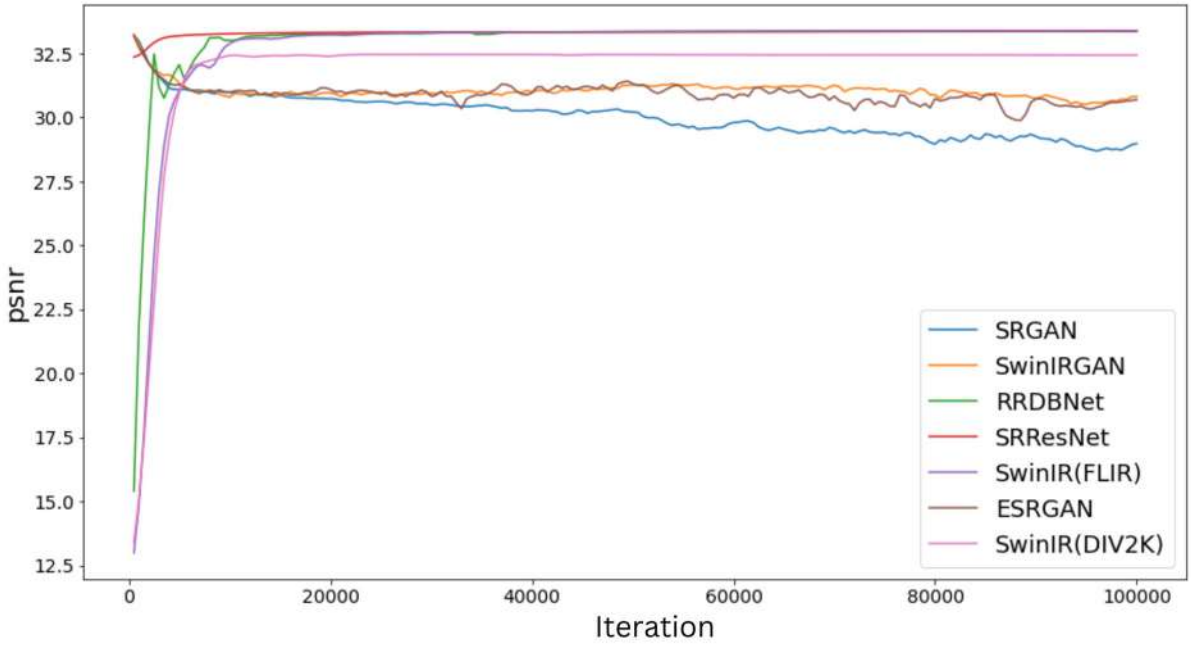


Figure 6: PRNR metric on the validation set during training

From the obtained results, it can be concluded that SwinIR trained on FLIR dataset shows the best results among other models. Additionally, CNNs such as SrResNet and RRDBNet trained separately are better than these CNNs trained as GANs. As it can be seen from qualitative results, GANs have a lot of noise on upscaled images which causes bad quantitative results, while CNNs and Transformers almost completely denoise upscaled images. It can be explained by the fact that GANs are sensitive to the input data and selected training hyperparameters. It is also visible that SwinIR trained on FLIR dataset performs better than SwinIR trained on the grayscale DIV2K dataset.

5. Conclusion

This paper implements modern SR methods and examines their results on thermal image SR problem. The aim of this paper was to analyze the performance of popular SR methods on upscaling of thermal images.

The main problem of this task is the lack of datasets for thermal image SR. Current datasets don't have enough training data, the size of thermal images in dataset is not as big as in RGB datasets for SR tasks, and quality of thermal images is worse than quality of RGB images in training sets.

According to the obtained results of thermal image upscaling it was concluded that Transformer and CNN models can perform better than GANs and classical algorithms.

In addition, the following steps can be taken to improve the quality of SR of thermal images: increase the size of the dataset and training batch to improve the generalization ability of the models; collect dataset of thermal images with higher quality and lower percentage of noise in the images to improve the results of SR methods; tune training hyperparameters such as optimization algorithm, learning rate, loss function, etc.; improve the architecture of used models or modify them for the task of thermal image SR.

6. References

[1] J. M. Hart A practical guide to infra-red thermography for building surveys, Building Research Establishment, Garston, 1991.

[2] B.B. Lahiri, S. Bagavathiappan, T. Jayakumar, John Philip, Medical applications of infrared thermography: A review, Infrared Physics & Technology, vol. 55(4), (2012) pp. 221-235. doi: 10.1016/j.infrared.2012.03.007.

[3] C. Meola, Infrared Thermography in the Architectural Field, The Scientific World Journal, vol. 2013, (2013). doi: 10.1155/2013/323948.

[4] J. Yang, W. Wang, G. Lin, Q. Li, Y. Sun and Y. Sun, Infrared Thermal Imaging-Based Crack Detection Using Deep Learning, IEEE Access, vol. 7, (2019) pp. 182060-182077. doi: 10.1109/ACCESS.2019.2958264.

[5] M. Najafi, Y. Baleghi, S. A. Gholamian and S. Mehdi Mirimani, Fault Diagnosis of Electrical Equipment through Thermal Imaging and Interpretable Machine Learning Applied on a Newly-introduced Dataset, 2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), (2020) pp. 1-7. doi: 10.1109/ICSPIS51611.2020.9349599.

[6] R. Ludovisi, F. Tauro, R. Salvati, S. Khoury, G. Scarascia Mugnozzaa, A. Harfouche, UAV-Based Thermal Imaging for High-Throughput Field Phenotyping of Black Poplar Response to Drought, Front. Plant Sci, vol. 8, (2017) doi: 10.3389/fpls.2017.01681.

[7] D. Glasner, S. Bagon, M. Irani, Super-resolution from a single image, 2009 IEEE 12th International Conference on Computer Vision, (2009) pp. 349-356. doi: 10.1109/ICCV.2009.5459271.

[8] S. Fadnavis, Image Interpolation Techniques in Digital Image Processing: An Overview, International Journal Of Engineering Research and Application, vol. 4(10), (2014) pp. 70-73

[9] C. Dong, C. C. Loy, K. He and X. Tang, Image Super-Resolution Using Deep Convolutional Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38(2), (2016) pp. 295-307. doi: 10.1109/TPAMI.2015.2439281.

[10] C. Dong, C. C. Loy, and X. Tang, Accelerating the super-resolution convolutional neural network, European Conference on Computer Vision, vol. 9906, (2016) pp. 391–407. doi: 10.1007/978-3-319-46475-6_25.

[11] X. Mao, C. Shen, and Y. Yang, Image restoration using convolutional auto-encoders with symmetric skip connections, Advances in Neural Information Processing Systems, (2016).

[12] Y. Choi, N. Kim, S. Hwang and I. S. Kweon, Thermal Image Enhancement using Convolutional Neural Network, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2016) pp. 223-230. doi: 10.1109/IROS.2016.7759059.

[13] K. Lee, Junhyeop Lee, Joosung Lee, S. Hwang, and S. Lee, Brightness-based convolutional neural network for thermal image enhancement, IEEE Access, vol. 5, (2017) pp. 26867–26879. doi: 10.1109/ACCESS.2017.2769687.

[14] R. E Rivadeneira, P. L. Suarez, A. D. Sappa, and B. X. Vintimilla, Thermal image superresolution through deep convolutional neural network, International Conference on Image Analysis and Recognition, (2019) pp. 417–426. doi: 10.1007/978-3-030-27272-2_37.
[15] V. Chudasama et al., TherISuRNet - A Computationally Efficient Thermal Image Super-Resolution Network, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2020) pp. 388-397. doi: 10.1109/CVPRW50498.2020.00051.
[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, Advances in Neural Information Processing Systems (NIPS), vol. 3(11), (2014) pp. 2672–2680. doi: 10.1145/3422622.
[17] C. Ledig, L. Theis, F. Husz´ar, et al, Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017) pp. 105-114. doi: 10.1109/CVPR.2017.19.
[18] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks, European Conference on Computer Vision 2018 Workshop, vol. 11133 (2019). pp. 63-79. doi: 10.1007/978-3-030-11021-5_5.
[19] R. E. Rivadeneira, A. D. Sappa, and B. X. Vintimilla, Thermal image super-resolution: a novel architecture and dataset, International Conference on Computer Vision Theory and Applications, (2020) pp 1–2. doi: 10.5220/0009173601110119.
[20] J. -Y. Zhu, T. Park, P. Isola and A. A. Efros, Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks, 2017 IEEE International Conference on Computer Vision (ICCV), (2017) pp. 2242-2251. doi: 10.1109/ICCV.2017.244.
[21] Vaswani, A.; Shazeer, N. M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need, Proceedings of the 31st International Conference on Neural Information Processing System, (2017) pp. 6000–6010. doi: 10.48550/arXiv.1706.03762.
[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, International Conference on Learning Representations, (2021). doi: 10.48550/arXiv.2010.11929.
[23] Z. Liu, Y. T. Lin, Y. Cao; H. Hu, B. N. Guo, Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), (2021) pp. 9992-10002. doi: 10.1109/ICCV48922.2021.00986.
[24] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool and R. Timofte, SwinIR: Image Restoration Using Swin Transformer, 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), (2021) pp. 1833-1844. doi: 10.1109/ICCVW54120.2021.00210.
[25] Xintao Wang, Liangbin Xie, Ke Yu, Kelvin C.K. Chan, Chen Change Loy and Chao Dong. BasicSR: Open Source Image and Video Restoration Toolbox, 2022. URL: https://github.com/xinntao/BasicSR.
[26] FREE Teledyne FLIR Thermal Dataset for Algorithm Training. URL: https://www.flir.eu/oem/adas/adas-dataset-form
[27] E. Agustsson and R. Timofte, NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study, 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2017) pp. 1122-1131. doi: 10.1109/CVPRW.2017.150.
[28] Z. Wang, E. P. Simoncelli and A. C. Bovik, Multiscale structural similarity for image quality assessment, The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, vol. 2, (2003) pp. 1398-1402. doi: 10.1109/ACSSC.2003.1292216.
[29] D. P. Kingma and J. Lei Ba, Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), (2015), pp 1–13. doi: 10.48550/arXiv.1412.6980.

Efficient Multithreading Computation of Modular Exponentiation with Pre-computation of Residues for Fixed-base

Ihor Prots'ko[a], Aleksandr Gryshchuk[b], Volodymyr Riznyk[a]

[a]*Lviv Polytechnic National University, str. S.Bandery, 12, Lviv, 79013, Ukraine*
[b]*LtdC "SoftServe", str. Sadova, 2d, Lviv, 79021, Ukraine.*

Abstract

Modular exponentiation is an important operation in many applications that requires a large number of operations. Efficient computations of the modular exponentiation are extremely necessary for efficient computations for provide high crypto capability of information data and in many other applications. Modular exponentiation is implemented using of the development of the right-to-left binary exponentiation method for a fixed base with pre-computation of redused set of residuals. To efficient compute the modular exponentiation over big numbers, the property of a periodicity for the sequence of residuals of a fixed base with exponents equal to an integer power of two is used. The multithreading software implementation of modular exponentiation is described. The MPIR library with an integer data type with the number of binary digits from 256 to 2048 bits is used to develop an algorithm for computing the modular exponentiation.

Comparison of the runtimes of four variants of functions for computing the modular exponentiation is performed. In the algorithm with pre-computation of residues for fixed-base provide faster computation of modular exponentiation compared to the functions of modular exponentiation of the MPIR, OpenSSL and Crypto++ libraries.

Keywords 1

Modular exponentiation, parallel computation, multithreading, big numbers.

1. Introduction

Computing the modular exponentiation for big numbers is widely used to find the discrete logarithm, in number-theoretic transforms, and in cryptographic algorithms. New methods, algorithms and means of their implementation are being researched for efficient computation of the modular exponentiation. There are three directions of modular exponentiation methods: general modular exponentiation, and computation of a modular exponentiation with a fixed exponent or with a fixed base [1]. Special functions have been developed to perform modular exponentiation in mathematical and cryptographic software libraries (Crypto++, OpenSSL, Pari/GP).

Modular exponentiation and discrete logarithm for big numbers require significant computing resources for their implementation. The discrete logarithm is considered a unidirectional function (1), because it is quite difficult to calculate it in a conventionally acceptable time, for example, for breaking a cryptographic code.

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: ihor.o.protsko@lpnu.ua (I. Prots'ko); ocr@ukr.net (O.Gryshchuk); volodymyr.v.riznyk@lpnu.ua (V. Riznyk).
ORCID: 0000-0002-3514-9265 (I. Prots'ko); 0000-0001-8744-4242 (O.Gryshchuk); 0000-0002-3880-4595 (V. Riznyk).

The discrete logarithm problem [2] is formulated as follows: for known integers A, N, y, find an integer x such that

$$x = log_A y, \ (0 \le x \le N - 1), \quad (1)$$

where (A, N)=1; A, N, y, $x \in Z$).

The number $x > 0$ is called the discrete logarithm of the number y with base A and module N according to formula (1).

The solution of the discrete logarithm problem can be the solution of the equation

$$A^x \bmod N = y. \quad (2)$$

That is, having determined the number x, which is the solution of equation (2), we will find the discrete logarithm. Thus, the problem of the discrete logarithm is reduced to the calculation of the modular exponentiation in the form (2).

The paper considers the basic iterative algorithm for computing the modular exponentiation (2) using pre-computation to form a shortened sequence of residues of the fixed base A. The software implementation of multi-threaded computation of the modular exponentiation for big numbers is described. In the discussion section of the results, a comparison of the average executing time of computing the modular exponentiation with the primitives of cryptographic libraries (OpenSSL, Crypto++) is made. The conclusions summarize the possibility of applying the computation of the modular exponentiation (2) using the pre-computation of the remainders of the fixed base A.

2. Related works

Modular exponentiation, A^x mod N, is a basic arithmetic operation in most public-key cryptosystems. Many effective methods of modular exponentiation have been developed and are often used to reduce the execution time of the modular exponentiation operation, which are reviewed in works [3-6]. In paper [3] is described well-known exponentiation algorithms with their complexity analysis and general exponentiation algorithm of zero-one sequences.

There are three types of exponentiation algorithms A^x mod N [1], which include:

1) basic techniques for exponentiation;
2) fixed-exponent x exponentiation algorithms;
3) fixed-base A exponentiation algorithms.

Among the main modular exponentiation algorithms, the following effective solutions are distinguished:

- right-to-left k-ary exponentiation,
- right-to-left k-ary exponentiation,
- left-to-right k-ary exponentiation,
- exponent with a sliding window (sliding window exponentiation),
- Montgomery ladder,
- simultaneous multiple exponentiation and their modifications.

In many works, the method of binary exponentiation with a right-to-left shift is used. Cohen's book [7] presents a more complete discussion of the binary right-to-left and left-to-right methods together with their generalizations to the k-binary method.

A fixed element of a group (generally z/q_z) is repeatedly raised to many different exponent in several cryptographic systems. A popular application of fixed-base exponentiation is in elliptic curve cryptography, for instance for Diffie-Hellman key agreement and elliptic curve digital signature algorithm verification. Therefore, many research works have been focused on a fixed base of modular exponentiation [1, 8].

Considerable attention is paid to the hardware implementation aimed at efficient computation of the modular exponentiation. One of the ways to speed up the computation of modular exponentiation is the parallelization of calculations using modern technologies in universal computer systems [9, 10] or the creation of specialized computing tools [11].

The modern software libraries are used to implement the computation of modular exponentiation. The software implementation of the modular exponentiation computation is included in the software libraries Crypto++ and OpenSSL, PARI/GP, MPIR designed for working with big numbers [12-15].

For example, the Pari/GP software library [12] contains a large set of programs for efficient computations of mathematical functions. The Pari/GP library also includes computation of the modular exponentiation function for big numbers and other special numbers. A highly optimized modification of the well-known GMP or GNU Multiple Precision Arithmetic Library the MPIR library [13] contains the function of the realization the computation of modular exponentiation. The library of cryptographic algorithms and schemes Crypto++ is implemented in C++ and fully supports 32 and 64-bit architectures of many operating systems and platforms [14]. The library contains a set of available primitives for theoretical and numerical operations, such as generation and verification of prime numbers, arithmetic over a finite field, operations on polynomials.

The importance of speeding up the software calculation of modular exponentiation leads to new algorithmic solutions and their implementations [16].

3. The technique of the computation of modular exponentiation with pre-computation of residues for fixed-base

The parallelization of computation of modular exponentiation, based on the use of the exponent x as a binary number ($x_{(k-1)}\ x_{(k-2)} \ldots\ x_2\ x_1\ x_0$), uses the application of the fundamental property of modularity was described in paper [9]. Accordingly, the computation of the modular exponentiation (1) takes the form

$$\begin{aligned} y &= A^{x_{(k-1)}x_{(k-2)}\cdots x_2 x_1 x_0} \bmod N = \\ &= (A^{x_{(k-1)}2^{k-1}} \bmod N * A^{x_{(k-2)}2^{k-2}} \bmod N * \ldots \\ &* A^{x_2 2^2} \bmod N * A^{x_1 2^1} \bmod N * A^{x_0 2^0} \bmod N) \bmod N; \end{aligned} \tag{3}$$

In accordance with formula (3), a scheme for computing of the modular exponentiation [17] is shown on Figure 1.

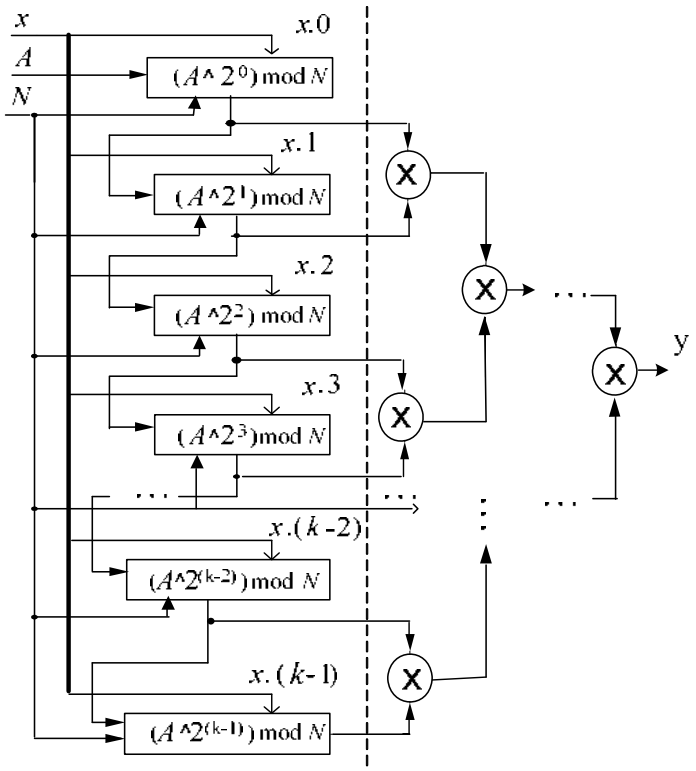


Figure 1: The scheme for computation of modular exponentiation A^x mod N.

The scheme for computing the modular exponentiation consists of the denotations:

- A is the input of the base number; N is the input of the module;
- $(A\text{^}2^i)$ mod N are blocks of computation of the integer exponent of exponent 2^i of the number A by the module, $i = 0,1,2,\ldots, (k\text{-}1)$;
- x is the input of an exponent with binary digits $x.(k\text{-}1), x.(k\text{-}2),\ldots,x.2$, x.1, $x.0$;
- (X) mod N is the block of multiplier under modulo N;
- y is the output of the modular exponentiation.

Consistently determined residuals of the number a raised to the power of 2^i under modulo N in blocks $(A\text{^}2^i)$ mod N, for i=0, 1,…, k-1 are multiplied. In the case of presence of the exponent of the number of binary the value of zero in i-th binary digit $(x.i)$ block $(A\text{^}2^i)$ mod N is not execute the operation, otherwise is computed the multiplication of exponent 2^i of the number A under modulo N, which corresponds to the denotation in Figure 1. The determined product, in the final stage, are formed in blocks (X) mod N the value y of the result of the computation of the modular exponentiation.

For the organization of modular exponentiation execution, in accordance with Figure 1, it is possible to consider the possibility of creating two threads. The thread I computes the numbers a raised to the power of 2^i under modulo N. The thread II computes the product of the results $(A\text{^}2^i)$ mod N, starting with the readiness of the first two values $(A\text{^}2^i)$mod N. Theoretical it is possible to reduce the computation time to 50% of the modular exponentiation in comparison with single thread computation. However, serial or parallel computations $(A\text{^}2^i)$ mod N for big data create a significant delay in the thread I execution [9].

Consider the basic iterative algorithm for computing the modular exponentiation (3) using pre-computation to form a shortened sequence of residues of the fixed base A. Compute, respectively (3), the value modulo N for a simple fixed-base A with exponents $x = 2^i = 1, 2, 4, 8, 16\ldots$, ($i$ =0,1,2,…, r-1).

Let A and N be relatively prime positive integers $(A, N) = 1$ and denote the least positive integer $x = \exp_N A$. Accordance, if A and N relatively prime $(A, N) = 1$, positive integer $\boldsymbol{x}$ is solution of the congruence, if and only if

$$x = q \cdot \exp_N A, \tag{4}$$

Accordance the Euler's theorem, if A and N relatively prime $(A, N) = 1$, that $A^{\varphi(N)} \equiv 1 \pmod N$. Consequently, we can do conclusion

$$\varphi(N) = q \cdot \exp_N A, \tag{5}$$

In case q=1, then $\varphi(N) = \exp_N R$, where R is the positive integer is called a primitive root modulo N. However the positive integer of modulo N, possesses a primitive root R if only if N=2, 4, P^k or $2P^k$, where k is positive integer. The primitive root for modulo $N = P_1^{k1} P_2^{k2} \ldots P_m^{km}$ does not have, except if $\varphi(P_1^{k1})$, $\varphi(P_2^{k2})$,…, $\varphi(P_m^{m1})$ are relatively prime.

Thus, calculating $(R)^i$ mod N (i = 0,1, 2 ,,… N-1), we form a sequence of residuals ($r_0, r_1, r_2, \ldots, r_{i,\ldots}, r_{N-1}$), which periodically repeated for $x > (N\text{-}1)$ exponents. For all values of $A \in Zp$, the sequence A^i mod P is cyclic for a non-primitive element.

For example, for a primitive element R = 7, the sequence of residual values $r_i = (7^i)$ mod 11,

$$(r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9) = (1, 7, 5, 2, 3, 10, 4, 6, 9, 8).$$

The maximum period of repetitions is equal to $\exp_{11} 7 = 10$, because 7^{10} mod 11=1, i =0, 1, 2, …, 9. The unique integer x with $1 \le x \le \varphi(N)$ and R^x mod $N \equiv A$ is called index (or discrete logarithm) $\text{ind}_R A$ of A to base R modulo N. Then the sequence of the indices is equal

$$(0, 1, 2, 3, 4, 5, 6, 7, 8, 9) = (\text{ind}_7 1, \text{ind}_7 7, \text{ind}_7 5, \text{ind}_7 2, \text{ind}_7 3, \text{ind}_7 10, \text{ind}_7 4, \text{ind}_7 6, \text{ind}_7 9, \text{ind}_7 8).$$

Accordingly of the property of indeces

$\text{ind}_7 1 = 0$,

$\text{ind}_7 6 = \text{ind}_7 (2*3) = \text{ind}_7 (2) + \text{ind}_7 3 \bmod 10 = 3+4=7$;

$\text{ind}_7 9 = \text{ind}_7 (3^2) = 2 * \text{ind}_7 3 \bmod 10 = 2*4=8$.

In the case of calculating 7^x mod 11 with index x = 32, the index will be equal to (32 mod ($\text{ind}_{11}7$)) = 2, and accordingly $\text{ind}_7 5$. In the case of determining $(7^{2\,\wedge\,6})$ mod 11, we find the number of the residue in the sequence with the index $\text{ind}_7 3$, which is equal to (2^6) mod 10 = 4. After all, the value of the modular exponentiation for 2 elements in the sequence of residual values $r_4 = 3 = (7^{2\,\wedge\,6})$ mod 11. For computations according to formula (3), we determine the residuals for exponents 2^i, (i =2, 3, 4,

…). As a result of computations $r_i = (7^{2\,\hat{}\,i}) \bmod 11$, ($i$ =2, 3, 4, …) we obtain the values of the residuals given in Table 1. For another primitive element R =13, the sequence of residual values $r_i = (13^i) \bmod 17$ for exponents 2^i, (i =2, 3, 4, …) are given in Table 1.

Table 1

Periodic repetition of residual values $7 \hat{} 2^i \bmod 11$

$7 \hat{} 2^i$	7^0	$7^{2\hat{}0}$	$7^{2\hat{}1}$	$7^{2\hat{}2}$	$7^{2\hat{}3}$	$7^{2\hat{}4}$	$7^{2\hat{}5}$	$7^{2\hat{}6}$	$7^{2\hat{}7}$	$7^{2\hat{}8}$	$7^{2\hat{}9}$	$7^{2\hat{}10}$	…
T' =4	1	7	**5**	**3**	**9**	**4**	5	3	9	4	**5**	**3**	…

Periodic repetition of residual values $13 \hat{} 2^i \bmod 17$

$13 \hat{} 2^i$	13^0	13^1	13^2	13^4	13^8	13^{16}	13^{32}	13^{64}	13^{128}	13^{256}	13^{512}	…
T' =1	1	13	16	**1**	1	1	1	1	1	1	1	…

That is, in the process of computing $(7\hat{}2^i) \bmod 11$, starting with the exponent $2^1 = 2$, we obtain periodic repetition of the values of the residuals $r_0, r_1, \boldsymbol{r_2}, \boldsymbol{r_4}, \boldsymbol{r_8}, \boldsymbol{r_{16}}, r_2, r_4, r_8, r_{16},\ldots$ with period T'= 4 and offset u = 0, because $i = 2^0 = 1$.

That is, in the process of computing $(13\hat{}2^i) \bmod 17$, starting with the exponent 2^2= 4, we obtain periodic repetition of the values of the residuals $r_0, r_1, r_2, \boldsymbol{r_4}, r_4, r_4, r_4, r_4, r_4, r_4,\ldots$ with period T'= 1 and offset u = 2, because $i = 2^2 = 4$.

The value $A\hat{}2^i \bmod N$ is found by the condition

$$A\hat{}2^i \bmod N \equiv A\hat{}2^{(i-t\cdot T')+u)} \bmod N,\ i > u + T' \ , \qquad (6)$$

where $t=\{i/T'\}$ is integer part from division.

Therefore, for a fixed-base A of the modular exponentiation (3), which is equal to the product of the residuals of the exponent $(A\hat{}2^i) \bmod N$, (i = 2, 3, 4, …), you can speed up the process of computing the modular exponentiation by pre-computing (Figure 2) the sequence of residuals, what repetitions with the period T' after the offset u.

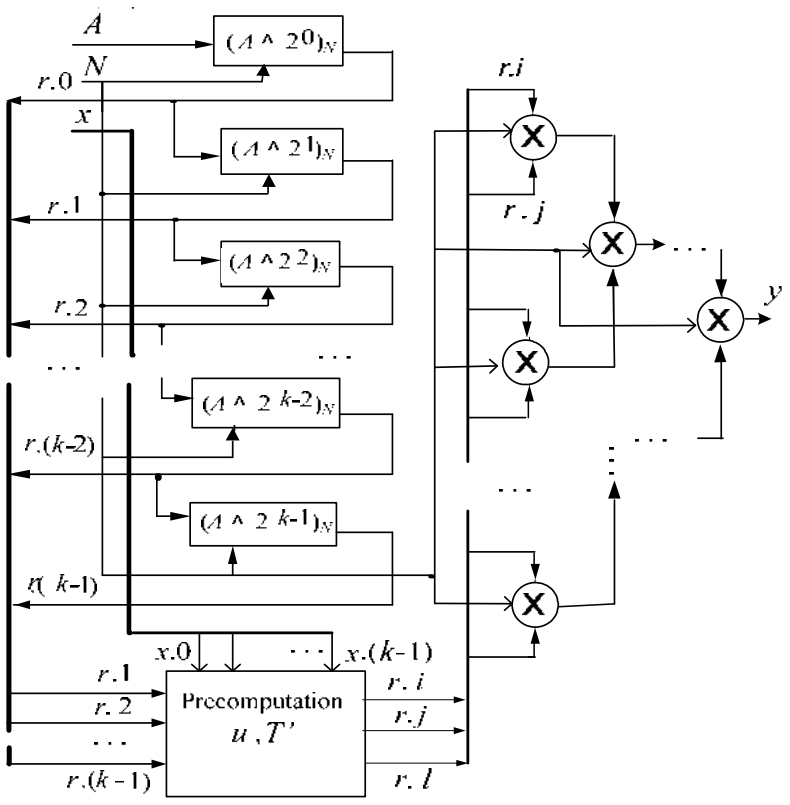


Figure 2: The scheme for computation of modular exponentiation $A^x \bmod N$ with pre-computation.

Thus, on base to the parallel execution of the computation of the modular exponentiation with pre-computation, in accordance with Figure 2, consider the possibility of creating the threads of execution of the product of the residual values under modulo defined in the parallel previous threads residuals of the exponents (A^2^i) mod N, (i = 2, 3, 4, …) under modulo operations. The only difficulty of organizing the computation with such threads is the need to synchronize the performance of threads to ensure the correct computation of the final value y of the modular exponentiation.

4. Software implementation of the computation of modular exponentiation with pre-computation of residues for fixed-base

The algorithm is implemented in C++ language as modular exponentiation function *precompute_parallel*() with aim to compare the performance execution. To implement the algorithm, the library functions mpz_init_set (*mul, base*), mpz_sizeinbase (*exp,* 2), mpz_tstbit (*exp, i*), mpz_mul (*r, r, mul*) from the MPIR library are used, the parameters of which are multi-bit data up to 2048 bits. The software implementation consist the functions:

precompute_parallel (*const RemaindersData&data, const mpz_class&exp, ctpl::thread_pool&pool*);
parallel (*const mpz_class& base, const mpz_class& exp, const mpz_class& mod*);
precompute (*const RemaindersData& data, const mpz_class& exp*);
find_remainders (*const mpz_class& base, const mpz_class& mod, size_t max_exp_bits*);
find_period (*const std::vector<mpz_class>& remainders*);
get_remainder (*const RemaindersData& data, size_t power*);
update_remainders (*RemaindersData& data*);
thread_function(*bool* quit_thread*);
multiplication_thread (*int id, MultiplicationThreadData* data*) and others.

The functions find_period (*const std::vector<mpz_class>& remainders*) computes the products modulo over the pre-computed values of the residuals (A ^2^i) mod N, which are read using the function get_remainder (*const RemaindersData & data, size_t power*). In the cycle of the function mpz_tstbit (*exp, i*) binary bits *x.i* of exponent *exp* are analyzed to determine to perform or not a multiplication operation modulo, accordance the Figure 3 in the pre-computation unit. The computation of the value of the modular exponentiation ends by writing the result in the variable *period_mod_exp_result*.

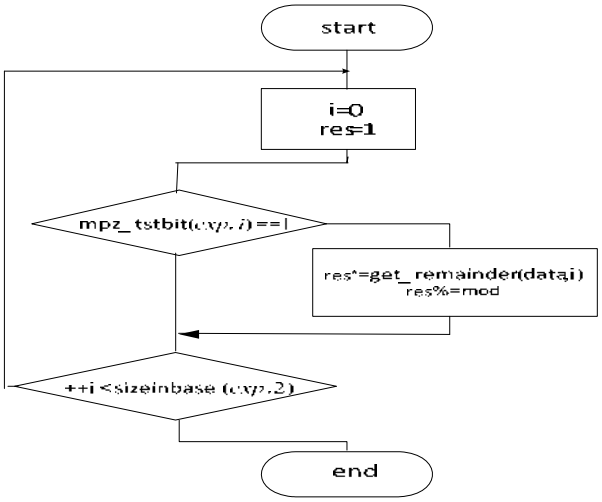


Figure 3: The chart of the algorithm for determining to perform or not a multiplication under modulo in the function *find_period* () to compute the value of the modular exponentiation.

The precompute_parallel() function finds the sequence of residuals for fixed numbers *Base* and *mod* for $Exp = 2^i$ (i = 0, 1, 2, …) and analysis of periodicity. In the program for computing the sequence of residuals is performed by the function find_remainders (*const mpz_class & base, const mpz_class & mod, size_t max_exp_bits*), which contains the bool function find_period (*const std::vector<mpz_class> & remainders*) to set the indication of finding the period. The precomputation have been made in a separate function find_remainders () to optimize multiple residual searches (A ^2^i) mod N. The function update_remainders (*RemaindersData & data*), shortens the length of the sequence of residuals to the end of the first periodicity. This function writes the offset *period_offset* beginning of the period and the length of the period *period_size* in the corresponding fields of the structure

RemaindersData
{ *mpz_class base;*
mpz_class mod;
std::vector <mpz_class> remainders;
size_t period_offset;
size_t period_size;
} also.

The peculiarity of the large values of *Base*, *mod* and *Exp* is also taken into account for which the residuals must be calculated, in case when the value of the period T' is many orders of magnitude greater than the number of bits of the *Exp* exponent.

To organize efficient multithreading computation of modular exponentiation according the precompute_parallel() function, the thread_function(*bool* quit_thread*) and parallel (*const mpz_class& base, const mpz_class& exp, const mpz_class& mod*) are used. Accordingly, these threads are created to perform the computation of the modular exponentiation. The result of the function is written to the variable *s_thread_result*, and the computation time is fixed and averaged to output.

To protect the queue from simultaneous access of the threads is used the *s_mutex* mutex with the basic *lock*() and *unlock*() methods. The *s_mutex* object is passed to our functions, which should use it for interacting. Before accessing the shared data queue *s_thread_queue*, the mutex is locked by the method *lock*(*s_mutex*) and is unlocked after the work with the shared data is completed. The use of a combination a conditional variable and a mutex lock indicates the complexity of the synchronization the performance of threads for computing the modular exponentiation.

5. Results and Discussions

To compare the computation efficiency of the developed modular exponentiation functions *precompute_parallel* () for a fixed base with pre-computation with three functions implemented from the Crypto++ 8.2, MPIR and OpenSSL libraries are used. The developed *precompute_parallel* () function uses multithreads computation of the modular exponentiation.

The Crypto++ library of cryptographic algorithms and schemes is implemented in the C++ language and supports Unix platforms (AIX, OpenBSD, Linux, MacOS, Solaris, etc.), Win32, Win64, Android, iOS, ARM [14]. The library contains a set of available primitives for number-theoretic operations, such as generation and verification of prime numbers, arithmetic over a finite field, operations over polynomials. Each of the primitives of the Crypto++ library includes a set of functions, among which is the function *exp_crypto* () for calculating the modular exponent.

The OpenSSL library (The Open Source toolkit for SSL/TLS) contains a set of tools for cryptography that implements the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols, as well as the corresponding cryptography standards [15]. OpenSSL supports Solaris, HP-UX, Linux, Android, BSD, UnixWare, Win 32 and 64, macOS, iOS, and other

platforms. The library includes three functions to calculate the modular exponent using Montgomery multiplication:

BN_mod_exp_mont () calculates A to the power of p modulo m;

BN_mod_exp_mont_consttime () calculates A to the power of p modulo m. This is a variant of the pre-function, that uses fixed windows and dedicated memory for pre-computation to keep data dependency to a minimum to protect secret exponents;

BN_mod_exp_mont_consttime_x2 () calculates two independent raises of A_1 to the power of p_1 modulo m_1 and A_2 to the power of p_2 modulo m_2. For some fixed and equal modulus values of m_1 and m_2, the function uses optimizations that make it possible to speed up the calculation.

A highly optimized modification of the well-known GMP or GNU Multiple Precision Arithmetic Library the MPIR library [13] contains the function *mpz_powm* () to realize computation of ME. The MPIR library uses an optimized version of the ME algorithm, called the "Sliding-window method" [3], which reduce the average number of multiplication operations.

Numerical experiments were carried out on a computer system with a multi-core microprocessor with shared memory in a 64-bit Windows. Testing was performed on computer systems with processors) an Intel Core i5-10600 (6 cores, 12 threads, 3.30GHz) and an Intel Core i9-10980XE (18 cores, 36 threads, 3.0GHz. According to hyper-threading technology, each physical core consists of two virtual ones.

To compute the modular exponentiation with a given number of trials the values of exponent *Exp*, number *Base* and *mod* were given by pseudo-random numbers with number of binary digit of 1024, 2048 bits. To reduce the total computation time on increasing the number of binary digits of long numbers the number of trials of latch-up of the computation time is decrease correspondent. The result of the execution of the function is recorded in the variable *expected_result*, and the computation time is fixed and averaged with the output of the average value "*average time*" in microseconds.

The results are presented in Table 2, which contains the values of average execution time (μs microseconds) of computing the modular exponentiation for pseudo-random data *Base*1, *Exp*1, *mod*1 for 1024 bits and *Base*2, *Exp*2, *mod*2 for 2048 bits, that are

```
    ================ bits=1024 trials=1000 =================
Base 1 =
155925877528394482614610625993674588019100771066359218078554587162570881239567576804461126115887903799308419845098579180815670096035521874870908961799638269196068560103752308669818128860677719460381304397587862593607968358286567068579479763671817955144283945749615768573725580291910494735428411976050787788916
Exp 1 =
142835209786489997170873255507946573556448214084628524605421188224099687322984673543614176852071053547415698252622573845683075452982474922517841367278609089136988583447756479483918441795733215771335093892746851604252279730368411597397577626119447392355080344631875081748708394736819710836176156337925349995164
mod 1 =
368223265393616765838289047484087924724083351214856164756403136919936573865892058431345272076782139089436981490798503001806033596893273007523252013138190688398066467075600577730915050017234560082947101924548982647158267357750118464079633252964273456146531893267335336011831833021666178050140492122038152864 3

    ================ bits=2048 trials=500 =================
Base 2 =
256773876049791747456501134392418703199486921699875869870642170952808629559929090723006531686316217942866914409024816653331169514458883444161809664073451110710653136235607137432150724953185446158678719720295928259781123638183830596292580376934671270834776657897129937849667886402861740861770565666978446548767497029913351286923714957597816949211708272732020400851990724183722906799368441003878461018521548890319346114385586816108217151247348288474481211605784542061549242679745890886509283127487243351737251588531055149430134861136434044363046876468018170052569298949044683219014189194447340722437694507812846021234 0
```

Exp 2 =
20697118667460294289329131926842862371260858718961622542390394128577965883462065464726476265621500584422101090324880590478894205064526853437187125765172491285762485391331954466581845397077420580593627651323516780475733067117136022933691007420276684081952396408441155446026400666835986702419934866824441890364774228140899158959010059757494492005981153055872187442495482411745134646120584804555929554183515697922429188623722060569894871268972465008089744410453542328276620895938777042971769880327762033155857980482303238918889333926985203629914823135222855353547016859173882001780159272297308256145856605458847223736800

mod 2 =
39379307836430088989989218592031490206361014341421542534554062785454215088157661285489227897171951382718524189693484043298053212367558745592744633067337966506831678671186228119376692608512196753389566952551248777411164835476367047487958360269823032678587498856633964013647348817387555870617118234271663489631617273558370935913977379846081927827707583129656866254202071251222632965868248463435437267182493244722131981575330008476925540749022614215472137894322948082016116493861482786577579582219290166703044662301180570363569358774151591894294024434883586470904927529897214681297523329711184436660057952144059718952 67

Testing for the average execution time of computation of modular exponentiation (Table 2) was performed by the functions: *crypto*++() from the Crypto++ library, *mpz_powm* () from the MPIR library, *BN_mod_exp_mont* () from the OpenSSL library. The comparison is performed with developed functions *precompute_parallel* ().The pre-computation time *precompute* () to determine of the sequence of redused set of residues is taken into account.

Table 2.
The average execution time (µs) of the functions of computing the modular exponentiation

Release/x86	Intel Core i5-10600 number of threads (2..64) >12		Intel Core i9-10980XE number of threads (2..64) > 36	
Data	Base1, Exp1, mod1	Base2, Exp2, mod2	Base1, Exp1, mod1	Base2, Exp2, mod2
bits / trials	1024 / 1000	2048 / 500	1024 / 1000	2048 / 500
crypto++()	442	3309	500	3713
BN_mod_exp_mont ()	312	2228	340	2497
mpz_powm ()	279	1983	311	2321
Precompute ()	236	1242	260	1399
Parallel ()	61	244	106	254
precompute_parallel ()	297	1486	366	1653

The developed function *precompute_parallel*() performs the computation of modular exponentiation by forming (precompute average time) a reduced sequence of residuals (Figure 4).

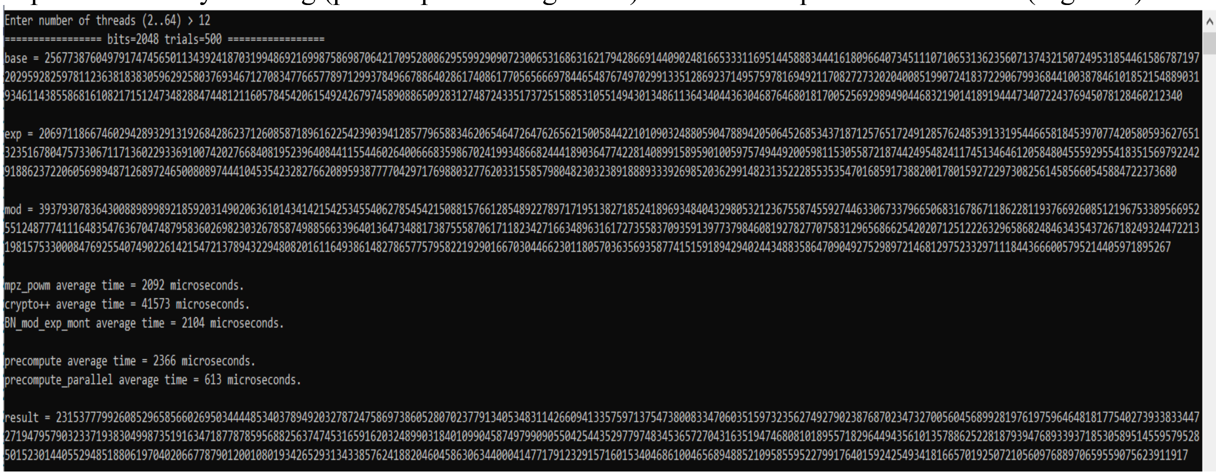


Figure 4: The result of testing the functions of calculating the modular exponent on a computer system with an Intel Core i5-11400H processor with a number of threads of 12

The *precompute_parallel*() function of the modular exponentiation reduces the computation time relative to other functions of software libraries Crypto++, OpenSSL and MPIR designed for working with big numbers. The execution time of modular exponentiation on computer systems with Intel Core i5-10600 and Intel Core i9-10980XE processors does not change significantly according to Table 2. The optimal number of threads is 12...16 for fast computation of modular exponentiation for universal computer systems.

In the process of solving many applied problems of efficient computation of modular exponentiation, one of the most important parameters is the calculation time. An important component of this parameter is, along with the use of algorithmic and software tools, the performance of computing tools. Therefore, based on of the developed software the further implementation of the computation of modular exponentiation using multithreaded technologies will provide an opportunity the efficient computation of modular exponentiation with a fixed base [19].

6. Conclusion

The work compares and analyses the developed software implementation *precompute_parallel*() function of the computation of modular exponentiation and the software implementation of the functions of Crypto++, OpenSSL and MPIR libraries. The computational scheme of modular exponentiation, the software implementation of the algorithm for computing of modular exponentiation, the run time results of the computation on multi-core microprocessors of universal computer systems have been described. As a result, has been developed the computation function *precompute_parallel*() speedups the execution of the computations using modular exponentiation of the right-to-left binary exponentiation method for a big number of fixed base with pre-computation of redused set of residuals.

The scientific novelty of obtained results lies in the implementation of parallelism using multithreading in the algorithm of computing the modular exponentiation based on the use of the representation of exponent in the form of a binary number and pre-computation for a fixed base of a reduced set of residuals.

The practical significance of the work lies in the fact that the obtained results can be successfully apply in the modern asymmetric cryptography, for efficient computation of number-theoretic transforms and other computational problems. The developed function *precompute_parallel*() speedups the execution of the computations of modular exponentiation of big numbers in comparison with existing libraries. The especially effective will be use function *precompute_parallel*() for application with fixed-base exponentiation is in elliptic curve cryptography, for instance for Diffie-Hellman key agreement and elliptic curve digital signature algorithm verification.

Prospects for further research are the development and implementation Montgomery Modular Multipliers in developed function *precompute_parallel*() of multithreading computations of modular exponentiation for big numbers.

7. References

[1] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, Handbook of applied cryptography. 5th printing, CRC Press, Boca Raton, 2001. doi:10.1201/9780429466335.

[2] C. Studholme, The Discrete Log Problem, 2002. URL: http://www.cs.toronto.edu/~cvs/dlog/research_paper.pdf

[3] A. Jakubski, R. Perlinski, "Review of general exponentiation algorithms", Scientific Research of the Institute of Mathematics and Computer Science, 10.2 (2011): 87-98.

[4] A. Rezai, P. Keshavarzi, "Algorithm design and theoretical analysis of a novel CMM modular exponentiation algorithm for large integers". RAIRO – Theoretical Informatics and Applications, 49.3, (2015): 255-268. doi:10.1051/ita/2015007.

[5] I. Marouf, M. M. Asad, Q. A. Al-Haija, "Comparative Study of Efficient Modular Exponentiation Algorithms". COMPUSOFT, International journal of advanced computer technology, 6.8 (2017): 2381-2392.
[6] S. Vollala, K. Geetha, N. Ramasubramanian, "Efficient modular exponential algorithms compatible with hardware implementation of public-key cryptography". Security and Communication Networks, 9.16 (2016): 3105-3115.
[7] H. Cohen, *A* course in computational algebraic number theory. Berlin, Heidelberg: Springer, 1993. doi:10.1007/978-3-662-02945-9.
[8] J.-M. Robert, C. Negre, T. Plantard, "Efficient Fixed Base Exponentiation and Scalar Multiplication based on a Multiplicative Splitting Exponent Recoding", Journal of Cryptographic Engineering, Springer, 9.2 (2019): 115-136. doi:10.1007/s13389-018-0196-7.
[9] P. Lara, F. Borges, R. Portugal, N. Nedjah, "Parallel modular exponentiation using load balancing without precomputation". Journal of Computer and System Sciences, 78.2 (2012): 575-582. doi:10.1016/j.jcss.2011.07.002.
[10] N. Emmart, F. Zheng, C. Weems, C. Faster Modular Exponentiation using Double Precision Floating Point Arithmetic on the GPU, in: Proceedings of the 25th IEEE Symposium on Computer Arithmetic (ARITH), Amherst, MA, USA, 2018, pp. 126-133. doi:10.1109/ARITH.2018.8464792.
[11] S. Li, J. Tian; H. Zhu; Z. Tian; H. Qiao; X. Li; J. Liu, Research in Fast Modular Exponentiation Algorithm Based on FPGA, in: Proceedings of the 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Qiqihar, China, 2019, pp. 79-82.
[12] PARI/GP home, 2021. URL: http://pari.math.u-bordeaux.fr/.
[13] MPIR: Multiple Precision Integers and Rationals. 2021.URL: http://mpir.org/.
[14] Crypto++ Library 8.6, 2021. URL: https://www.cryptopp.com
[15] OpenSSL. Cryptography and SSL/TLS Toolkit, 2021. URL: http://www.openssl.org/.
[16] C. Negre, T. Plantard, "Efficient Regular Modular Exponentiation Using Multiplicative Half-Size Splitting", Journal of Cryptographic Engineering, Springer, 7.3 (2017): 245-253. doi:10.1007/s13389-016-0134-5.
[17] I. Prots'ko, N. Kryvinska, O. Gryshchuk, "The Runtime Analysis of Computation of Modular Exponentiation", Radio Electronics, Computer Science, Control, 3 (2021): 42-47. doi:10.15588/1607-3274-2021-3-4.
[18] I. Prots'ko, O. Gryshchuk, "The Modular Exponentiation with precomputation of redused set of resedues for fixed-base". Radio Electronics, Computer Science, Control, 1 (2022): 58-65. doi:10.15588/1607-3274-2022-1-7 .
[19] Exponentiation by squaring, 2022. URL: https://en.wikipedia.org/wiki/Exponentiation_by _squaring.

Approach to Search Result Rankings Based on User Ratings

Viacheslav Zosimov [a], Oleksandra Bulgakova [a]

[a] *Taras Shevchenko National University of Kyiv, Bohdan Hawrylyshyn str. 24, Kyiv, 04116, Ukraine*

Abstract

This paper proposed an approach to rank search results based on user ratings, with a subjective approach to the ranking process. Expert groups unique to each user are created to achieve this effect. When no common ratings are available, the expert group is formed based on a model built on the user's social profile, utilizing inductive algorithms such as a polynomial neural network with active neurons.

The approach of ranking search results provides a unique order of elements for each user's list of web resources. This effect is achieved by using evaluations from an expert group that is unique to each user, as well as by incorporating each evaluation into the model for calculating final rankings of web resources with its unique weight, calculated based on their previous activity in the system. The resulting ranking model is much more difficult to falsify because it is based on subjective factors. The ranking model will be unique for each user. Falsifying it would require reproducing the preferences of each user, which is much more difficult than acquiring links to one's web resource from authoritative sources.

Keywords 1

Ranking, search results, user ratings, expert groups, social profile, inductive algorithms, polynomial neural network, active neurons

1. Introduction

Currently, online advertising is the most effective way of promoting a business. This has resulted in search engines no longer being a tool for obtaining information, but rather becoming advertising platforms that display to the user not the pages that are most relevant to their informational needs, but rather those for which more money has been invested in their promotion. Search engines benefit from artificially lowering the quality of organic search results, as contextual advertising, which is often irrelevant to the search query, appears more appropriate against that backdrop.

The ranking algorithms of search engines take into account a large number of factors, but the main weight is given to the page ranking, which is calculated based on the analysis of the quantity and quality of external links to the page. Such evaluation methods are objective, but they can be easily falsified with a certain advertising budget by purchasing the necessary amount of high-quality links from external sources. This implies that they are oriented towards satisfying the needs of advertisers rather than users.

In [1], described learning-based ranking model is proposed to improve recommendation systems with implicit user feedback. In [2], a ranking model is described that uses adaptive learning to improve content-based recommendation systems. In [3] develops a hybrid ranking model for scientific articles that combines content-based and citation-based methods. The ranking model using neural networks and weak supervision is described in [4-5]. It can work with incomplete data, making it more universal and convenient to use. However, the drawback is the complexity of understanding the principles of neural network operation. [6] is dedicated to using BERT (Bidirectional Encoder Representations from Transformers) for ranking in search engines. Research results have shown that

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: zosimovvv@gmail.com (A. 1); sashabulgakova2@gmail.com (A. 2)
ORCID: 0000-0003-0824-4168 (A. 1); 0000-0002-6587-8573 (A. 2);

BERT provides high accuracy compared to traditional ranking methods. However, its implementation requires significant computational resources. [7] proposed a new approach to ranking that uses reinforcement learning to aggregate different page ratings. Results have shown improvement in ranking accuracy. However, the drawback is the need for significant computational resources and implementation complexity.

In [8] authors discuss sentiment analysis methods, including rule-based and machine learning-based methods, and provide examples of their applications in various domains, such as social media monitoring and product review analysis. In conclusion, the authors highlight the importance of integrating recommender systems and sentiment analysis for creating more effective and personalized recommendation systems.

The development describes in [9] is designed to improve the relevance of search within organizations and enterprises by automatically identifying and capturing the knowledge and expertise of their employees. This can provide more accurate and relevant search results by linking search queries to employees who possess the necessary expertise.

However, one potential drawback of this system is its potential inefficiency in using contextual information. It's important to consider not only the keywords but also the context in which they are used and other related factors when conducting searches. If the system doesn't account for context and can't adapt to changing user requirements, it can lead to irrelevant or incomplete results.

Brytsov R.A. researches this problem. In his work [10], he presented a theoretical ranking model based on web resource visit statistics and document viewing time. However, it does not take into account the opinions of users about the web resources, as well as the level of agreement between users and experts. Ranking methods based on user ratings are widely used for ranking products in online stores. However, they are also based solely on the number of ratings and their numerical values.

Overall, recent research has shown that the use of neural networks and reinforcement learning can improve ranking accuracy in search engines. However, these approaches require significant computational resources.

Information retrieval is the process of searching for unstructured documentary information that satisfies the user's information needs [11]. Given that user information needs are individual and subjective, this definition suggests that search result ranking algorithms should be based on subjective factors specific to each user.

Therefore, the development of methods for personalizing search engine results by providing users with search management tools and the development of new ranking models based on subjective user information needs is a pressing task.

In this paper proposed an approach to ranking search results based on user ratings. The main difference of this method is the subjective approach to the ranking process. This effect is achieved by creating expert groups unique to each user. In the absence of any common ratings, the expert group is formed based on a model built on the user's social profile using inductive algorithms, namely a polynomial neural network with active neurons.

2. Formation of expert groups

In the real task of ranking search results, user ratings are used as input data for which their status as an expert is not defined. Obviously, accepting the opinions of all users who have rated a web resource as expert would be incorrect. It is also evident that the ratings and personal data provided during registration are not sufficient to unambiguously identify a user as an objective expert in the subject area. However, this data is sufficient to determine subjective expert groups for each user based on the criterion of proximity of the user's ratings to those of each expert.

In this work, for the search query of the current user, the ranks of web resources from the search output are calculated as the weighted harmonic mean of all ratings from the members of the expert group. Expert groups for each network user are unique and are formed in the background mode of the system using three methods, depending on the presence of common ratings on a certain set of web resources between the current user and a potential expert. For convenience of presentation, experts are divided into three levels according to the method of calculating their weight. The division into levels

is only of a logical nature. The weight of experts from all levels is equivalent and is taken into account when ranking search results without additional coefficients.

To establish clarity in the definitions, the following terms are proposed within the framework of the research:

Current user U_0 - the user for whom the group of experts is formed and for whom the ranking of search results is carried out.

Expert weight - a measure of the agreement between the expert's opinion and the current user's opinion, calculated based on the similarity of their assessments for a certain set of web resources.

First-level potential expert U_i - a user who shares ratings with the current user, but whose level of agreement has not yet been calculated.

Second-level potential expert $\hat{U}_j$ - a user who does not share ratings with the current user, but shares ratings with a first-level expert.

Third-level potential expert $\widetilde{U}_k$ - a user who does not share ratings with the current user or first-level experts.

2.1. Calculating the weight of first-level experts

Direct calculation can be applied in case the current user shares ratings with a set of potential experts X for some set of web resources. It allows calculating the similarity of ratings for each pair of "current user - potential expert" $d(U_0, U_i)$ separately. User ratings have values from 1 to 10, where 10 represents the most acceptable option. The expert's weight value is determined as the arithmetic mean of the absolute differences between each pair of the user's and potential expert's ratings:

$$d(U_0, U_i) = \sum_{j=1}^{m} \frac{\left|U_{0j} - U_{ij}\right|}{m} \quad (1)$$

where m is the number of common ratings that the current user has with the i-th potential expert from the set X, and j is the ordinal number of the rating. The quality of the connection is evaluated using the Shewhart chart. Candidates with high and very high connection strength are selected as experts. The weight of expert U_i relative to the current user U_0 is expedient to consider as a metric $d(U_0, U_i)$ in the metric space (X, d), where $X[U_0, \ldots, U_n]$ is the set of all system users. The function d satisfies the axioms of identity, symmetry, and triangle, but it cannot be defined for every pair of elements in the set X, since not all users have common ratings. Therefore, in the context of this problem, it is incorrect to use the term metric space, so in the future, we will refer to the function $d(U_0, U_i)$ for determining the weight of the expert as an analogue of a metric. Expert weight values are numbers within the interval $d(U_0, U_i) \in [0,\ldots,0.9]$, therefore, to determine a qualitative assessment, the result of a normalizing function is used, rather than the value itself:

$$W(d) = 1 - \left(1{,}1 \cdot \frac{d(U_0, U_i)}{10}\right) \quad (2)$$

2.2. Calculating the weight of second-level experts

Forming expert groups only from those users who have common ratings with the current user significantly narrows down the circle of potential experts. To solve this problem, a method for calculating the weight of experts in the absence of common ratings with the current user has been proposed. It assumes the presence of common ratings with first-level experts and determines the overall weight of a potential second-level expert with respect to the current user $d(U_0, \hat{U}_j)$ taking into account their weight relative to the first-level expert $d(U_i, \hat{U}_j)$ and the weight of the first-level expert relative to the current user $d(U_i, \hat{U}_j)$.

The weight of second-level experts relative to the current user is expedient to calculate as the product of the weight of the first-level expert relative to the second-level potential expert who has common ratings and the weight of the first-level expert relative to the current user:

$$W(d(U_0,\hat{U}_j)) = W(d(U_i,\hat{U}_j)) \cdot W(d(U_0,U_i)) \quad (3)$$

The expert group includes second-level potential experts whose weight relative to the current user is $W(d(U_0,\hat{U}_j)) > 0.7$.

2.3. Calculating the weight of third-level experts

The proposed method can be applied when there are a low number or absence of shared ratings, based on a model for calculating the weight of potential experts constructed from personal social profiles. The calculation of the weight of potential experts based on their past activity always yields the most accurate results. However, during the implementation and early stages of system operation, there will inevitably be situations where there is insufficient data to apply this method. Accumulating a certain base of ratings, i.e., the presence of shared ratings in the database in sufficient quantities to form expert groups for most system users, is necessary for its application. Therefore, to ensure correct system operation at an early stage of implementation, a method for determining the weight of users who have no shared ratings was developed. The user's social profile is formed based on the information they provide during registration in the system [12]. To build a model for calculating the weight of experts, a number of subjective features x_m were selected, which may directly or indirectly affect the visitor's evaluation.

For modeling the degree of consensus among experts, a generalized iterative algorithm (GIA) of inductive modeling was chosen - a polynomial network with active neurons [13].

Formally, in the general case, the GIAI for a layer r is defined:

1. The input matrix is $X_{r+1} = (y_1^r,...,y_F^r,x_1,\ldots,x_m)$.
2. Transition operators of the form $y_l^{r+1} = f(y_i^r, y_j^r)$, $l = 1,2,\ldots,C_F^2$, $i,j = \overline{1,F}$ and $y_l^{r+1} = f(y_i^r, x_j), l = 1,2,\ldots,Fm, i = \overline{1,F}, j = \overline{1,m}$ with a quadratic partial description are applied.
3. For each description, the optimal structure $f(u,v) = a_0 d_1 + a_1 d_2 u + a_2 d_3 v + a_3 d_4 uv + a_4 d_5 u^2 + a_5 d_6 v^2$, $d_{opt} = \arg\min_{l=1,q} CR_l$, $q = 2^p - 1$, $d_k = \{0, 1\}$, $f_{opt}(u,v) = f(u,v,d_{opt})$ is found;
4. The algorithm stops when $CR_{\min}^{r+1} \geq CR_{\min}^r$ and the optimal model corresponds to the value $CR_{\min}^r$ for the r-th layer. Usually, the criterion of regularity is applied:

$$AR_{B|A} = \left\| y_B - \hat{y}_B \right\|^2 = \left\| y_B - X_B \hat{\theta}_A \right\|^2, \quad (4)$$

based on dividing the data sample into two non-overlapping sub-samples - training set A and validation set B, $W^T = (A^T B^T)$ where $\hat{y}_B$ is the estimate of the output on B by a model, $\hat{\theta}_A$ vector of parameters of which is calculated on A.

Figure 1 represents a system that takes an input *X* and uses it to generate different models on next layer *r*. The generated models are passed to an active neuron block that optimizes their complexity using combinatorial optimization and selects *F* best models. The system then computes the error for each model, selects *F* best models again, and repeats the process until only one model is left. The final model is selected based on having the lowest error. Finally, the system outputs the value of *y*.

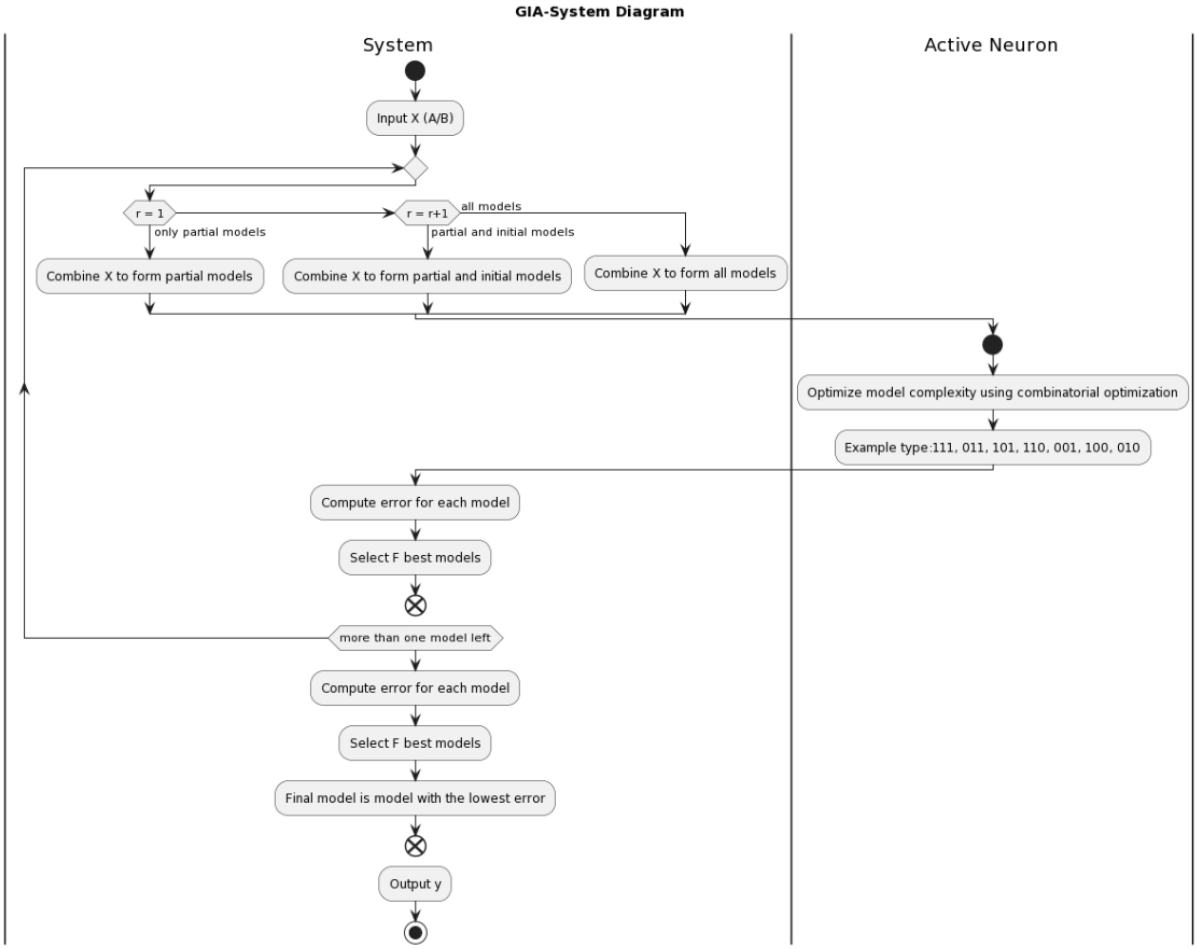


Figure 1: GIA scheme

3. Results

For comparing the results of the coherence of experts' opinions using the methods described above, a data sample was used in which 20 users evaluated 20 web resources based on the quality of presented information and ease of use. The evaluation scale ranged from 1 to 10, where 10 is the best score. Expert #0 is the current user, and a group of experts is selected based on their ratings. For ease of understanding, the web resources will be denoted by capital Latin letters with a digital index equal to the number of the data sample, and the user numbers will be represented by digits.

For each potential expert paired with user #0, a measure of consensus of opinions was calculated (Table 1).

Table 1

Measures of concordance of user #0's opinions with each of the potential experts

User	The weight value of the potential expert | correlation strength >0.7
1	0.9285 | +
2	0.8735 | +
3	0.9010 | +
4	0.9120 | +
5	0.6370 | -
6	0.9120 | +

7	0.5765 | -
8	0.8955 | +
9	0.8955 | +
10	0.9010 | +
11	0.9285 | +
12	0.9175 | +
13	0.6975 | -
14	0.8570 | +
15	0.6315 | -
16	0.5600 | -
17	0.5270 | -
18	0.7580 | +
19	0.7195 | +
20	0.7525 | +

To develop a methodology for considering the weight of second-level experts in ranking, it is necessary to choose criteria for assessing the ratio of the weight of second-level experts to the current user, expressed through the weight of first-level experts. Fig. 2-3 shows the modules of the differences in weight of second-level experts and their weights relative to the current user.

The diagram in Fig. 2 shows the dependence of the average difference in weight of second-level experts relative to the weight of first-level experts. The graph shows that the weight deviation significantly increases when the weight values of first-level experts are less than 0.8.

Potential second-level experts whose weight relative to the current user is $W(d(U_0,\hat{U}_j)) > 0.7$ are selected for the expert group.

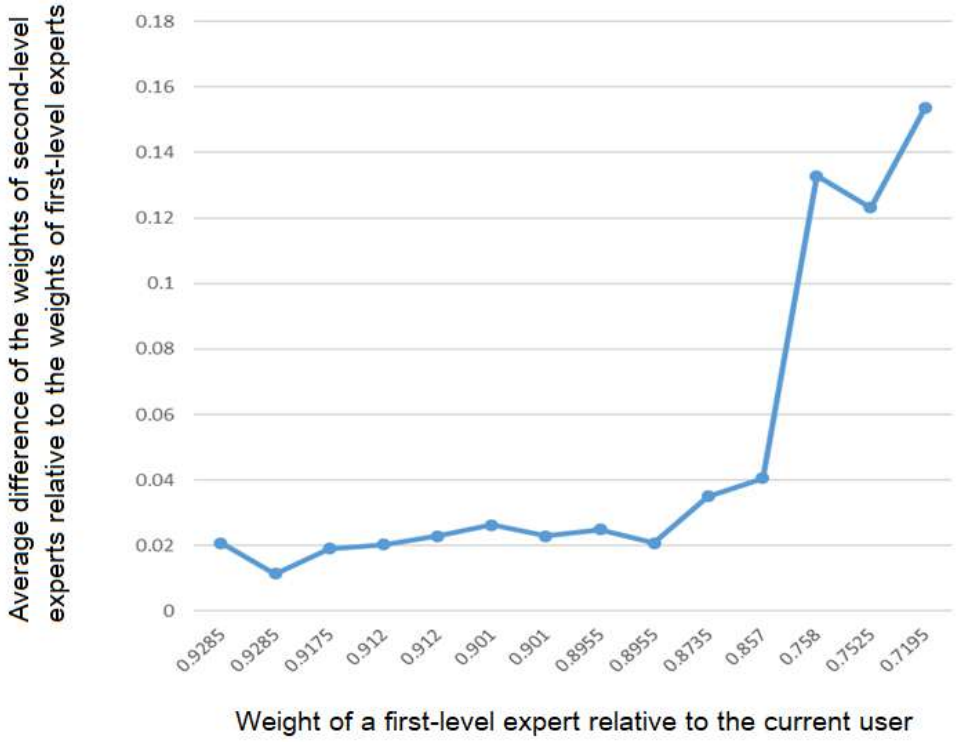


Figure 2: The dependence of the average difference in weight of a second-level user on the weight of a first-level expert

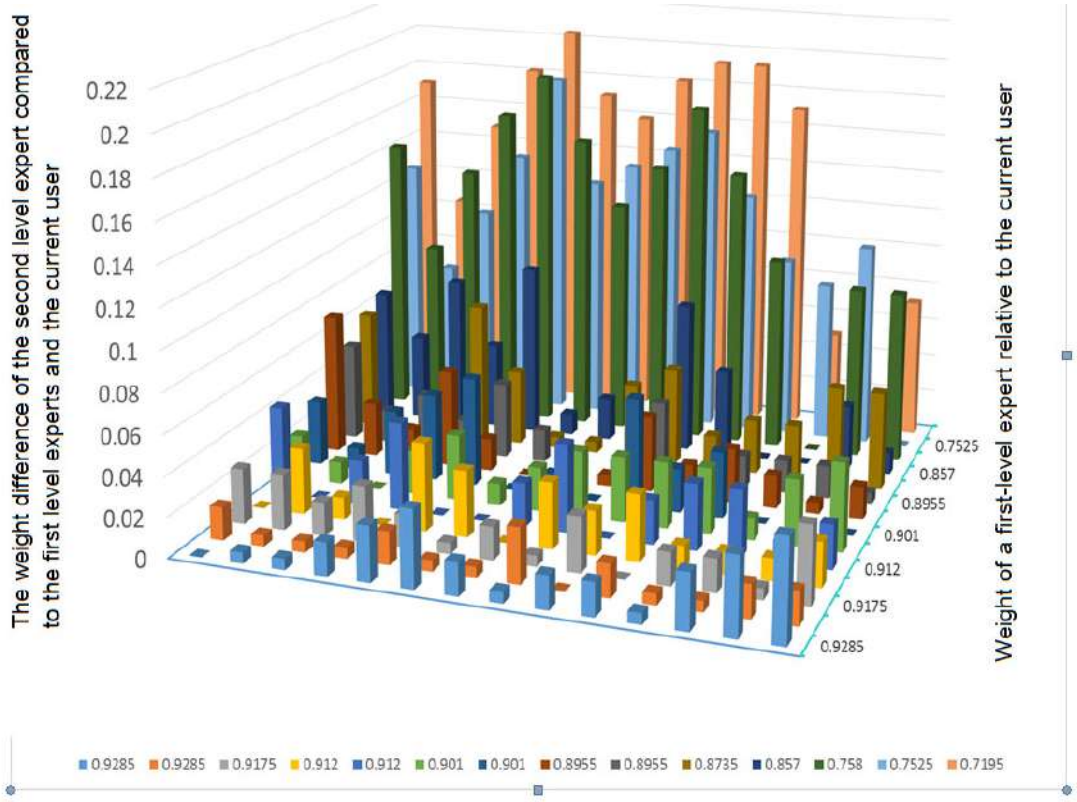


Figure 3: The weight differences in of second level users

The calculation of the weight of potential experts based on the analysis of their previous activities always gives the most accurate results. However, during the system's introduction and the period from the beginning of its operation until a certain database of ratings has accumulated, situations will inevitably arise where there is not enough data for its application. By accumulating a certain database of ratings, it is meant having a sufficient number of shared ratings in the database to form expert groups for most users of the system. Therefore, to ensure the proper functioning of the system at an early stage of its operation, the GIA algorithm is used at the third level to determine the weight of users who have no shared ratings at all. The generalized iterative algorithm is a set of iterative and iterative-combinatorial algorithms defined by the components of a vector of three index sets: DM (Dialogue Mode), IC (Iterative-Combinatorial), MR (Multilayered-Relaxative), where any iterative algorithm is defined as a specific case of the generalized UIA = {DM, IC, MR}. DM can take three values: 1 - standard automatic mode; 2 - planned automatic mode; 3 - interactive mode; IC can take two values: 1 - iterative and 2 - iterative-combinatorial algorithms; MR can take three values: 1 - classical multilayered, 2 - relaxative, 3 - combined algorithms [13].

The input data consisted of social profiles of users (U_0-U_{60}) who participated in previous experiments. Since their weight relative to user U_0 had already been calculated on the rating data samples, the same expert groups were used for building the model.

To determine in terms of users who have no common ratings with both the current user and the members of the expert group $U_{0,exp}$, we will refer to them as potential third-level experts. It is necessary to build a model of the influence of socio-personal factors of Internet users on the degree of agreement of opinions between the potential third-level expert and the current user. The sample was randomly divided into two sub-samples in the following proportions: 70% - testing, 30% - validation. The obtained model is an intermediate stage for solving the given task under time constraints. Table 2 shows the values of actual and modeled results on samples A and B for GIA. In Figure 4 the generalized iterative algorithm reaches a minimum on the 7th layer.

The proposed methodology is applied to the task of ranking search results in the search engine Google.com.ua. At first glance, this task seems very laborious, considering the large number of results, which in most cases reaches hundreds of thousands, or even several million. However, upon

closer analysis of the algorithms used by search engines, it turns out that the computational complexity of the task is several orders of magnitude lower [14]. The actual number of search results that a user has access to be much smaller than what the system claims when performing a search query.

Table 2
The values of actual and modeled results on samples A and B for GIA

Sub-samples	Real data, y	Model data, $\hat{y}$
	1	0,904
	0,9285	0,895
	0,8735	0,752
	0,901	0,882
	0,912	0,810
	0,637	0,628
	0,912	0,865
A	0,5765	0,637
	0,8955	0,818
	0,8955	0,961
	0,901	0,911
	0,9285	0,886
	0,9175	0,928
	0,6975	0,737
	0,857	0,856
	0,6315	0,730
	0,56	0,680
B	0,527	0,574
	0,758	0,788
	0,7195	0,814
	0,7525	0,750

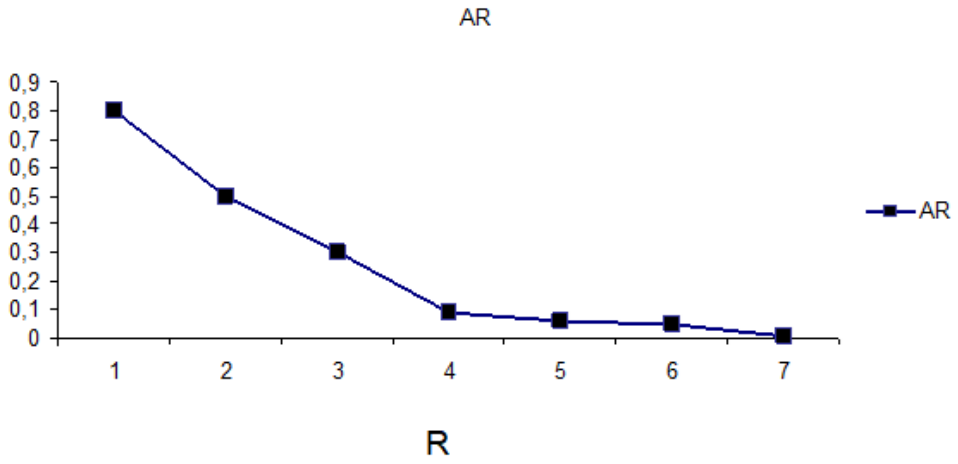


Figure 4: The values of the evaluation criterion (AR criterion) for the layers R

To verify the effectiveness and correctness of the methods presented above for solving the problem of ranking web resources based on user ratings, four experiments were conducted. Four data samples were formed, one for each experiment (Table 1-2). In each sample, 20 users rated 20 web resources based on the quality of the presented information and ease of use. A total of 60 users were involved in forming the four samples. Some users participated in rating web resources in more than one sample.

However, the same users were denoted with the same numbers in all four experiments, for example, user #1 in each experiment is the same person. Each experiment involved several experts from the previous experiment and new users. This was done to be able to trace the weight values of the same experts in different data samples. Before starting the ranking of web resources according to the chosen methods, it is necessary to obtain a reference ranking against which the results will be compared. Simple sorting of web resources in decreasing order of user 0's ratings only provides approximate results. Such a method only orders a group of web resources with the same ratings in decreasing order. The true order of web resources within each group remains unknown. Therefore, to construct a reference ranking, user 0 assigned a rank from 1 to 20 to each web resource, where 1 is the highest rank.

Manual ranking was used to create the reference ranking for the current user. The next step is to construct a ranking based on median values and assign new ranks. The new rank for a web resource with a unique value is equal to its position in the ordered list. In case several web resources have the same ranks based on median values, their new ranks are the arithmetic means of their positions in the ordered list. As some web resources received the same score, they are considered equivalent and are grouped into an equivalence class. To evaluate the efficiency of the aforementioned methods for calculating average scores, the average deviation from the reference ranking is used. It is calculated as the arithmetic mean of the differences between the positions of web resources.

The summarized results of the conducted experiments are presented in Table 3.

Table 3

Measures of concordance of user #0's opinions with each of the potential experts

Experiment number	Weighted harmonic mean
#1 The sum of deviations	18
Average value of deviation:	0,9
#2 The sum of deviations	24
Average value of deviation:	1.2
#3 The sum of deviations	22
Average value of deviation:	1.1
#4 The sum of deviations	20
Average value of deviation:	1
Best results	4

The last row of the Table 3 shows the number of times the determined method provided the best ranking results. If more than one method showed the best results during the experiment, they are all considered the best.

The methodology for building a personalized web ranking model based on user ratings has shown high effectiveness, indicating the potential for developing this direction further.

The developed method for ranking search results generates a unique ordering of web resources for each user. This is achieved by using ratings from expert groups unique to each user, and by assigning each rating a unique weight calculated based on the user's previous activity in the system.

The resulting ranking model is much more difficult to falsify because it is based on subjective factors. Each user will have a unique ranking model, and falsifying it would require reproducing each user's preferences, which is much more difficult than acquiring links to their website from authoritative sources.

4. Conclusions

An approach to ranking search results based on user ratings is proposed. The main difference of this method is the subjective approach to the ranking process. This effect is achieved by forming expert groups unique to each user. Experts are selected based on the degree of agreement with the current user, calculated based on the common ratings for a certain set of web resources. Selection of

users for the expert group is based on their weight relative to the current user, which is a measure of their agreement.

A methodology for forming unique expert groups for each user is proposed, which includes three levels depending on the presence of common ratings for a certain set of web resources between the current user and potential experts. The first level involves selecting potential experts based on their common ratings with the current user for a certain set of web resources. The second level involves selecting potential experts based on their agreement with the current user on a certain set of web resources. The third level involves selecting potential experts based on their weight relative to the current user, which is a measure of their agreement.

The proposed method has several advantages over traditional ranking methods. First, it takes into account the subjective nature of user preferences. Second, it allows for the formation of unique expert groups for each user, which can improve the accuracy of the ranking results. Third, it provides users with tools to manage their search results.

The next step in the development of this approach could be to construct ranking models based on a large number of factors, similar to modern search engines. By incorporating additional factors such as click-through rates, time spent on a page, and other user behavior data, it may be possible to further improve the accuracy of the ranking results.

5. References

[1] R. Yu, Q. Liu, Y. Ye, M. Cheng, E. Chen and J. Ma, "Collaborative List-and-Pairwise Filtering From Implicit Feedback". IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 6, pp. 2667-2680, 1 June (2022), doi: 10.1109/TKDE.2020.3016732.

[2] N. S. Raj and R. V G, "A Rule-Based Approach for Adaptive Content Recommendation in a Personalized Learning Environment: An Experimental Analysis". IEEE Tenth International Conference on Technology for Education (T4E*)*, (2019), pp. 138-141, doi: 10.1109/T4E.2019.00033.

[3] Li Li, Yujie Zhang, Shuai Zhang, and Xinyu Li. A Hybrid Approach for Ranking Research Articles. Information Processing & Management (2017), vol. 53, no. 6, pp. 1407-1419.

[4] Zhaopeng Meng, Jianxun Lian, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng, "Personalized Ranking Framework via Multilayer Perceptron Learning". IEEE Transactions on Knowledge and Data Engineering (2017), vol. 29, no. 7, pp. 1509-1521.

[5] Shuai Shen, Zhiyuan Liu, Maosong Sun, and Yang Liu, "Neural Ranking Models with Weak Supervision." in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (2019), pp. 111-120. doi:10.48550/arXiv.1704.08803

[6] M. Makarov, E. Tutubalina, P. Braslavski, "BERT for Ad-hoc Retrieval: Baselines and Analysis", in Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, (2020), pp. 516-530.

[7] Yongqiang Zhang, Min Zhang, Yiqun Liu, Shaoping Ma, and Jintao Li, "A Reinforcement Learning Approach for Rank Aggregation in Information Retrieval", IEEE Access (2021), vol. 9.

[8] S.Mohammed AL-Ghuribi Shahrul, A. Mohd Noah. A Comprehensive Overview of Recommender System and Sentiment Analysis. URL: https://arxiv.org/ftp/arxiv/papers/2109/2109.08794.pdf

[9] S. Brave, R. Bradshaw, J. Jia, C. Minson. Method and apparatus for identifying, extracting, capturing, and leveraging expertise and knowledge. URL: https://patents.google.com/patent/US7698270

[10] R. Britsov Ranking of information based on user ratings and behavior. T-Comm, Telecommunications and Transport. (2016) Vol. 10. No. 1. P. 62.

[11] Christopher D. Manning. Introduction to Information Retrieval. Cambridge University Press, (2008) 520 p. URL: https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

[12] V. Zosimov, O. Bulgakova, V. Pozdeev. Semantic Profile of Corporate Web Resources. CEUR Workshop Proceedings (2021), 3179, pp. 389–397. URL: https://ceur-ws.org/Vol-3179/Short_13.pdf

[13] Bulgakova, O., Stepashko, V., Zosimov, V. ”Numerical study of the generalized iterative algorithm GIA GMDH with active neurons” in Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017, (2017), 1, pp. 496–500. doi**:** 10.1109/STC-CSIT.2017.8098836.
[14] Markus Spies (Ed.) Database and Expert Systems Applications. Inductive Building of Search Results Ranking Models to Enhance the Relevance of Text Information Retrieval. Los Alamitos: IEEE Computer Society, (2015) P. 291-295. doi: 10.1109/DEXA.2015.70

Efficiency Increasing of No-Reference Image Quality Assessment in UAV Applications

Oleg Ieremeiev[a], Vladimir Lukin[a], Krzysztof Okarma[b] and Karen Egiazarian[c]

[a] *National Aerospace University, Chkalova 17, Kharkiv, 61070, Ukraine*
[b] *West Pomeranian University of Technology in Szczecin, al. Piastów 17, Szczecin, 70-310, Poland*
[c] *Tampere University of Technology, Kalevantie 4, Tampere, FIN 33101, Finland*

Abstract

Unmanned aerial vehicle (UAV) imaging is a dynamically developing field, where the effectiveness of imaging applications highly depends on quality of the acquired images. No-reference image quality assessment is widely used for quality control and image processing management. However, there is a lack of accuracy and adequacy of existing quality metrics for human visual perception. In this paper, we demonstrate that this problem persists for typical applications of UAV images. We present a methodology to improve the efficiency of visual quality assessment by existing metrics for images obtained from UAVs, and introduce a method of combining quality metrics with the optimal selection of the elementary metrics used in this combination. A combined metric is designed based on a neural network trained to utilize subjective assessments of visual quality. The metric was tested using the TID2013 image database and a set of real UAV images with embedded distortions. Verification results have demonstrated the robustness and accuracy of the proposed metric.

Keywords

image quality assessment, no-reference metric, visual quality, UAV images, correlation analysis, artificial neural network

1. Introduction

A scope of applications of drones and other unmanned aerial vehicles (UAVs) has expanded rapidly in recent few decades. Since most of UAVs contain cameras, there is a growing interest in analysis and processing of visual data. UAVs mainly use optical band cameras, thus, the existing digital image processing solutions are applicable [1-2]. However, the mobility and autonomy of these systems can impose significant restrictions and one must consider all these factors.

The key problems of applying digital image processing in UAV applications are as follows:

1. UAVs require an adaptive integrated approach to suppress the present noise, motion blur, and other typical distortions, which can only partially be compensated by a camera stabilization.
2. For data transmission from drones, a wireless connection is used. The range and reliability of data transmission determine one of the key characteristics of UAVs – the flight range. In this sense, the efficiency of processing and compression of high-resolution data for transmission over a radio channel with limited bandwidth is decisive.
3. The data obtained at the end device, in addition to storage and more complex post-processing, can be used in various applications. Among them, high level vision tasks based on machine learning, such as detection, recognition, etc., become more widespread [1, 3, 4].

Common to all above mentioned challenges, there is a need to accurately assess image quality and measure distortion parameters, which will be used in image reconstruction methods, to enhance the

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: o.ieremeiev@khai.edu (O. Ieremeiev); v.lukin@khai.edu (V. Lukin); okarma@zut.edu.pl (K. Okarma); karen.eguiazarian@tuni.fi (K. Egiazarian)
ORCID: 0000-0001-7865-0570 (O. Ieremeiev); 0000-0002-1443-9685 (V. Lukin); 0000-0002-6721-3241 (K. Okarma); 0000-0002-8135-1085 (K. Egiazarian)

visual quality of the acquired images. In addition, an effective lossy compression is required. Certain results of UAV image processing have already been reported [5, 6, 7]. Nevertheless, robust methods that can accurately assess the visual component and determine the optimal parameters for subsequent image processing methods are required.

Image quality assessment (IQA) is usually applied by visual quality metrics. To improve their accuracy, some features of human perception are employed. There are two main classes of visual quality assessment methods. Full-reference (FR) visual quality metrics are widely used to verify image processing methods by evaluating the relative changes in image quality. No-reference (NR) metrics assess the quality based on the characteristics of the image itself and can be applied as a tool in many UAV applications [8, 9].

There are many developed NR IQA methods, but their common problem is a low accuracy, due to only limited amount of information available for analysis, and these metrics inability to accurately separate image elements (textures, borders, gradients, etc.) from distortions (noise, blur, etc.) [10, 11].

To design and verify visual quality metrics, special test image databases [11] are used. They contain images distorted by certain types of distortions. For each image, a visual quality score (mean opinion score (MOS)) is formed based on the results of a large number of subjective experiments with volunteers. Correlation analysis between metric values and MOS serves as a quantitative indicator of its compliance with human vision. Considering the most universal and large test image databases with tens of distortion types such as TID2013 [12], the efficiency of no-reference metrics usually does not exceed 0.5, according to the Spearman rank order correlation coefficient (SROCC).

Fortunately, one can increase accuracy of IQA using existing methods through their joint use, e.g., using methods presented in [13, 14]. In this paper, we propose a method of combining no-reference visual quality metrics based on an artificial neural network (ANN) that is focused on solving various problems of processing UAV images. Since many tasks with UAVs require the mobility of computing devices, the priority of this work is to ensure high accuracy of visual quality estimation while maintaining acceptable performance.

2. The efficiency of metrics for UAV purposes

Drones can lead to a significant amount of various distortions for an image during its acquisition, processing, compression and transmission over a communication channel. In this regard, the design of a combined metric requires the presence of test image databases that allow simulating such situations. As a result of the analysis of many image databases [11], we have chosen TID2013.

A distinctive feature of this image database is that it contains 24 types of various distortions, including such unique ones as bit errors in the transmission of compressed images. TID2013 contains 25 reference images that have been distorted by 24 types of distortion at 5 levels of intensity, for a total of 3000 test images. A complete list of distortions and their applicability to solving the current problem is given in Table 1.

Let us analyze the distortions listed in Table 1 and their relation to imaging from UAVs:

- Additive Gaussian noise (##1-2) is the basic model for representing most of the physical processes that cause noise. It is more pronounced in low light conditions.
- Spatially correlated noise (#3) is a characteristic of optical images due to the use of the Bayer filter or its modifications on sensors. It significantly increases with digital zoom.
- Impulse noise (#6) may be a manifestation of dead pixels and a lot of other causes such as coding/decoding artifacts.
- Quantization noise (#7) may occur during image acquisition and transformations.
- Blurring (#8) is one of the most relevant distortions due to the motion and vibrations of the UAV.
- Denoising (#9) is a manifestation of the noise reduction built into most cameras.
- Compression (##10-11) is a typical stage in the image processing chain to reduce data redundancy.
- Transmission errors (##12-13) are typical for wireless communication channels, especially over long distances.

- Changes in brightness, contrast and saturation (## 16-18) allow simulating changes in lighting conditions at different time instances of a day and weather conditions.
- Multiplicative noise (#19) is relevant because sensor noise is mostly signal-dependent.
- Noise (#20) allows the simulation of some artifacts of image processing and compression.
- Lossy compression of noisy images (#21) is a typical example of a real situation where an image with some noise is compressed.
- Chromatic aberration (#23) is a result of the refraction of light in the camera's optics.

Table 1
List of TID2013 distortions and their relevance for UAV purposes

##	Distortion type	Relevance for UAV imaging
1	Additive Gaussian noise	+
2	Additive noise (more intensive in color components)	+
3	Spatially correlated noise	+
4	Masked noise	–
5	High-frequency noise	–
6	Impulse noise	+
7	Quantization noise	+
8	Gaussian blur	+
9	Image denoising	+
10	JPEG compression	+
11	JPEG2000 compression	+
12	JPEG transmission errors	+
13	JPEG2000 transmission errors	+
14	Non-eccentricity pattern noise	–
15	Local block-wise distortions of different intensity	–
16	Mean shift (intensity shift)	+
17	Contrast change	+
18	Change of color saturation	+
19	Multiplicative Gaussian noise	+
20	Comfort noise	+
21	Lossy compression of noisy images	+
22	Image color quantization with dither	–
23	Chromatic aberrations	+
24	Sparse sampling	–

The listed 18 distortions comprehensively allow a use of the vast majority of noise types and distortions that can occur in UAV images or be the result of weather conditions. These distortion types give together 2250 test images from the TID2013 dataset that will be used in the paper.

Let us analyze the performance of the existing NR metrics on this subset of images. Since our task is to ensure high accuracy of estimation, the maximum possible number of different metrics is included. The SROCC values for the entire TID2013 database and the selected subset are given in Table. 2.

As it can be seen from the results in Table 2, the best performance is demonstrated by the ILNIQE metric, but its SROCC values (equal to 0.492 for all and 0.529 for the selected 18 UAV distortions) are relatively (inappropriately) low. It should be noted that Table 2 shows the absolute SROCC values because the metrics have been developed using different image databases that can evaluate the visual quality (MOS values) in two ways: as a higher value for better quality, or vice versa - a higher value as a larger difference from the perfect quality.

Table 2
SROCC values of no-reference IQA on TID2013 subsets

##	Metric	SROCC (All)	SROCC (UAV)	##	Metric	SROCC (All)	SROCC (UAV)
1	ILNIQE [15]	0.492	0.529	23	DIQU [34]	0.240	0.251
2	CORNIA [16]	0.435	0.521	24	SDQI [35]	0.224	0.248
3	HOSA [17]	0.471	0.515	25	DIPIQ [36]	0.140	0.209
4	C-DIIVINE[18]	0.373	0.448	26	MLV [37]	0.201	0.195
5	BLIINDS2 [19]	0.395	0.425	27	FISHBB [38]	0.145	0.152
6	BRISQUE[20]	0.367	0.416	28	JNBM [39]	0.141	0.152
7	BIQI [21]	0.405	0.409	29	DESIQUE[40]	0.069	0.150
8	SSEQ [22]	0.341	0.406	30	GMLOG [41]	0.109	0.139
9	NIQE [23]	0.313	0.403	31	NIQMC [42]	0.113	0.124
10	QAC [24]	0.372	0.379	32	ARISM [43]	0.145	0.109
11	SISBLIM_SM [25]	0.318	0.360	33	CPBDM [44]	0.112	0.109
12	LPSI [26]	0.395	0.357	34	LSSn [31]	0.168	0.105
13	LPC-SI [27]	0.323	0.354	35	PSS [31]	0.022	0.087
14	SISBLIM_SFB[25]	0.336	0.348	36	LSSs [31]	0.114	0.084
15	DIIVINE [28]	0.344	0.343	37	ARISMc [43]	0.138	0.081
16	BIBLE [29]	0.281	0.333	38	PSI [45]	0.001	0.075
17	OG-IQA [30]	0.276	0.327	39	SMETRIC[46]	0.097	0.074
18	BPRI [31]	0.229	0.313	40	FISH [38]	0.052	0.041
19	TCLT [32]	0.233	0.308	41	NR-PWN [47]	0.016	0.039
20	SISBLIM_WFB [25]	0.293	0.301	42	NMC [48]	0.054	0.033
21	MSGF-PR [33]	0.244	0.274	43	BLUR [49]	0.008	0.020
22	SISBLIM_WM [25]	0.239	0.265	44	NJQA [50]	0.100	0.007

3. The problem of metrics selection

It is possible to increase the accuracy of image quality assessing by combining several metrics. Successfully selected metrics are able to complement each other and provide a comprehensive analysis of the image taking into account various types of distortions. As it was shown in [13], the greatest efficiency is achieved through multi-parameter optimization using artificial neural networks. Combining the listed 44 metrics can potentially give the best accuracy of visual quality assessment. However, most of these metrics can make a low contribution requiring significant computing resources. High mobility and minimal computing costs are among the key requirements for UAV applications. Therefore, it is necessary to reduce the number of metrics without a significant decrease in the accuracy of IQA. Several possible solutions can be employed for the correct choice of elementary metrics (listed in Table 2) as inputs of an ANN, but not all of them are feasible or give an effective solution:

1. A complete enumeration of options is not possible in practice, since even for 5 or 10 incoming metrics, it will be necessary to calculate 1.6×10^{8} and 2.7×10^{16} combinations, respectively.
2. The choice of the best metrics with high SROCC rates or the exclusion of similar metrics with high cross-correlation values has shown insufficient efficiency in [51].
3. “Intelligent” selection of appropriate metrics. As a possible solution, the approach of using regularization was tested in [13] and proven to be effective. Lasso (least absolute shrinkage and selection operator) regularization is widely used in machine learning to reduce the model complexity and prevent overfitting. As a result of introducing restrictions, it allows determining the least important input features (corresponding metrics) and excludes them by setting zero weight coefficients. This approach can be applied to reduce the number of metrics.

To display the influence of the number of elementary metrics used on the accuracy of the trained ANN, we employ several of their combinations defined using Lasso in the range of values from the minimum 3-5 to all 44 metrics. The Lasso parameters were selected in such a way as to obtain non-zero weights for a given number of the metrics. Totally, 10 dimensions are considered in the paper: 4, 5, 7, 10, 16, 20, 25, 30, 35, and 44. Metric combinations with 16 metrics and less, which are focused on, are presented in Table 3.

Table 3
List of the metrics, defined by Lasso

Metrics' number	Metrics' names
4	ARISM, CORNIA, DIPIQ, ILNIQE
5	Above 4 + LPCSI
7	Above 5 + MLV, NIQMC
10	Above 7 + MSGF-PR, NIQE, PSS
16	Above 10 + C-DIIVINE, GMLOG, HOSA, JNBM, PSI, TCLT

4. Preliminary results

Despite the popularity of neural networks, their use in the field of image quality assessment has some limitations.

First, there are limited variety and size of datasets, because only image databases containing MOS values can be applied. It should be noted that due to the limited number of distortion levels and the variety of reference images, it can be assumed that some test images have unique properties and their distribution into training or test subset may affect the accuracy of the trained neural networks. Therefore, it is impossible to choose exactly which images should be in each of these sets. To ensure a result approaches the optimal one, for each ANN configuration over 100 repetitions with a random distribution of images on training (70%) and testing (30%, respectively) subsets have been completed.

Second, the choice of the type of ANN can have a significant impact on the final efficiency. Two types of networks are considered: feed-forward and cascade networks, which have a non-linear relationship between layers since the resulting value of each layer, including the input one, affects all subsequent layers.

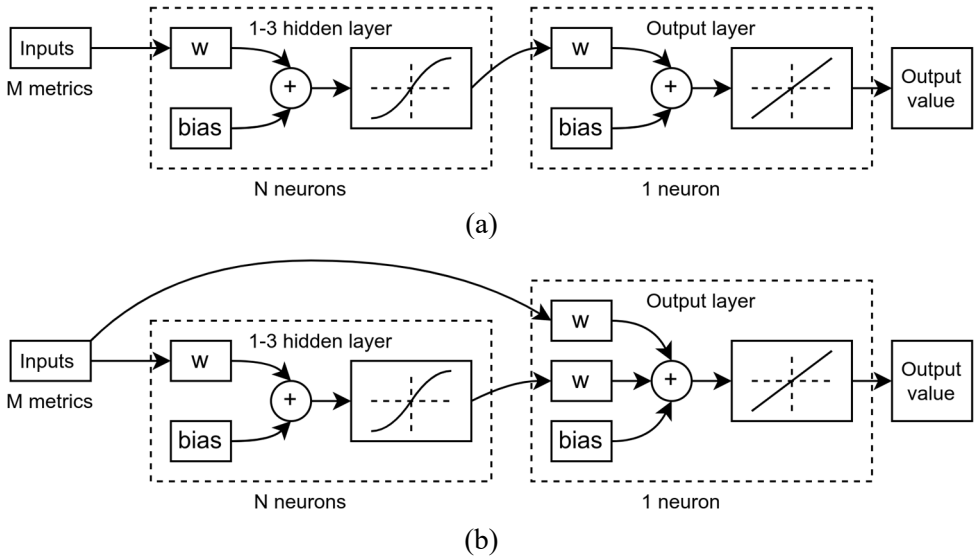


Figure 1: Generalized schemes of the used feed-forward (a) and cascade (b) networks

Further, the efficiency of ANN is also determined by its structure (the number of hidden layers and the number of neurons in each of them). Since a significant number of factors affecting the efficiency

of the final neural network have already been indicated, several basic configurations are used at the preliminary stage of the analysis. A more precise configuration of the ANN will be determined at the final stage of creating the combined metric. At this stage, variants of the neural network structure with 1-3 hidden layers are used. For each of them, there are two options for the number of neurons N in each layer: 1) in all layers, it is equal to the number of input metrics M (N = M), and 2) each next layer starting from the second one the number is divided by two (N1 = M, N2 = M/2, N3 = M/4). There are only 5 options totally because for a single-layer network they are identical.

As the activation function, a sigmoid function is used, which allows, regardless of the value ranges of the used metrics, to obtain, after the 1st hidden layer, the values in the fixed range [0,1]. This procedure allows us to implement the built-in function fitting and value normalization. This stage involves the construction of 10,000 variants of ANN (2 types × 5 configs × 100 repetitions × 10 metric combinations). All calculations were performed using the MatLab software.

Let us analyze the results obtained after training all these ANNs. The main dependence is that the accuracy of the combined metric grows with the number of elementary metrics used. The maximum is achieved for all 44 metrics. The graph of SROCC dependence on the number of metrics is shown in Fig. 2 for ANNs with maximum SROCC rates among repetitions of each configuration for the feed-forward network.

Based on these results, several conclusions can be drawn. Thus, the use of an ANN for metrics combination is an effective solution for UAV applications, since even the minimal number of them (4) significantly exceeds in accuracy the maximum result among elementary metrics (SROCC = 0.53). The current 5 configurations of the ANN structures give similar indicators, their comparison will be carried out in more detail later. This graph allows making some recommendations for choosing the structure of an ANN depending on the requirements and constraints of the problem solved. For example, if it is necessary to ensure maximum performance, the desired choice would be a combined metric of 5 elementary ones, its result reaches SROCC = 0.74, which is much higher than for 4 metrics, but a further increase of accuracy with the number of input parameters is slow. Nevertheless, if accuracy or balance with performance is a priority, then the options of 10 or 16 elementary metrics can be useful. Their accuracy reaches 0.82 – 0.84 of SROCC. Further, the accuracy at the level of 0.85 is practically independent of the number of metrics. Considering that one of the requirements of this study is to maintain acceptable performance with high accuracy, we will use a combined metric consisting of 10 elementary metrics.

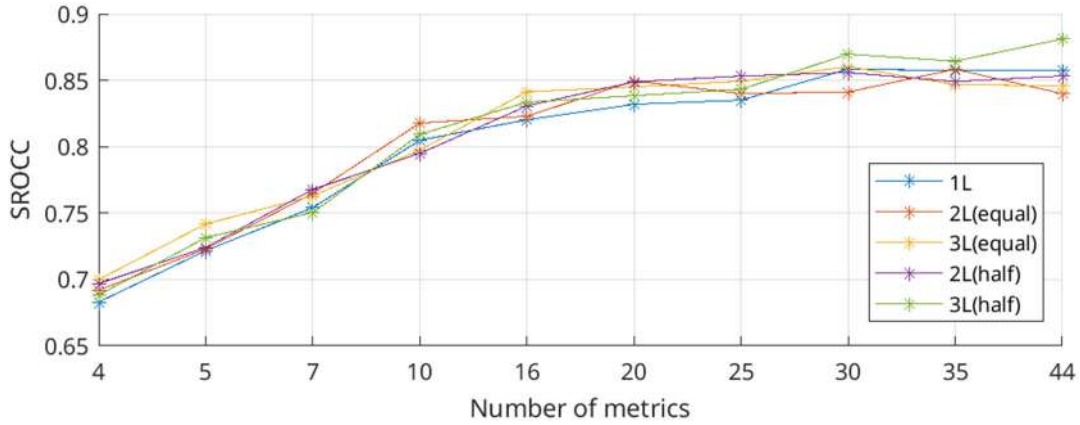


Figure 2: Dependence of SROCC on the number of elementary metrics selected by the Lasso criterion

To display the main statistical indicators and some problems, Fig. 3 shows a box chart for 4 (full graph and limited range higher than 0.5 under it), 5 (similarly to the previous one), 10, and all 44 metrics. Its advantage is the ability to display simultaneously the median, the lower (0.25) and upper (0.75) quartiles, any outliers (computed using the interquartile range), and the minimum and maximum values that are not outliers. From these graphs, it can be noted that with a small number of metrics (4 and 5), the complexity of the neural network (number of neurons) is not enough for proper training, as a result of which anomalous results were obtained - incorrectly trained neural networks with indicators below individual metrics. This is also the problem of multilayer neural networks with

fewer neurons in each layer. For 10 and more metrics, this problem is no longer observed. The highest values for each presented network configuration are already denoted in Fig 1. Quantitative indicators of the best neural networks for the feed-forward network from Fig. 1 and Fig. 2 are given in Table 4, where M means the number of elementary metrics.

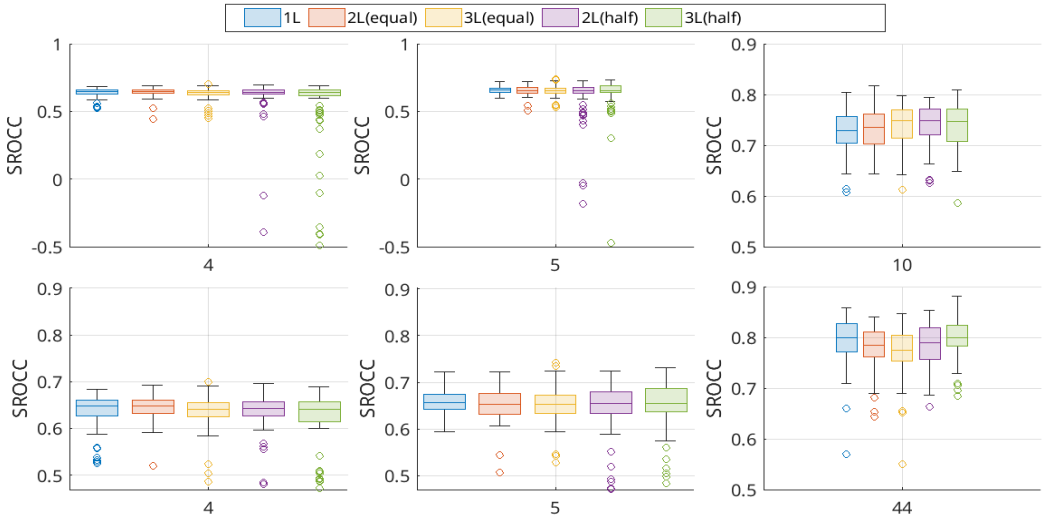


Figure 3: Box charts of the results of the obtained neural networks for 4, 5, 10, and 44 input metrics.

Table 4
Results of the best feed-forward networks for different numbers of inputs (4, 5, 10, and 44)

NN config	Description (in NN layers)	SROCC			
		M = 4	M = 5	M = 10	M = 44
1	[M]	0.683	0.722	0.805	0.858
2	[M, M]	0.692	0.723	0.818	0.840
3	[M, M, M]	0.700	0.742	0.797	0.846
4	[M, M/2]	0.697	0.724	0.795	0.853
5	[M, M/2, M/4]	0.688	0.732	0.809	0.881

5. Final network modifications

In the first phase of experiments, when forming a neural network for 10 input metrics, the following configurations were used for neural networks with 1-3 hidden layers: [10], [10, 10], [10, 10, 10], [10, 5] and [10, 5, 2].

The general trend in Fig. 2 shows that the number of neurons in layers less than 10 may not be enough. Therefore, additional configurations with a number of neurons up to 20 per layer (×2 compared to the number of input metrics) were additionally built. More than 30 configurations for both network types have been used and the best results of the ANN for each number of hidden layers and some statistics are partly shown in Table 5. It shows lists of neural network configurations, both the best 2 from the initial five and additionally trained for 10 input metrics (50 repetitions).

To evaluate the effectiveness of each configuration and both types of networks, some statistical indicators are given: the maximum (best neural network) and minimum value, median, skewness, and quartiles 0.75 and 0.95. Skewness is a measure of the asymmetry of the data around the sample mean. If skewness is positive, the data spread out more to the higher values. The skewness of the normal distribution (or any perfectly symmetric distribution) is zero.

The maximum performance for both types of networks in Table 5 has been achieved by configuration #8. Despite the random learning process, in general, for a feed-forward network, an increase in the number of neurons to 20 leads to an increase in accuracy. This is also confirmed by the values of the quartiles 0.75 and 0.95. A further increase in the number of neurons does not provide a significant improvement. According to skewness values, it can be noted that there is a slight tendency

toward obtaining neural networks with low performance, and in the worst cases they differ a little from elementary metrics (SROCC can be less than 0.6). Cascade neural networks do not provide any advantages demonstrating somewhat lower performance for almost all configurations. This network shows the advantage in terms of maximum SROCC for configurations with a small number of neurons (#2 and #4), therefore, it is presumably the most effective for solutions with a small amount of input data and simpler layer structures.

According to the results of Table 5, the ANN with the maximum Spearman correlation coefficient of 0.8307 was chosen as a combined metric for visual quality assessment tasks. The list of metrics used in it and a visual comparison of its effectiveness for elementary metrics is shown in Fig. 4. This metric is available at https://github.com/OlegIeremeiev/CNNM-UAV.git .

Table 5
Results of the best feed-forward networks for 10 inputs

NN config	Description (in NN layers)	SROCC Max	Min	Median	Skewness	0.75 Quartile	0.95 Quartile
			Feed-forward network				
1	[10] [10]	0.8178	0.6433	0.7351	-0.0261	0.7621	0.8033
2	[10] [5]	0.7949	0.6256	0.7487	-0.9202	0.7716	0.7899
3	[20]	0.8111	0.6475	0.7483	-0.5799	0.7704	0.8041
4	[20] [10]	0.8074	0.6312	0.7563	-0.7113	0.7862	0.8024
5	[20] [20]	0.8280	0.6787	0.7625	-0.2461	0.7906	0.8209
6	[15] [10] [10]	0.8214	0.6430	0.7452	-0.3190	0.7726	0.8116
7	[10] [15] [20]	0.8050	0.6383	0.7386	-0.3225	0.7608	0.8009
8	**[20] [15] [10]**	**0.8307**	0.6500	0.7607	-0.5689	0.7806	0.8161
9	[10] [20] [15]	0.8195	0.5945	0.7387	-0.6796	0.763	0.8120
			Cascade network				
1	[10] [10]	0.8010	0.6074	0.7507	-1.2935	0.7698	0.7902
2	[10] [5]	0.8213	0.5437	0.7435	-1.6415	0.7643	0.7844
3	[20]	0.7989	0.5954	0.7521	-1.0453	0.7685	0.7966
4	[20] [10]	0.8176	0.5606	0.7582	-1.5616	0.7811	0.8098
5	[20] [20]	0.8080	0.6108	0.7577	-1.5696	0.7784	0.8000
6	[15] [10] [10]	0.8126	0.6253	0.7487	-0.9623	0.7739	0.8040
7	[10] [15] [20]	0.8185	0.6326	0.7618	-0.9489	0.7821	0.8041
8	[20] [15] [10]	0.8214	0.6193	0.7726	-1.3170	0.7895	0.8053
9	[10] [20] [15]	0.8177	0.6681	0.7572	-0.2515	0.7797	0.8122

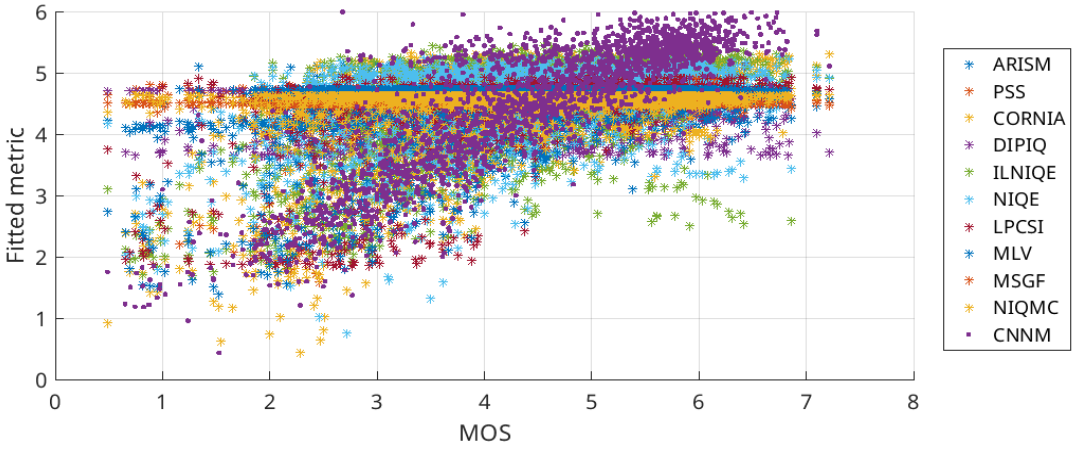


Figure 4: Scatter plot of the 10 elementary metrics and the combined metric (CNNM) including them.

A visual representation of the effectiveness of assessing the quality of certain types of distortions is shown in the graph in Fig. 5. The numbers of distortions correspond to the serial number of the distortions selected for analysis (see Table 1). It can be seen that the combined metric provides consistently high results with a decrease in accuracy at distortions #12 (mean shift) and #14 (change of color saturation), these distortions are problematic for all the metrics used in the paper.

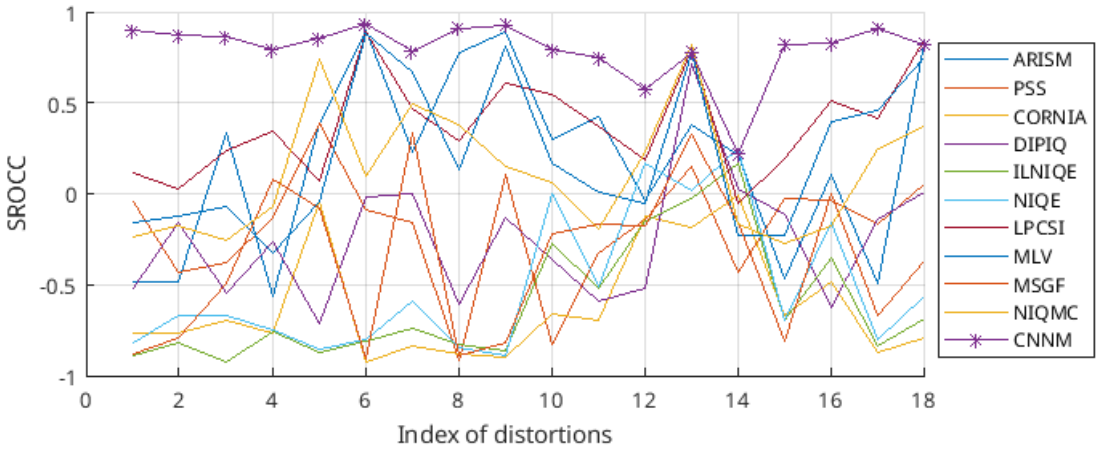


Figure 5: Dependency of the metrics' SROCC values on the type of distortion (absolute values)

6. Combined metric analysis

The purpose of creating a combined metric was to improve the accuracy of the visual quality assessment of images in various UAV tasks. However, there is a limitation: general-purpose color image database TID2013 with the corresponding MOS values was taken to train the neural network. Therefore, it is necessary to analyze the effectiveness of the obtained metric in practice for real images.

It should be noted that the application area and its inherent types of distortion significantly affect the results obtained. Thus, in [14], the visual quality metric was proposed for the assessment of remote sensing images. Its SROCC value reached the level of 0.8813. At the same time, verification on UAV-related distortions from TID2013 showed significantly worse results - SROCC has decreased to 0.7083. The reason lies in the different sets of distortions. In particular, transmission errors are rare in remote sensing practice, since these systems operate in more static and predictable conditions. Distortions in brightness and contrast as a factor of weather and daylight conditions changing were also not taken into account in the design in [14]. This confirms the fact that individual metrics are often not enough for application areas with unique features and the combined approach based on neural networks allows for an increase of 50% or more.

The practicality and applicability of the proposed solution can only be assessed on the basis of real images from the UAV. At the same time, this approach has significant limitations: the absence of MOS values and the complexity of obtaining images with all the considered distortions and needed combinations. Taking this into account, a number of assumptions and simplifications have been made, and the results obtained are mostly illustrative.

1. Verification of visual metrics requires MOS, which values can only be obtained from a significant amount of subjective experiments and require considerable time. The first simplification is that the missing MOS values can to some extent be replaced by objective indicators, the accuracy of which significantly exceeds the analyzed metrics. For a comparative analysis of the combined and individual metrics, this may be sufficient. Such a condition can be provided by full-reference quality metrics - the accuracy of some of them reaches SROCC = 0.9 for the entire TID2013 and more than 0.96 for certain types of distortions and significantly exceeds SROCC for existing no-reference metrics.

2. It is technically difficult to ensure the presence of real test images with the considered distortions, therefore, it is proposed to artificially simulate their presence by adding the distortions under the interest of different intensities to the selected images.
3. The level of distortion should preferably have a wide range of intensities from inconspicuous to significant.

To verify the metrics, real images from UAVs were used. As a basis, some images of the UAVDT (Unmanned Aerial Vehicle Benchmark Object Detection and Tracking) dataset [52] were taken, examples of which are shown in Fig. 6. The dataset contains more than 40,000 images with a resolution of 1080 × 540 pixels. Of these, 16 images were selected with different terrain, daylight, and weather conditions.

Figure 6: Examples of reference images of the UAV test set.

Creation of test images with the necessary types of distortion requires special skills. TID2013 distortions were generated in accordance with a certain strategy, however, their generation code is not available. Therefore, our mechanisms for generating distortions are used in the paper, and from the list of selected types of distortions, 9 main ones are taken into account:

- Gaussian white noise;
- Multiplicative noise;
- Gaussian blur;
- Denoising (applying BM3D filter to images with Gaussian white noise);
- JPEG and JPEG2000 compression;
- Brightening, darkening, and mean shift (darkening and lightening).

According to the variety of intensities, 9 different levels were chosen for a more accurate gradation of distortion, in contrast to 5 levels for TID2013. Their intensity varies from inconspicuous to significant. The distribution of peak signal-to-noise ratio (PSNR) values is shown in Fig. 7.

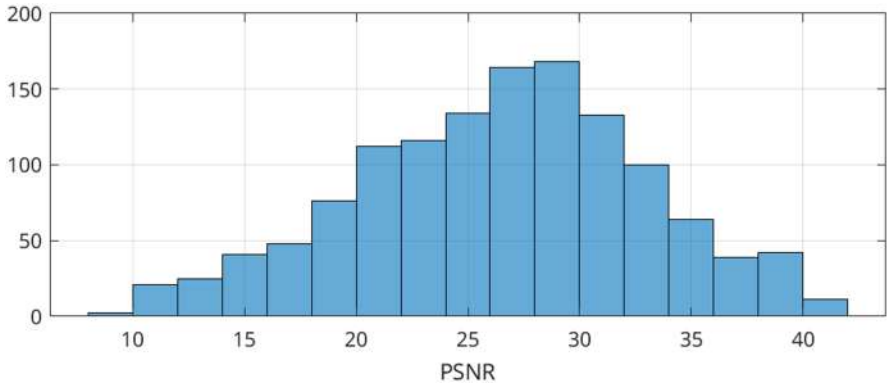


Figure 7: Histogram of PSNR values of the test images

As a result, the verification test set based on real UAV images consists of 1296 images (16 images x 9 distortions x 9 intensity levels).

In the role of MOS values for no-reference metrics verification, the best full-reference quality metrics are used. The SROCC values of some well-known FR IQA for all TID2013 images and UAV -related test set are given in Table 6. Since their problems and solutions are similar to those solved in the article, a combined full-reference metric was formed to improve the accuracy. It uses the metrics listed in Table 6 as input and consists of a two-layer neural network (marked as C_MOS) with the number of neurons [16, 8] and all other parameters listed above. Since its SROCC for the task considered is almost 0.04 higher than for the best of elementary metrics, this combined metric has been chosen as the analog of MOS for UAV test images.

Table 6

SROCC values of the full-reference visual metrics on the TID2013 image dataset

Metric	VSI [53]	PSIM [54]	MDSI [55]	HaarPSI [56]	UNIQUE [57]	CVSSI [58]	IQM2 [59]	ADM [60]	C_MOS
SROCC	**0.8967**	0.8926	0.8897	0.8730	0.8599	0.8090	0.7955	0.7861	**0.9107**
SROC(UAV)	0.8274	0.8519	**0.8873**	0.8811	0.8496	0.8478	0.8507	0.8075	**0.9261**

The results of the verification of the combined and elementary no-reference metrics are shown in Table 7. In addition to the overall assessment, the SROCC values for individual types of distortions are also shown. The two best results for each type of distortion are highlighted in bold.

Table 7

The results of verification of the no-reference metrics on the UAV test set

Distortion	ARISM	PSS	CORNIA	DIPIQ	ILNIQE	NIQE	LPCSI	MLV	MSGF	NIQMC	CNNM
All	0.069	0.307	0.581	0.109	0.548	**0.616**	0.330	0.024	0.499	0.074	**0.659**
AWGN	0.804	0.202	0.861	0.470	**0.890**	**0.886**	0.511	0.239	0.794	0.180	0.874
Multiplicative noise	0.702	0.257	0.784	0.053	0.722	0.701	0.557	0.313	0.624	0.309	**0.903**
Blur	0.942	0.898	0.904	0.254	0.869	0.900	**0.966**	0.949	0.784	0.200	**0.956**
Denoise	**0.360**	0.072	0.186	0.196	0.004	0.153	0.020	0.181	**0.228**	0.118	0.154
JPEG	0.769	**0.958**	**0.868**	0.534	0.713	0.728	0.166	0.087	0.866	0.367	0.787
JP2k	0.134	0.378	**0.738**	0.337	0.240	0.632	0.054	0.486	0.028	0.433	**0.764**
Brighten	**0.524**	0.070	0.027	0.027	0.166	0.035	0.183	0.372	0.197	0.378	**0.474**
Darken	0.359	0.044	0.057	**0.575**	0.341	0.301	0.342	**0.460**	0.021	0.052	0.281
Mean shift	**0.515**	0.328	0.027	0.429	0.335	0.442	0.169	0.391	0.253	0.054	**0.520**

From the obtained results, it can be seen that despite the limitations of the approximate MOS values, the combined metric provides the maximum overall accuracy and is one of the best for most of the indicated types of distortions, providing the best balance between various distortions. It should be noted that these results have been obtained for the most common types of distortions, which are commonly used in the design of elementary metrics. Considering the types of distortions used in TID2013, but not modeled in this set (e.g. transmission errors, etc.), it can be expected that the combined metric can have additional benefits by providing more stable visual quality estimation.

7. Conclusions

The paper is devoted to visual quality assessment of UAV images, which is actual for automating the image processing and improving image quality for UAV applications. A list of more than 40 known no-reference visual quality metrics is considered. To analyze the effectiveness of visual quality metrics, the TID2013 image database and a subset with actual types of distortions have been selected. The verification of existing visual quality metrics has shown an accuracy of less than 0.53 for the best one and less than 0.3 for most metrics according to SROCC. Therefore, the method of combining visual quality metrics using the neural network has been proposed to improve the accuracy of visual quality assessment. The problem of the optimal choice of elementary metrics for reducing the redundancy and rational use of computing resources has been considered and the solution based on the Lasso regularization method has been proposed, which determines the weight coefficient equal to 0 for the excluded and least important metrics. Training the neural networks of different types and their configurations has been carried out, taking into account the limitations of the test image database used in experiments. The analysis of the effectiveness of this approach, which reaches a result of about 0.85 for 20 metrics or more, has been carried out, and the dependence on the number of metrics used in the paper together with the main statistics is shown. For 10 metrics, as the optimal solution for high accuracy and performance, the results have been refined with the training of additional configurations of the structure of neural networks. It is shown that the accuracy of the final combined metric reaches SROCC = 0.83.

To evaluate the effectiveness of the metric on real images, a test image database of almost 1300 images was formed. As an alternative to the missing MOS values, a combined full-reference metric has been created. Its accuracy reaches 0.926 for the used TID2013 distortion set and is significantly higher than the values of any no-reference metric, which is acceptable for their comparison. It is shown that, on this test set, the obtained metric provides the best result.

In the future, research in this area can be expanded by adding new distortions typical for UAV images and new neural network models including deep-learning models of limited complexity.

8. References

[1] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 4th ed., Pearson, New York, NY, 2018.

[2] W. Burger, M.J. Burge, Principles of Digital Image Processing, Springer, New York, NY, 2009.

[3] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, 2nd ed., Springer, New York, NY, 2009. doi: 10.1007/978-0-387-84858-7.

[4] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[5] R. Wang, X. Xiao, B. Guo, Q. Qin, R. Chen, An Effective Image Denoising Method for UAV Images via Improved Generative Adversarial Networks, Sensors 18 (2018) 1–23. doi: 10.3390/s18071985.

[6] T. Sieberth, R. Wackrow, J. H. Chandler, UAV image blur - its influence and ways to correct it, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-1/W4 (2015) 33–39. doi: 10.5194/isprsarchives-XL-1-W4-33-2015.

[7] V. S. Alfio, D. Costantino, M. Pepe, Influence of Image TIFF Format and JPEG Compression Level in the Accuracy of the 3D Model and Quality of the Orthophoto in UAV Photogrammetry, J. Imaging 6 (2020) 1–22. doi: 10.3390/jimaging6050030.

[8] M. Kedzierski, D. Wierzbicki, Radiometric quality assessment of images acquired by UAV's in various lighting and weather conditions, Measurement 76 (2015) 156–169. doi: 10.1016/j.measurement.2015.08.003.
[9] G. Koretsky, J. Nicoll, M. Taylor, A Tutorial on Electro-Optical/ Infrared (EO/IR) Theory and Systems, IDA Document D-4642, 2013.
[10] W. Lin, C.-C. Jay Kuo, Perceptual Visual Quality Metrics: A Survey, Journal of Visual Communication and Image Representation 22 (2011) 297–312. doi: 10.1016/j.jvcir.2011.01.005.
[11] Y. Niu, Y. Zhong, W. Guo, Y. Shi, P. Chen, 2D and 3D Image Quality Assessment: A Survey of Metrics and Challenges, IEEE Access 7 (2018) 782–801. doi: 10.1109/ACCESS.2018.2885818.
[12] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, etc, Image database TID2013: Peculiarities, results and perspectives, Signal Processing: Image Communication 30 (2015), pp. 57–77. doi: 10.1016/j.image.2014.10.009.
[13] O. Ieremeiev, V. Lukin, K. Okarma, K. Egiazarian, Full-Reference Quality Metric Based on Neural Network to Assess the Visual Quality of Remote Sensing Images, Remote Sensing 12 (2020) 1–31. doi: 10.3390/rs12152349.
[14] A. Rubel, O. Ieremeiev, V. Lukin, J. Fastowicz, K. Okarma, Combined No-Reference Image Quality Metrics for Visual Quality Assessment Optimized for Remote Sensing Images, Applied Sciences 12 (2022) 1–19. doi: /10.3390/app12041986.
[15] L. Zhang, L. Zhang, A.C. Bovik, A Feature-Enriched Completely Blind Image Quality Evaluator, IEEE Trans. Image Processing 24 (2015) 2579–2591. doi: 10.1109/TIP.2015.2426416
[16] P. Ye, J. Kumar, L. Kang, D. Doermann, Unsupervised Feature Learning Framework for No-Reference Image Quality Assessment, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR '2012, Providence, RI, USA, 2012, pp. 1098–1105. doi: 10.1109/CVPR.2012.6247789.
[17] J. Xu, P. Ye, Q. Li, H. Du, Y. Liu, D. Doermann, Blind Image Quality Assessment Based on High Order Statistics Aggregation, IEEE Trans. Image Process. 25 (2016) 4444–4457. doi: 10.1109/TIP.2016.2585880.
[18] Y. Zhang, A.K. Moorthy, D.M. Chandler, A.C. Bovik, C-DIIVINE: No-Reference Image Quality Assessment Based on Local Magnitude and Phase Statistics of Natural Scenes, Signal Process. Image Commun. 29 (2014) 725–747. doi: 10.1016/j.image.2014.05.004.
[19] M. A. Saad, A. C. Bovik, C. Charrier, Blind Image Quality Assessment: A Natural Scene Statistics Approach in the DCT Domain, IEEE Trans. Image Process. 21 (2012) 3339–3352. doi: 10.1109/TIP.2012.2191563.
[20] A. Mittal, A. K. Moorthy, A. C. Bovik, No-Reference Image Quality Assessment in the Spatial Domain, IEEE Trans. Image Process. 21 (2012) 4695–4708. doi: 10.1109/TIP.2012.2214050.
[21] A. K. Moorthy, A.C. Bovik, A Two-Step Framework for Constructing Blind Image Quality Indices, IEEE Signal Process. Lett. 17 (2010) 513–516. doi: 10.1109/LSP.2010.2043888.
[22] L. Liu, B. Liu, H. Huang, A.C. Bovik, No-Reference Image Quality Assessment Based on Spatial and Spectral Entropies, Signal Process. Image Commun. 29 (2014) 856–863. doi: 10.1016/j.image.2014.06.006.
[23] A. Mittal, R. Soundararajan, A. C. Bovik, Making a "Completely Blind" Image Quality Analyzer, IEEE Signal Process. Lett. 20 (2013) 209–212. doi: 10.1109/LSP.2012.2227726.
[24] W. Xue, L. Zhang, X. Mou, Learning without Human Scores for Blind Image Quality Assessment, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR '2013, Portland, OR, USA, 2013, pp. 995–1002. doi: 10.1109/CVPR.2013.133.
[25] K. Gu, G. Zhai, X. Yang, W. Zhang, Hybrid No-Reference Quality Metric for Singly and Multiply Distorted Images, IEEE Trans. Broadcast. 60 (2014) 555–567. doi: 10.1109/TBC.2014.2344471.
[26] Q. Wu, Z. Wang, H. Li, A Highly Efficient Method for Blind Image Quality Assessment, in: Proceedings of the IEEE Int. Conf. on Image Processing, ICIP '2015, Quebec City, QC, Canada, 2015, pp. 339–343, 2015. doi: 10.1109/ICIP.2015.7350816.
[27] R. Hassen, Z. Wang, M. M. A. Salama, Image Sharpness Assessment Based on Local Phase Coherence, IEEE Trans. Image Process. 22 (2013) 2798–2810. doi: 10.1109/TIP.2013.2251643.

[28] A. K. Moorthy, A. C. Bovik, Blind Image Quality Assessment: From Natural Scene Statistics to Perceptual Quality, IEEE Trans. Image Process 20 (2011) 3350–3364. doi: 10.1109/TIP.2011.2147325.
[29] L. Li, W. Lin, X. Wang, G. Yang, K. Bahrami, A. C. Kot, No-Reference Image Blur Assessment Based on Discrete Orthogonal Moments, IEEE Trans. Cybern. 46 (2016) 39–50. doi: 10.1109/TCYB.2015.2392129.
[30] L. Liu, Y. Hua, Q. Zhao, H. Huang, A. C. Bovik, Blind Image Quality Assessment by Relative Gradient Statistics and Adaboosting Neural Network, Signal Process. Image Commun. 40 (2016) 1–15. doi: 10.1016/j.image.2015.10.005.
[31] X. Min, K. Gu, G. Zhai, J. Liu, X. Yang, C.W. Chen, Blind Quality Assessment Based on Pseudo-Reference Image, IEEE Trans. Multimed. 20 (2018) 2049–2062. doi: 10.1109/TMM.2017.2788206.
[32] Q. Wu, H. Li, F. Meng, K.N. Ngan, B. Luo, C. Huang, B. Zeng, Blind Image Quality Assessment Based on Multichannel Feature Fusion and Label Transfer, IEEE Trans. Circuits Syst. Video Technol. 26 (2016) 425–440. doi: 10.1109/TCSVT.2015.2412773.
[33] Q. Wu, H. Li, F. Meng, K. N. Ngan, S. Zhu, No Reference Image Quality Assessment Metric via Multi-Domain Structural Information and Piecewise Regression, J. Vis. Commun. Image Represent. 32 (2015) 205–216. doi: 10.1016/j.jvcir.2015.08.009.
[34] L. Li, Y. Yan, Z. Lu, J. Wu, K. Gu, S. Wang, No-Reference Quality Assessment of Deblurred Images Based on Natural Scene Statistics, IEEE Access 5 (2017) 2163–2171. doi: 10.1109/ACCESS.2017.2661858.
[35] M. Rakhshanfar, M. A. Amer, Sparsity-Based No-Reference Image Quality Assessment for Automatic Denoising, Signal Image Video Process. 12 (2018) 739–747. doi: 10.1007/s11760-017-1215-3.
[36] K. Ma, W. Liu, T. Liu, Z. Wang, D. Tao, DipIQ: Blind Image Quality Assessment by Learning-to-Rank Discriminable Image Pairs, IEEE Trans. Image Process. 26 (2017) 3951–3964. doi: 10.1109/TIP.2017.2708503.
[37] K. Bahrami, A. C. Kot, A Fast Approach for No-Reference Image Sharpness Assessment Based on Maximum Local Variation, IEEE Signal Process. Lett. 21 (2014) 751–755. doi: 10.1109/LSP.2014.2314487.
[38] P. V. Vu, D. M. Chandler, A Fast Wavelet-Based Algorithm for Global and Local Image Sharpness Estimation, IEEE Signal Process. Lett. 19 (2012) 423–426. doi: 10.1109/LSP.2012.2199980.
[39] R. Ferzli, L. J. Karam, A No-Reference Objective Image Sharpness Metric Based on the Notion of Just Noticeable Blur (JNB), IEEE Trans. Image Process. 18 (2009) 717–728. doi: 10.1109/TIP.2008.2011760.
[40] Y. Zhang, D. M. Chandler, No-Reference Image Quality Assessment Based on Log-Derivative Statistics of Natural Scenes, J. Electron. Imaging 22 (2013) 043025. doi: 10.1117/1.JEI.22.4.043025.
[41] W. Xue, X. Mou, L. Zhang, A. C. Bovik, X. Feng, Blind Image Quality Assessment Using Joint Statistics of Gradient Magnitude and Laplacian Features, IEEE Trans. Image Process. 23 (2014) 4850–4862. doi: 10.1109/TIP.2014.2355716.
[42] K. Gu, W. Lin, G. Zhai, X. Yang, W. Zhang, C.W. Chen, No-Reference Quality Metric of Contrast-Distorted Images Based on Information Maximization, IEEE Trans. Cybern. 47 (2017) 4559–4565. doi: 10.1109/TCYB.2016.2575544.
[43] K. Gu, G. Zhai, W. Lin, X. Yang, W. Zhang, No-Reference Image Sharpness Assessment in Autoregressive Parameter Space, IEEE Trans. Image Process. 24 (2015) 3218–3231. doi: 10.1109/TIP.2015.2439035.
[44] N. D. Narvekar, L. J. Karam, A No-Reference Perceptual Image Sharpness Metric Based on a Cumulative Probability of Blur Detection, in Proceedings of the International Workshop on Quality of Multimedia Experience, QoMEx '2009, San Diego, CA, USA, 2009, pp. 87–91. doi: 10.1109/QOMEX.2009.5246972.
[45] C. Feichtenhofer, H. Fassold, P. Schallauer, A Perceptual Image Sharpness Metric Based on Local Edge Gradient Analysis, IEEE Signal Process. Lett. 20 (2013) 379–382. doi: 10.1109/LSP.2013.2248711.

[46] N. N. Ponomarenko, V. V. Lukin, O. I. Eremeev, K. O. Egiazarian, J. T. Astola, Sharpness Metric for No-Reference Image Visual Quality Assessment, SPIE 8295 (2012) 829519. doi: 10.1117/12.906393.
[47] T. Zhu, L. Karam, A No-Reference Objective Image Quality Metric Based on Perceptually Weighted Local Noise, EURASIP J. Image Video Process. (2014) 1–5. doi: 10.1186/1687-5281-2014-5.
[48] Y. Gong, I. F. Sbalzarini, Image Enhancement by Gradient Distribution Specification, Lecture Notes in Computer Science 9009 (2015) 47–62. doi: 10.1007/978-3-319-16631-5_4.
[49] F. Crété-Roffet, T. Dolmiere, P. Ladret, M. Nicolas, The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric, In Proceedings of the Human Vision and Electronic Imaging XII, HVEI '2007, San Jose, CA, USA, 2007, pp. 649201. doi: 10.1117/12.702790.
[50] S. A. Golestaneh, D. M. Chandler, No-Reference Quality Assessment of JPEG Images via a Quality Relevance Map, IEEE Signal Process. Lett. 21 (2014) 155–158. doi: 10.1109/LSP.2013.2296038.
[51] O. Ieremeiev, V. Lukin, K. Okarma, K. Egiazarian, B. Vozel, On properties of visual quality metrics in remote sensing applications, in: Proceedings of the IS&T Int'l. Symp. on Electronic Imaging: Image Processing: Algorithms and Systems, IPAS '2022, San Francisco, CA, USA, 2022, pp. 354-1-354-6. doi: 10.2352/EI.2022.34.10.IPAS-354.
[52] D. Du, Y. Qi, H.g Yu, Y. Yang, K. Duan, G. Li, W.g Zhang, Q. Huang, Q. Tian, The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking, in: Proceedings of the European Conference on Computer Vision, ECCV'2018. doi: 10.48550/arXiv.1804.00518.
[53] L. Zhang, Y. Shen, H. Li, VSI: A Visual Saliency-Induced Index for Perceptual Image Quality Assessment, IEEE Trans. Image Process 23 (2014) 4270–4281. doi:10.1109/TIP.2014.2346028.
[54] K. Gu, L. Li, H. Lu, X. Min, W. Lin, A Fast Reliable Image Quality Predictor by Fusing Micro- and Macro-Structures, IEEE Trans. Ind. Electron. 64 (2017) 3903–3912. doi:10.1109/TIE.2017.2652339.
[55] H. Ziaei Nafchi, A. Shahkolaei, R. Hedjam, M. Cheriet, Mean Deviation Similarity Index: Efficient and Reliable Full-Reference Image Quality Evaluator, IEEE Access 4 (2016) 5579–5590. doi:10.1109/ACCESS.2016.2604042.
[56] R. Reisenhofer, S. Bosse, G. Kutyniok, T. Wiegand, A Haar wavelet-based perceptual similarity index for image quality assessment, Signal Process. Image Commun. 61 (2018) 33–43. doi:10.1016/j.image.2017.11.001.
[57] D. Temel, M. Prabhushankar, G. AlRegib, UNIQUE: Unsupervised Image Quality Estimation. IEEE Signal Process. Lett. 23 (2016) 1414–1418. doi:10.1109/LSP.2016.2601119.
[58] H. Jia, L. Zhang, T. Wang, Contrast and Visual Saliency Similarity-Induced Index for Assessing Image Quality, IEEE Access 6 (2018) 65885–65893. doi:10.1109/ACCESS.2018.2878739.
[59] E. Dumic, S. Grgic, M. Grgic, IQM2: new image quality measure based on steerable pyramid wavelet transform and structural similarity index, Signal Image Video Process. 8 (2014) 1159–1168. doi:10.1007/s11760-014-0654-3.
[60] S. Li, F. Zhang, L. Ma, K. N. Ngan, Image Quality Assessment by Separately Evaluating Detail Losses and Additive Impairments, IEEE Trans. Multimed. 13 (2011) 935–949. doi: 10.1109/TMM.2011.2152382.

Improving Low-Contrast Images Using Frequency and Fuzzy Transformations

Lyudmila Akhmetshina[a], Artyom Yegorov[a] and Stanislav Mitrofanov[a]

[a] *Dnieper National University Named by Oles Honchar, Gagarin Avenue, house 72, Dnieper, 49010, Ukraine*

Abstract

The informational capabilities of a method for processing grayscale images aimed at improving contrast and increasing object detail to increase the accuracy of diagnosis based on them are presented. The proposed adaptive composite algorithm is based on multi-stage processing, which includes the use of two-dimensional frequency Fourier transformation and the method of fuzzy intensification in the spatial domain, and makes several transitions between different feature spaces. The application of the Fourier transform involves the correction of its coefficients and the reconstruction of the image by inverse transformation. Only arguments of complex coefficients can be adjusted. The impact of the frequency transformation parameters on the detail of the resulting image is analyzed. The method of fuzzy intensification is used as a refinement for the second stage of frequency transformation. The results of processing are presented on the example of real X-ray images.

Keywords 1

low-contrast images, two-dimensional Fourier transform, fuzzy intensification operator.

1. Introduction

The number of areas in which the raw data comes in the form of images is constantly growing. These include surveillance, monitoring, polygraphy, medicine and many other areas. The role of computer vision systems, which implement methods of improving the quality of digital images for further visual or automatic analysis in order to make accurate decisions based on them, is growing [1].

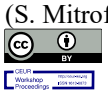
For example, medical images, which are an integral part of the diagnosis of various diseases, usually have low resolution (in the spatial and spectral domain), high level of noise, weak contrast, geometric deformations, as well as various types of uncertainty and inaccuracy. In addition, the insufficient sensitivity of the human eye perceives, according to Weber's law, only a 2% difference in brightness (the gray level of a standard monitor is approximately 0.04%), and this value significantly depends on the surrounding background, which reduces the ability to detect low-intensity objects of interest based on visual analysis [2].

It is important to note that appropriately assessing the quality of the image is quite a difficult task, since the characteristics of the image as a whole and in local areas (areas of interest) can differ significantly. This makes it difficult to automatically calculate the quantitative value of the overall quality assessment of both the raw data and the processing result. In practice, expert evaluations are usually used to determine the quality of images.

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine

EMAIL: akhmlu1@gmail.com (L. Akhmetshina); for___students@ukr.net (A. Yegorov); stas.mitrofanov1337@gmail.com (S. Mitrofanov)

ORCID: 0000-0002-5802-0907 (L. Akhmetshina); 0000-0002-7558-785X (A. Yegorov); 0009-0005-5491-7740 (S. Mitrofanov)



Digital processing of medical images allows to significantly improve their quality, in particular, contrast and resolution [3, 4]. Due to the huge variety of types of images, there are currently no universal methods that provide a guaranteed result of solving this problem [5].

X-ray images, which are an important diagnostic tool for many diseases, are characterized by low intensity, uneven background, high level of noise, weak contrast, and poorly defined boundaries of structures, and are particularly difficult to analyze and choose an effective processing method [6].

2. Review of Literature

Approaches to the improvement of digital images are usually divided into two categories: processing methods in the spatial and frequency domains [7]. The concept of "spatial transformations" combines approaches that are based on the direct change of brightness values of pixels of raster images [8-10].

Frequency methods, in particular, change not the image, but the form of its representation, converting the output signal into its components of different frequencies and amplitudes. In this form, it is much easier to perform filtering or amplification of individual components of the signal, to highlight important parameters whose detection by other methods is less effective or impossible [11]. Such algorithms are quite effective from the perspective of denoising signal, do not require a priori information, which is often absent in practice. This mathematical tool is widely used in medical imaging in the formation of CT, MRI and ultrasound images of human anatomy [12].

These medical images, besides the shortcomings caused by weak lighting exposure, noise, low contrast, etc., include uncertainty and fuzziness, which complicates the extraction of necessary information. Thus, the task of improving their quality is an essential task for carrying out a proper diagnosis. In [13], various fuzzy logic methods are considered for this purpose, which, like the frequency domain, allows obtaining an additional feature space for analysis.

Since the 1980s, fuzzy set theory [14] has been used for image processing, which has the ability to model the problems associated with uncertainty and inaccuracy, which are always present in digital images. Their presence is determined both by the features of the physical processes of image formation systems and by the stage of creating a digital image [15].

The main difference between fuzzy methods and other processing methodologies is that the input data (gray levels, histograms, features, etc.) are transformed into a fuzzy domain. An $M \times N$ image G with L L gray levels can be represented as an array of fuzzy sets with respect to the analyzed property with the value of the membership function μ_{mn}, which varies in the interval $[0,1]$ for each pixel x_{mn}:

$$G = \bigcup_{m=1}^{M} \bigcup_{n=1}^{N} \frac{\mu_{mn}}{x_{mn}}. \quad (1)$$

The transition to a fuzzy domain (fuzzification) can be interpreted as a specific type of input data encoding, which depends on both the goal and the characteristics of the output image. The use of fuzzy logic makes it possible to obtain new effective algorithms for processing digital images (fuzzy clustering, equalization, intensification, etc.) [17-19].

Based on the features and characteristics of different types of medical images, a combination of different algorithms is usually used to obtain a good processing result [18, 20, 21].

3. Problem statement

The paper is devoted to the description and experimental research of a new adaptive composite method for enhancing low-contrast images by increasing contrast and detail level, which combines frequency and fuzzy transformations and makes several transitions between different feature spaces.

4. Materials and Methods

For an image of size $M \times N$, which is described by a real two-dimensional discrete function $F(x, y)$ the discrete two-dimensional Fourier transform (2D DFT) provides the creation of a complex

two-dimensional function, which is defined in a frequency coordinate system (u,v) based on the expression:

$$F(u,v) = \frac{1}{M \cdot N} \cdot \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(x,y) e^{i \cdot 2 \cdot \pi \cdot \left(\frac{ux}{M} + \frac{vy}{N} \right)}. \quad (2)$$

The basis functions are sinusoidal and cosinusoidal waves with increasing frequencies. $F(0,0)$ represents the constant component of the image, which corresponds to the average brightness, and $F(M-1, N-1)$ represents the highest frequency [11].

The 2D DFT is a selective Fourier transform and therefore does not contain all frequencies that make up the image, but only a set of samples that are sufficient to describe the image in the spatial domain. The number of frequencies corresponds to the number of pixels in the image and allows access to the geometric characteristics of the image in the spatial domain.

The inverse two-dimensional discrete Fourier transform (2D IDFT) is given by:

$$F(x,y) = \sum_{u=o}^{M-1} \sum_{v=0}^{N-1} F(u,v) \cdot e^{i \cdot 2 \cdot \pi \cdot \left(\frac{ux}{M} + \frac{vy}{N} \right)}. \quad (3)$$

The ranges of changes in spatial $x = 0,1,2,...,M-1$, $y = 0,1,2,...,N-1$ and frequency coordinates $u = 0,1,2,...,M-1$, $v = 0,1,2,...,N-1$ are the same.

The Fourier transform always creates an output image with a complex number value:

$$F(u,v) = R(u,v) + i \cdot I(u,v), \quad (4)$$

where $R(u,v)$ and $I(u,v)$ are real and imaginary components of $F(u,v)$.

The main method of using this transform for image analysis and transformation is by computing and visualizing the spectrum.

$$| F(u,v) | = \sqrt{R^2(u,v) + I^2(u,v)}. \quad (5)$$

It is the amplitude of the Fourier transform that contains most of the image information in the spatial domain. The higher the value of $| F(u,v) |$, the brighter the point with coordinates (u,v). The bright center of the spectrum means that the original image contains mostly homogeneous areas, without differences in brightness. Bright periphery - many local differences in brightness.

In the phase spectrum which is calculated by the formula:

$$\phi(u,v) = \arctan\left(\frac{I(u,v)}{R(u,v)} \right), \quad (6)$$

the value of each point determines the shift from the components of the image. At zero phase, all sinusoids are centered in the same place, resulting in a symmetrical image, the structure of which has no real correlation with the original image. The phase reflects to a certain extent the location of the harmonic components in a spatial domain.

The importance of phase is particularly evident in certain specific areas, such as optical metrology, materials science, adaptive optics, X-ray diffraction optics, electron microscopy, and biomedical imaging. The most interesting samples relate to phase objects with very low absorption but with a non-uniform spatial distribution of their refractive index or thickness. This leads to small variations in amplitude but significant variations in phase components. [11].

Figure 1a shows a typical X-ray image and its histogram. The weak contrast is due to the limited range of reproducible brightness. An optimally visually perceptible image has a brightness distribution close to normal and a wide dynamic range, and is interpreted as a high-contrast image when the levels are distributed close to uniform. A classical low-contrast image has a narrow histogram located near the center of the brightness range. In this case, standard methods such as histogram equalization, linear contrast enhancement, gamma correction, and gradient mapping provide good results [1, 2].

Unlike low-contrast images, the characteristics of the image shown in Figure 1a, which we call "weakly-contrasted," can be formulated as follows:

- a full (or almost full) range of gray levels with significant dark and light areas;
- a multimodal histogram;

- a smooth transition of brightness in "regions of interest" (below the threshold of visual perception) and blurred boundaries between them;
- intensity levels are changed significantly in different areas;
- anomalies are not always obvious (and may be absent) and are often compared to the level of noise;
- low signal-to-noise ratio and complex structured background;
- significantly different in quality (for the same object and recording method), depending on the conditions of formation (quality of film, equipment, recording time), making the analysis task particularly difficult.

Figure 1b shows the result of applying adaptive histogram equalization (window size 8x8 pixels, uniform transformation function for the window), while Figure 1c shows the frequency domain transformation based on the expression.

$$F'(u,v) = \left|F(u,v)\right|^{r} \cdot e^{i\cdot\phi(u,v)}, \tag{7}$$

where r is the transformation parameter (which significantly affects the result and requires tuning; the value used to obtain the image in Figure 1c was $r = 0.8$). 2D IDFT (3) is applied to the matrix $F'(u,v)$, as a result, the resulting image is formed, which is scaled to the range of $[0,1]$.

Figure 1d shows the result of processing with the fuzzy intensification operator [14]. The transition to the fuzzy domain is based on the values $I_{\max}$, I_{mid} are the maximum and average values of the input image, respectively, and the parameters $F_e = 2$ and F_d, which is calculated using the formula:

$$F_d = \frac{I_{\max} - I_{mid}}{0.5^{\frac{-1}{Fe}} - 1}, \tag{8}$$

The membership function for I_{mn} is calculated according to the formula:

$$\mu_{mn} = \left(1 + \frac{I_{\max} - I_{mn}}{F_d}\right)^{-F_e}, \tag{9}$$

and its modification is performed as follows:

$$\mu'_{mn} = \begin{cases} 2\cdot\mu^2{}_{mn}, 0 \le \mu_{mn} \le 0.5 \\ 1 - 2\cdot(1-\mu_{mn})^2, 0.5 < \mu_{mn} \le 1 \end{cases}, \tag{10}$$

The final result of processing is formed according to the following formula:

$$I_1 = I_{\max} - F_d \cdot \left(\left(\mu'_{mn}\right)^{-1/F_e} - 1\right), \tag{11}$$

The image processing results in Figure 1a indicate that, all methods provide a similar degree of improvement in contrast. However, there is no significant improvement in terms of visual analysis, such as object detection, boundary highlighting, and detailing of individual structures.

The algorithm proposed in this work can be formulated as follows:

1. 2D DFT (2) is applied to an image I, scaled to the range of $[0,1]$. This step allows to make a transition to a new feature space;
2. frequency domain transformation based on formula (7) with a coefficient of $r = 0.8$ and obtaining the matrix F_1';
3. 2D IDFT (3) is applied to matrix F_1' (returning to the original feature space), with results scaled to the range of $[0,1]$, afterwards adaptive histogram equalization is applied, which results in an image I_2. In the proposed algorithm, adaptive histogram equalization where used with window size 8x8 pixels, and uniform transformation function for the window;
4. formation of I_3, according to the formula:

$$I_3 = I_2 - \overline{I_2}, \tag{12}$$

where $\overline{I_2}$ is mean of gray levels of the I_2;

5. contrast enhancement for the image I_3, using a method based on the application of the fuzzy intensification operator, resulting in the formation of the image I_4. This step also leads to transition to a new feature space;
6. 2D DFT (2) is applied to the image I_4 and obtaining the function $F_5(u,v)$. This transformation makes transition to a new features space again;

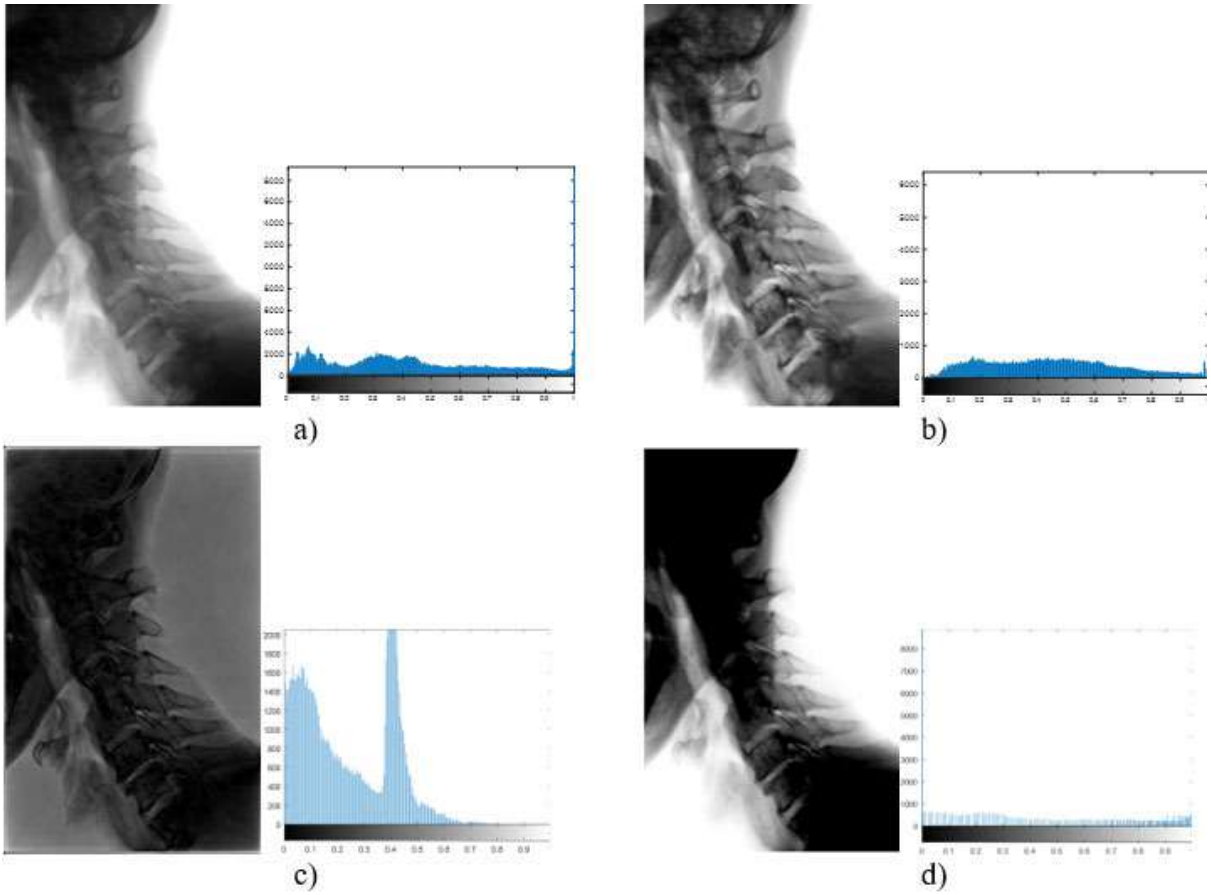


Figure 1: X-ray image processing: a – input image (459x290); results of processing by different methods: b – adaptive histogram equalization; c – Fourier transform; d – fuzzy intensification operator

7. formation of the matrix F_6 according to the following formula:

$$F_6(u,v) = \left(|F_5(u,v)|^{r_1} - |F_5(u,v)|^{r_2} \right) \cdot e^{i \cdot \phi(u,v)}. \quad (13)$$

It should be noted that the coefficients and require manual adjustment as they have a significant impact on the detailing of the result and may differ for different input images. This leads to additional time costs, so we also proposed formulas for the automatic calculation of matrices for these coefficients

$$r_1(u,v) = r_c + I_4^{'} + \frac{I_4^{'} + F_5^{'}(u,v)}{2 \cdot 10 \cdot \left(1 + I_4^{'}\right)}, \quad (14)$$

$$r_2(u,v) = I_4^{'} + \frac{F_5^{'}(u,v)}{10}, \quad (15)$$

where $r_c = 0.5$ and $I_4^{'}$ is defined by the next formula:

$$I_4^{'} = \left(0.5 + \left(\frac{I_4^{\max} + I_4^{\min}}{2} + \overline{I_4} \right) \right) / 2, \quad (16)$$

where $\overline{I_4}$ is mean of gray levels of the I_4; $I_4^{\min}$, $I_4^{\max}$ are the minimal and the maximal values of the I_4, correspondingly; $F_5'(u,v)=|F_5(u,v)|$ with following scaling of obtained matrix to the range of $[0,1]$. This method of calculation allows to control the level of detail by changing coefficient r_c in formula (14). Increasing this coefficient leads to decreasing level of detail of final image while decreasing r_c leads to increasing the level of details of resulting image;

8. 2D IDFT is applied to matrix F_6, resulting in the formation of image I_7 (returning to the original feature space), with results scaled to the range of $[0,1]$. After that, adaptive histogram equalization is used, which leads to the formation of the final result.

5. Results and Discussion

The described algorithm was used to process various X-ray images. Analysis of Figure 1a is used to calculate indicators of X-ray planimetry in patients with spinal cord and cervical spine injuries in the practice of medical and social expertise. Blurred boundaries of objects of interest and a lack of detail in their structure complicate the solution of this task.

To obtain experimental results we used Matlab 2016. Source code for described methods was written in internal Matlab language (except for standard functions like fft2, ifft2, adapthisteq, mat2gray, etc.).

Figure 2 shows the results of processing the X-ray image in Figure 1a. The study of the effect of the coefficient r shows, that its variation allows for the detection of object structures and the adjustment of their level of detail. Experimental studies have shown that using the algorithm formula (7) at step 7 also allows for an increase in contrast and resolution, but increases the contribution of noise components. Figures 2a and 2b show the results of processing the image in Figure 1a using formula (7) at step 7 for different values of r, which demonstrate an increase in the level of detail and noise reduction as it decreases.

Figure 3 shows brightness graphs of the line (its first 100 pixels) in the center of the image in Figure 1a when using formula (7) at step 7 of the proposed algorithm for values of $r=0.5$; $r=0.8$ and $r=1.2$. Based on the obtained graphs, it can be concluded that the value of parameter r in formula (7) affects the ratio of high-frequency and low-frequency components.

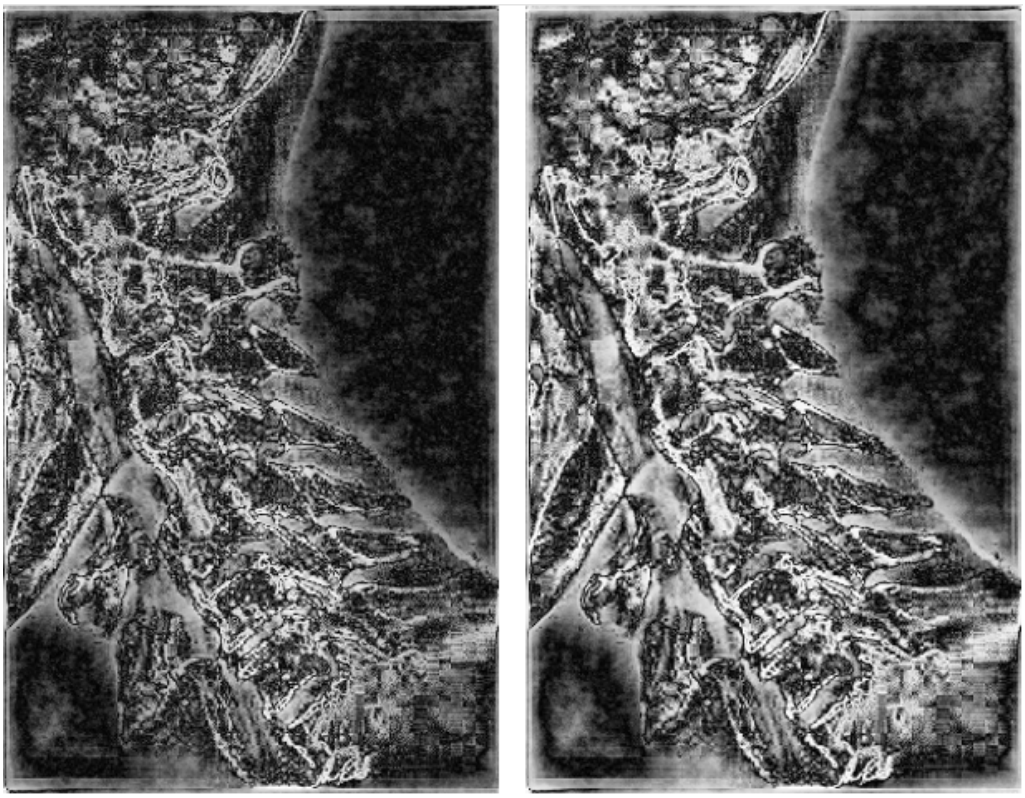
The use of formula (13) at step 7 of the proposed algorithm provides an adjustable increase in image detail by changing the parameters r_1 and r_2, with a decrease in the noise component. Figure 4a shows the result of the algorithm for $r_1=1.2$ and $r_2=0.7$, which show a significant improvement in visualization of the region of interest. Figure 4b presents the result of the proposed algorithm with automatic calculation of parameters r_1 and r_2 based on formulas (14), (15), which is very close to the manual selection of these parameters.

In processing many medical images, the interest is not so much in increasing the image contrast as in the redistribution of brightness levels in order to make small brightness variations more noticeable and distinguishable, which is important for visually highlighting objects of interest. In this case, the fact of their presence and location is usually unknown. In such a case, the assessment of such integral numerical characteristics as contrast or brightness level is uninformative.

In cases where the analysis area is defined (e.g., in monitoring the dynamics of disease progress), improving the visual distinguishability of the object of interest does not necessarily imply an improvement in the overall contrast level due to the increased contribution of the background component. Valid tracking of image quality changes can only be achieved by visually assessing the differences detected in the images.

One of the ways to evaluate the processing result is to analyze the brightness change graphs in the region of interest (Fig. 5). The presented brightness graphs of a portion (pixels from 40 to 140) of an arbitrary row (in this case, 220th from the top) of the input I and resulting I_7 images indicate both an increase in the dynamic range – $\Delta I_7 / \Delta I \approx 3$, and a redistribution of brightness levels in low-

contrast areas of interest. The usage of the classical method of adaptive histogram equalization (Fig. 1b) does not provide such an effect.



a b

Figure 2: Results of the proposed algorithm when using formula (7) at step 7 for different parameter values: a – $r = 0.7$; b – $r = 0.8$

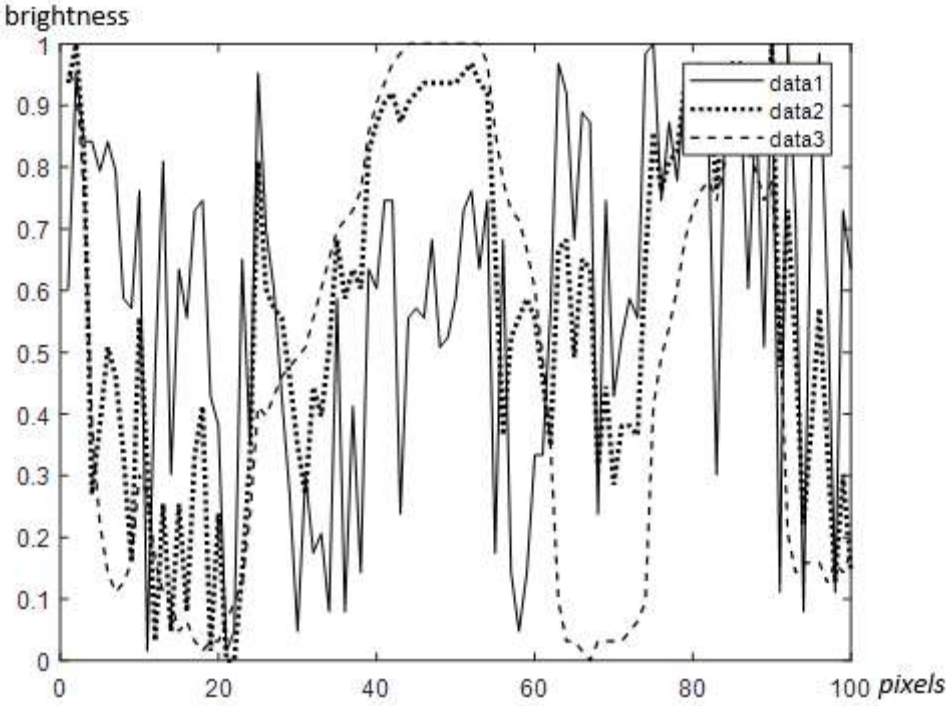
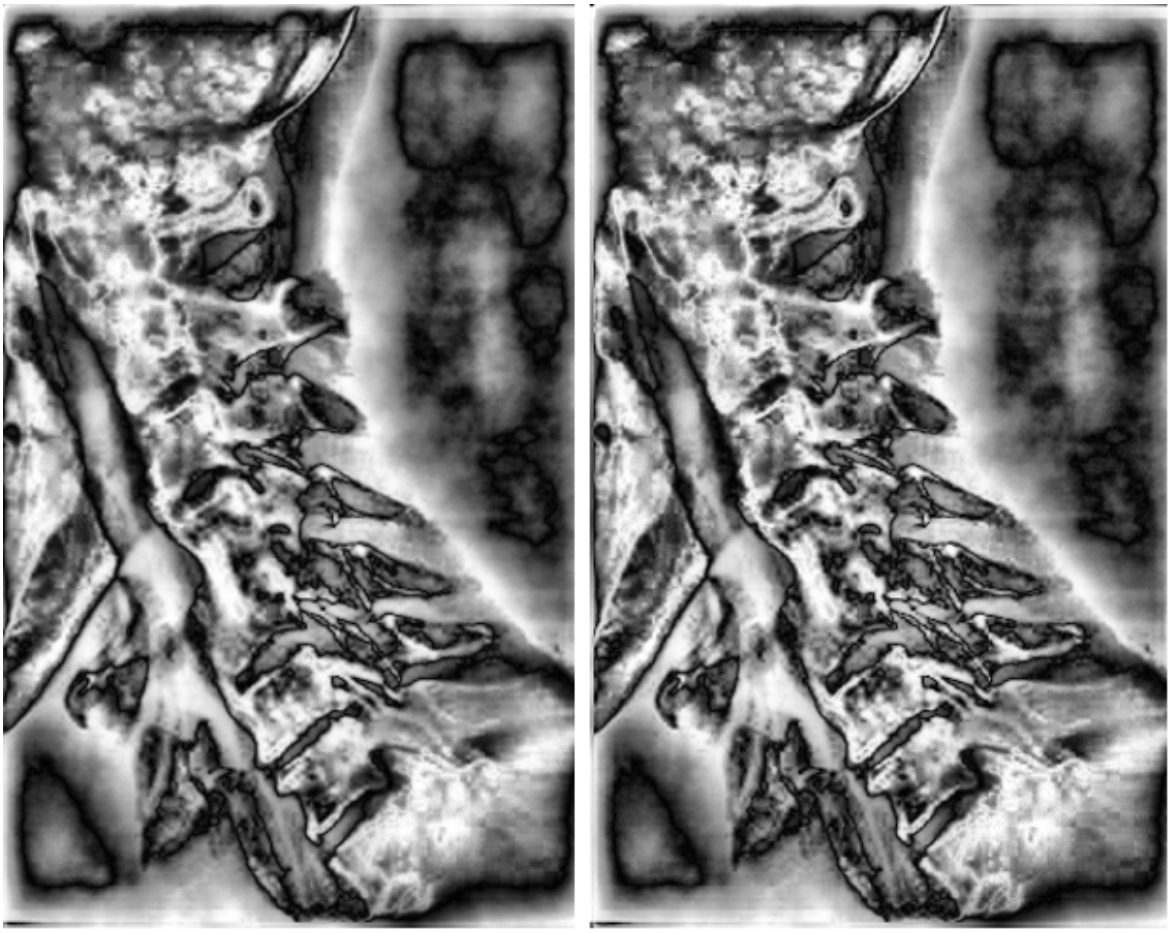


Figure 3: Changes in the brightness of a line in the center of the image for the values $r = 0.5$ (data1); $r = 0.8$ (data2); $r = 1.2$ (data3).



a b

Figure 4: The results of the proposed algorithm using formula (13) in step 7 for different parameter values are shown: a – $r_1 = 1.2$, $r_2 = 0.7$; b – for automatic calculation of r_1 and r_2 using formulas (14) and (15), respectively.

Figure 6a shows an X-ray image used for diagnosing inflammation (sinusitis), indicated by an arrow. The object of interest is not visible for direct visual analysis in the original image because of its location in an area of high brightness. The use of adaptive histogram equalization (Figure 6b) slightly improves the image but does not provide the necessary level of tissue structure detailing. The intensification operator (Figure 6c) worsens the original X-ray image.

The application of the proposed algorithm (Figure 6d-6f) significantly increases the image detailing in the area of low brightness values and the object of interest. In particular, the detection of the structure of the nasopharynx and oral cavity region should be noted, while the inability to visualize soft tissue is a significant drawback of X-ray research methods.

The values of r_1 and r_2 also affect the level of detail in the resulting image, as demonstrated by the comparison between Figure 6d ($r_1 = 0.95, r_2 = 0.7$) and Figure 6e ($r_1 = 1.2, r_2 = 0.7$). The improvement in the clarity of tissue structure delineation is most noticeable in the eye area. When using the automatic calculation of r_1 and r_2 based on formulas (14) and (15), the processed image (Figure 6f) is close to the best result obtained by manual selection of coefficients ($r_1 = 1.2, r_2 = 0.7$), but has a little higher level of detail due to adaptive nature of automatically calculated coefficients r_1 and r_2 . This fact confirms the effectiveness of the proposed method for automatic calculation of the method's parameters.

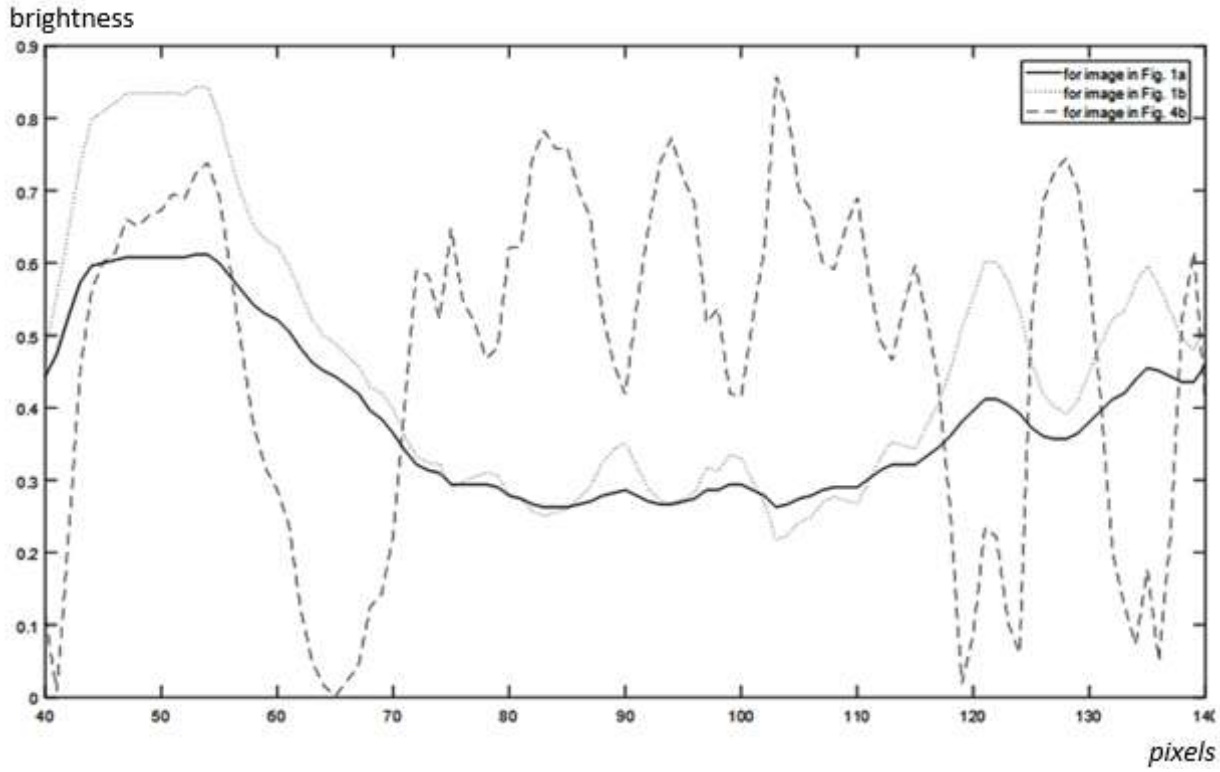


Figure 5: Changes in the brightness of a line number 220th (from the top) for its pixels from 40 to 140 for initial image (Fig. 1a) and processed images (Fig. 1b and Fig. 4b)

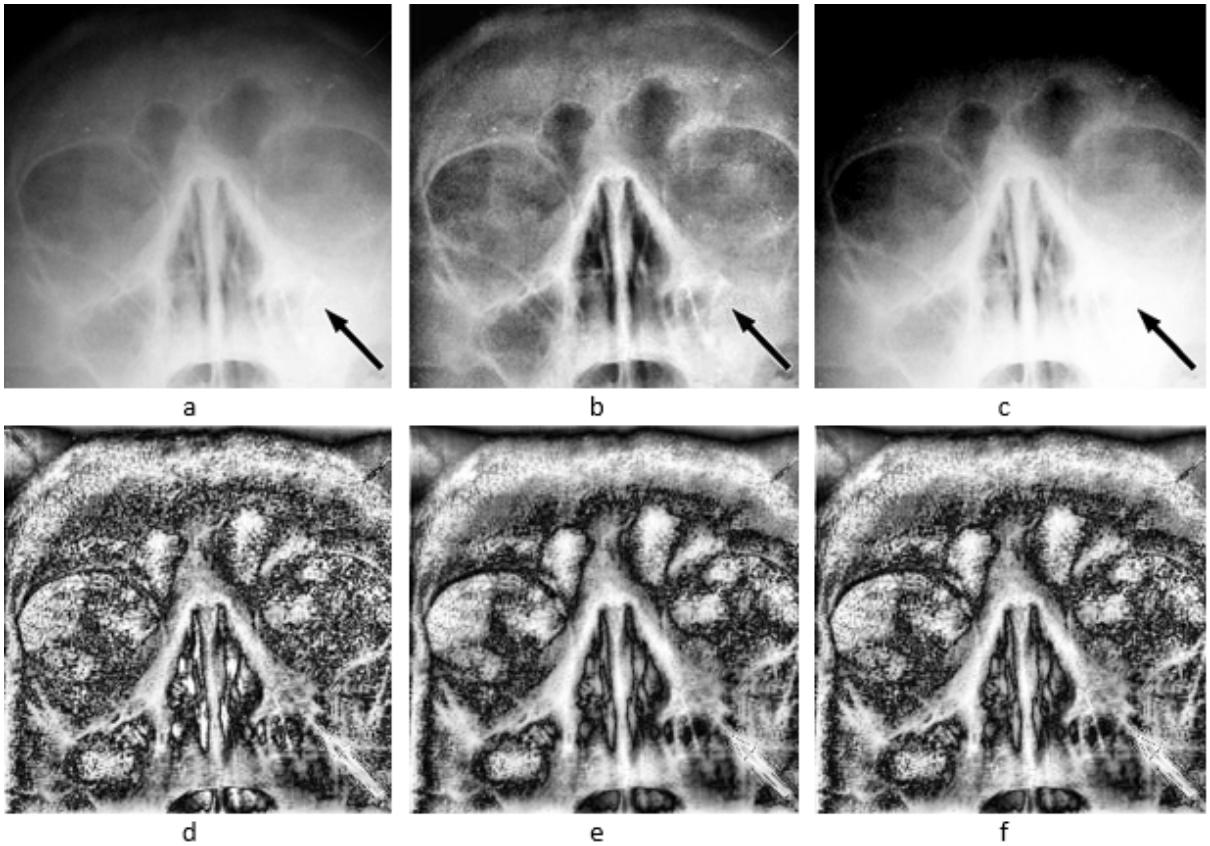


Figure 6: X-ray image of a human face: a – original image (266x272); b – adaptive histogram equalization; c – intensification operator; proposed algorithm using: d – $r_1 = 0.95$, $r_2 = 0.7$; e – $r_1 = 1.2$, $r_2 = 0.7$; f – automatic calculation of r_1, r_2 using formulas (14) and (15), respectively.

6. Conclusions

On the basis of analysis of the experimental results obtained, the following conclusions are made:

- the application of a composite adaptive algorithm for processing grayscale images, which includes the transformation of the frequency response of the 2D DFT and the method of fuzzy intensification;
- the algorithm has parameters that can be adjusted to control the level of detail in the object structure. We proposed an automatic calculation of these parameters (r_1, r_2), which reduces the time required for their selection and allows achieving a high level of detail without excessive detail effect;
- researches have shown the effectiveness of this algorithm for low-contrast images of different physical nature;
- the usage of various fuzzy transformation methods represents a promising direction for further research.

7. References

[1] D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, 2nd. ed., Prentice Hall, 2011.

[2] R. Gonzalez, R. E. Woods, Digital Image Processing, 4th ed., Pearson, 2017.

[3] C. H. Chen, P. S. P. Wang, Handbook of pattern recognition & computer vision, 3rd ed., World Scientific Publishing Co. Pte. Ltd., 2009. doi.org/10.1142/7297.

[4] J M. Bentourkia, Fourier Analysis and Medical Image Filtering, Cambridge Scholars Publishing, 2022.

[5] S. Angenent, E. Pichon, A. Tannenbaum, Mathematical methods in medical image processing, Bulletin (New Series) of the American Mathematical Society, volume 43(3), pp. 365–396. doi: 10.1090/s0273-0979-06-01104-9.

[6] A. M. Ikhsan, A. Hussain, M. A. Zulkifley, N. Tahi, An analysis of x-ray image enhancement methods for vertebral bone segmentation Conference: 2014 IEEE 10th International Colloquium on Signal Processing & its Applications (CSPA), pp. 208-211. doi: 10.1109/CSPA.2014.6805749.

[7] W.K. Pratt, Digital Image Processing, John Wiley and Sons Inc., New York, NY, 2001.

[8] E. Pichon, A. Tannenbaum, Mathematical methods in medical image processing sigurd angenent, Bulletin (new series) of the American Mathematical Society, volume 43(3), 2006, pp. 365–396. doi: 10.1090/s0273-0979-06-01104-9.

[9] A. Mustapha, A. Hussain, S. A. Samad, A new approach for noise reduction in spine radiograph images using a non-linear contrast adjustment scheme based adaptive factor, Sci. Res. Essays, volume 6(20), 2011, pp. 4246–4258.

[10] R. Kaur, S. Guru, G. Sahib, Feature extraction and principal component analysis for lung cancer detection in ct scan images. Int. J. Adv. Res. Comput. Sci. Softw. Eng., volume 3(3), 2013, pp. 187–190.

[11] A. McAndrew, An Introduction to Digital Image Processing with Matlab, 2004.

[12] J. Luce, J. Gray, Mark A. Hoggarth, Jeffery Lin and others, Medical Image Registration Using the Fourier Transform, International Journal of Medical Physics, Clinical Engineering and Radiation Oncology, volume 3(1). doi: 10.4236/ijmpcero.2014.31008.

[13] J. Krayla, S. Kumar, U. K. Acharya and others, Comparative Analysis of Fuzzy Logic-based Image Enhancement Techniques for MRI Brain Images in: R. Asokan, D.P. Ruiz, Z.A. Baig, S. Piramuthu (Eds.) Smart Data Intelligence. Algorithms for Intelligent Systems, Springer, Singapore, 2022, pp. 447–458. doi:10.1007/978-981-19-3311-0_38.

[14] I. Bloch, Fuzzy sets for image processing and understanding, volume 281 of Fuzzy Sets and Systems, 2015, pp. 280-291. doi:10.1016/j.fss.2015.06.017.

[15] H.R. Tizhoosh, H. HauBecker, Fuzzy Image Processing: An Overview, in: B. Jähne, H. HauBecker. and P. GeiBler, (Eds.), Handbook on Computer Vision and Applications, Academic Press, Boston, volume 1, 2000, pp. 683-727.
[16] Z. Chi, H. Yan, T. Pham, Fuzzy algorithms: With Applications to Image Processing and Pattern Recognition, Singapore, New Jersey, London, Hong Kong, Word Scientific, 1998.
[17] A. Nirmala. Medical Image Denoising by Nonlocal Means with Level Set Based Fuzzy Segmentation. Indian Journal of Science and Technology, volume 10(36), 2017, pp. 1-11. doi: 10.17485/ijst/2017/v10i36/112402.
[18] A. Yegorov, L. Akhmetshina, Optimizatsiya yarkosti izobrazheniy na osnove neyro-fazzi tekhnologiy, Lambert, 2015.
[19] T. Chaira, A novel intuitionistic fuzzy C means clustering algorithm and its application to medical images, Applied Soft Computing, volume 11(2), 2011, pp. 1711–1717. doi: 10.1016/j.asoc.2010.05.005.
[20] L. Akhmetshina. A. Egorov, Low-Contrast Image Segmentation by using of the Type-2 Fuzzy Clustering Based on the Membership Function Statistical Characteristics. International Scientific Conference Lecture Notes in Computational Intelligence and Decision Making. AISC, 2020, - v. 1020. pp. 689-700.
[21] H. S. S. Ahmed, M. J. Nordin, Improving Diagnostic Viewing of Medical Images using Enhancement Algorithms, Journal of Computer Science, volume 7(12), 2011, pp. 1831-1838.