

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ПРОГНОЗУВАННЯ ЗАХВОРЮВАНОСТІ НА ДІАБЕТ ЗА ДОПОМОГОЮ
ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ
RESEARCH AND SOFTWARE IMPLEMENTATION OF THE METHODS FOR
PREDICTING THE INCIDENCE OF DIABETES
USING A CONVOLUTIONAL NEURAL NETWORK

Виконав: студент(ка) 2 курсу, групи КНТ-120м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Білевський А.Г.

(прізвище та ініціали)

Керівник Гофман Є.О.

(прізвище та ініціали)

Рецензент Коцур М.І.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ

Кафедра програмних засобів

Ступінь вищої освіти магістр

Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Білевського Андрія Геннадійовича
 (прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Research and Software Implementation of the Methods for Predicting the Incidence of Diabetes Using a Convolutional Neural Network

керівник проєкту (роботи) Гофман Євгеній Олександрович, к.т.н., доцент,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “5” листопада 2021 року № 435

2. Строк подання студентом проєкту (роботи) грудень 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Гофман Є.О., доцент		
Нормоконтроль	Липовець М.В., асистент		

7. Дата видачі завдання “4” вересня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз та дослідження предметної області.	2-3 тижні	Розділи 1, 2
3	Розробка структури програми.	4-5 тижні	Розділ 2
4	Розробка програми.	7-10 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми.		Розділ 4
		11-12 тижні	
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування.		Додатки
		13-15 тижні	
7	Захист роботи.	16 тиждень	

Студент(ка)

(підпис) Білевський А.Г.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Гофман Є.О.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
103 с., 30 рис., 5 табл., 4 дод., 36 джерел.

ДІАБЕТ, ЗГОРТКОВІ МЕРЕЖІ, НЕЙРОННІ МЕРЕЖІ,
ПРОГНОЗУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес діагностування на основі медичних даних.

Предмет дослідження – методи та програмні засоби для діагностування цукрового діабету.

Метою роботи є дослідження та програмна реалізація програмного забезпечення для прогнозування захворюваності на діабет.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки Eclipse, моделі на основі нейронних мереж та методи їх синтезу.

Результати. Розроблено модифіковану модель CNN та програмні засоби для прогнозування захворюваності на діабет.

Практична цінність роботи полягає у розробці програмного забезпечення для прогнозування захворюваності на діабет.

Висновки. Розроблено модифіковану модель CNN та програмні засоби для прогнозування захворюваності на діабет.

Галузь використання – медичне діагностування, прогнозування за допомогою згорткових нейронних мереж.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 103 pages, 30 figures, 5 tables, 4 appendixes, 36 sources.

DIABETES, CONVOLUTIONAL NETWORKS, NEURAL NETWORKS, FORECASTING, SOFTWARE.

The object of research is the process of diagnosis based on medical data..

The subject of the research is methods and software for diagnosing diabetes.

The purpose of this work is to research and software implementation of software for predicting the incidence of diabetes.

Materials, methods and technical tools: Python programming language, Eclipse development environment, models based on neural networks and methods of their synthesis.

Results. The modified CNN model and a program for predicting the incidence of diabetes have been developed.

The practical value of the work lies in the development of software for predicting the incidence of diabetes.

Conclusions. A modified CNN model and a program for predicting the incidence of diabetes have been developed.

Scope of use – medical diagnosis, prediction using convolutional neural networks.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Поняття цукрового діабету та принципи терапії хворих на діабет.....	11
1.2 Огляд існуючих програмних продуктів, що використовуються для полегшення життя пацієнтів, хворих на цукровий діабет	14
1.3 Аналіз діагностичних моделей для прогнозування на основі медичних даних.....	25
1.4 Висновки за розділом 1	33
2 Матеріали і методи.....	35
2.1 Вибір та обґрунтування технологій розробки для створення програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.....	35
2.2 Вибір середовища для розробки програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі	40
2.3 Структурна схема програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі	43
2.4 Висновки за розділом 2	45
3 Опис програми.....	46
3.1 Розробка модифікованого методу синтезу моделі на основі згорткової нейронної мережі	46
3.2 Функціонування програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі	50
3.3 Основні рішення по реалізації програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі	52
3.4 Висновки за розділом 3	57

4 Експлуатація, тестування та експериментальне дослідження програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі	59
4.1 Призначення й умови застосування програми.....	59
4.2 Характеристики програми.....	60
4.3 Інструкція по експлуатації програми програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі	61
4.3.1 Звернення до програми.....	61
4.3.2 Вхідні й вихідні дані	61
4.3.3 Повідомлення	61
4.4 Тестування та результати функціонування програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі	62
4.5 Висновки за розділом 4	64
Висновки	66
Перелік джерел посилання	69
Додаток А Технічне завдання	73
Додаток Б Опис програми	77
Додаток В Текст програми	81
Додаток Д Слайди презентації.....	96

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CART – classification and regression tree;

CNN – convolutional neural networks;

DCNN – deep convolutional neural networks;

DNN – deep neural networks;

ГА – генетичний алгоритм;

ЗНМ – згорткова нейронна мережа;

ПЗ – програмне забезпечення;

ПО – предметна область;

ПП – програмний продукт;

ТЗ – технічне завдання;

ЦД – цукровий діабет.

ВСТУП

Цукровий діабет у наш час є глобальною проблемою та хворобою ендокринної системи, рівень розповсюдження якої зростає у всьому світі. Відповідно даних Всесвітньої організації охорони здоров'я цукровий діабет є серйозним захворюванням, яке призводить до порушення обміну речовин і поразці всіх органів і систем організму. Це веде до інвалідності та передчасної смертності [1]-[2]. Цукровий діабет часто є причиною сліпоты, іншим недугом, що виникає як наслідок діабету, є ниркова недостатність, крім того, така хвороба збільшує ризик судинних проблем, зокрема інфарктів та інсультів у декілька разів, у тяжких випадках можливі ампутації нижніх кінцівок, а зменшення тривалості життя на декілька років. Такі тяжкі наслідки роблять діабет суттєвою медико-соціальною проблемою, що змушує лікарів-ендокринологів та інших фахівців шукати способи ефективного лікування та профілактики цього захворювання [1]-[6].

Тому доцільною є розробка методів та програмних засобів медичного діагностування, які будуть придатними для виконання поставленої задачі прогнозування захворюваності на цукровий діабет.

Метою роботи є дослідження та програмна реалізація програмного забезпечення для прогнозування захворюваності на діабет.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- проаналізувати методи та засоби медичного діагностування;
- здійснити вибір засобів створення програмного забезпечення для прогнозування захворюваності на діабет;
- розробити модифікований метод синтезу моделі на основі згорткової нейронної мережі;
- розробити програмне забезпечення прогнозування захворюваності на діабет;
- здійснити тестування розробленого програмного забезпечення прогнозування захворюваності на діабет.

Об'єкт дослідження – процес діагностування на основі медичних даних.

Предмет дослідження – методи та програмні засоби для діагностування цукрового діабету.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття цукрового діабету та принципи терапії хворих на діабет

Цукровий діабет 1 типу, званий ще інсулінозалежний діабет, виявляється стабільно високим рівнем цукру в крові. Для інсулінозалежного діабету типово руйнування клітин острівців Лангерганса підшлункової залози, які синтезують інсулін, що призводить до зниження або повного припинення продукції інсуліну [1]-[6].

Основними причинами розвитку цукрового діабету 1 типу вважаються гострі вірусні інфекції, у тому числі грип, епідемічний паротит, а також захворювання, що супроводжуються зниженням синтезу і секреції інсуліну в організмі. Існує думка про виникнення діабету 1 типу в результаті гострого емоційного стресу. Значення генетичної схильності до розвитку захворювання остаточно не доведено [1]-[6].

Цукровий діабет 1 типу можливий в будь-якому віці, але найчастіше він розвивається у дітей, підлітків і дорослих людей не старше 30 років [1]-[6].

Оскільки підшлункова залоза не може самостійно виробляти інсулін, хворим необхідно щоденне введення інсуліну шляхом ін'єкцій. Крім розвитку гіперглікемії, при цукровому діабеті 1 типу з'являється сухість у роті, що призводить до появи інтенсивної постійної спраги, на рівні підвищеного апетиту спостерігається зниження маси тіла, часте сечовипускання [1]-[6].

Інсулінозалежний цукровий діабет або діабет типу 1 є аутоімунним захворюванням. Хвороби, що відносяться до групи аутоімунних, не піддаються лікуванню. При цьому захворюванні власні клітини підшлункової залози організм сприймає як чужорідні і знищує їх, що призводить до руйнування тканини залози, тому вилікувати цукровий діабет 1 типу проблематично [1]-[6].

Життя хворого при цукровому діабеті першого типу залежить від замісної терапії, що проводиться введенням інсуліну. Лікування діабету 1 типу

проводиться інсуліном, який може застосовуватися тільки у вигляді ін'єкцій, так як це білкові речовини, що при прийомі всередину відразу руйнується під дією травних ферментів шлункового соку [1]-[6].

Препарат і доза інсуліну підбираються кожного пацієнта індивідуально з урахуванням особливостей перебігу захворювання в даному випадку. Застосуванням інсулінотерапії можна досягти тривалої компенсації захворювання з відсутністю прогресу і розвитку ускладнень.

Крім застосування інсуліну, лікування цукрового діабету 1 типу вимагає дотримання спеціальної дієти з обмеженням споживання вуглеводів, виконання адекватних фізичних навантажень [1]-[6].

У дітей зазвичай розвивається інсулінозалежний цукровий діабет. Причинами розвитку захворювання зазвичай бувають гострі вірусні захворювання. Поява діабету 2 типу може провокувати штучне вигодовування. Молочні суміші містять коровій білок, що наносить шкідливу дію на тканину підшлункової залози дитини [1]-[6].

Діабет у дітей і підлітків характеризується важким перебігом і, як правило, гострим початком захворювання. Діабет типу 1 у дітей виявляється появою спраги, схуднення, виділення великої кількості сечі, загальною слабкістю, сухістю шкіри. Від часу появи перших симптомів захворювання до розвитку важкого стану і значних порушень обміну речовин зазвичай триває від двох до чотирьох років [1]-[6].

Дітям, які страждають на цукровий діабет, потрібне обов'язкове лікування інсуліном, дієта і постійний лікарський контроль [1]-[6].

Внаслідок того що цукровий діабет I типу інсулінозалежний, введення інсуліну є єдино можливим засобом лікування даної категорії хворих. Не існує альтернативи підшкірної замісної терапії інсуліном, хоча вона і являє собою лише імітацію фізіологічної дії інсуліну. У нормальних умовах інсулін надходить у систему ворітної вени, далі - в печінку, де наполовину інактивується, решта опиняється на периферії. Все це відбувається настільки швидко, що рівень глікемії може підтримуватися в досить вузьких межах

навіть після їжі. Інший шлях спостерігається у інсуліну, введеного під шкіру: він із запізненням потрапляє у кров і тим більше в печінку, після чого концентрація інсуліну в крові тривалий час залишається не фізіологічно підвищеної [1]-[6]. Це недосконалість підшкірної інсулінотерапії і пояснює той факт, що хворі на діабет змушені проводити самоконтроль обміну речовин, облік свого харчування, фізичних навантажень та інших факторів, що впливають на рівень глікемії. Але сучасна стратегія і тактика інсулінотерапії дозволяють зробити образ життя хворих на цукровий діабет I типу дуже близьким до нормального. Здійснити це можна тільки шляхом навчання хворих на цукровий діабет [1]-[6].

Для контролю цукру у крові використовуються препарати короткої та довгої дії [1]-[6].

Хумалог є аналогом інсуліну людини, відрізняється від нього зворотного послідовністю пролінового та лізинового амінокислотних залишків в положеннях 28 і 29 В-ланцюга інсуліну [1]-[6].

Особливістю дії препарату є більш швидкий початок і закінчення ефекту в порівнянні з усіма препаратами інсуліну короткої дії, які застосовують в даний час. Прискорення ефекту обумовлено прискоренням абсорбції (всмоктування) з підшкірного депо за рахунок збереження в розчині мономерної структури молекул Хумалог [1]-[6]. Початок дії – через 15 хв після підшкірного введення. Максимум дії – між 0,5 і 2,5 год. Тривалість дії – 3-4 год [1]-[6].

Хумулін НПХ (Humulin NPH): препарат інсуліну, ідентичного людському, середньої тривалості дії. Початок дії препарату – через 1 годину після введення. Максимум дії – між 2-м і 8-м часом. Тривалість дії – від 18 до 20 годин [1]-[6].

Таким чином, цукровий діабет – це захворювання, під час якого розвиваються ускладнення різних типів та різних мір. Щоб підтримувати стан хворого – необхідно бути постійно під наглядом, дотримуватись основних правил та принципів. Нажаль, під наглядом лікаря знаходитись постійно,

просто, не можливо, тому необхідно створити програмне забезпечення (ПЗ), яке допоможе хворим на цукровий діабет слідкувати за собою у той період часу, коли лікарів немає поруч.

1.2 Огляд існуючих програмних продуктів, що використовуються для полегшення життя пацієнтів, хворих на цукровий діабет

Проаналізуємо існуючі програмних продуктів, що використовуються для полегшення життя пацієнтів, хворих на цукровий діабет.

Програма Diabetes:M (рисунок 1.1) є мобільним застосунком для роботи під керуванням систем Android, Android Wear, iOS, Apple Watch [7]-[9].

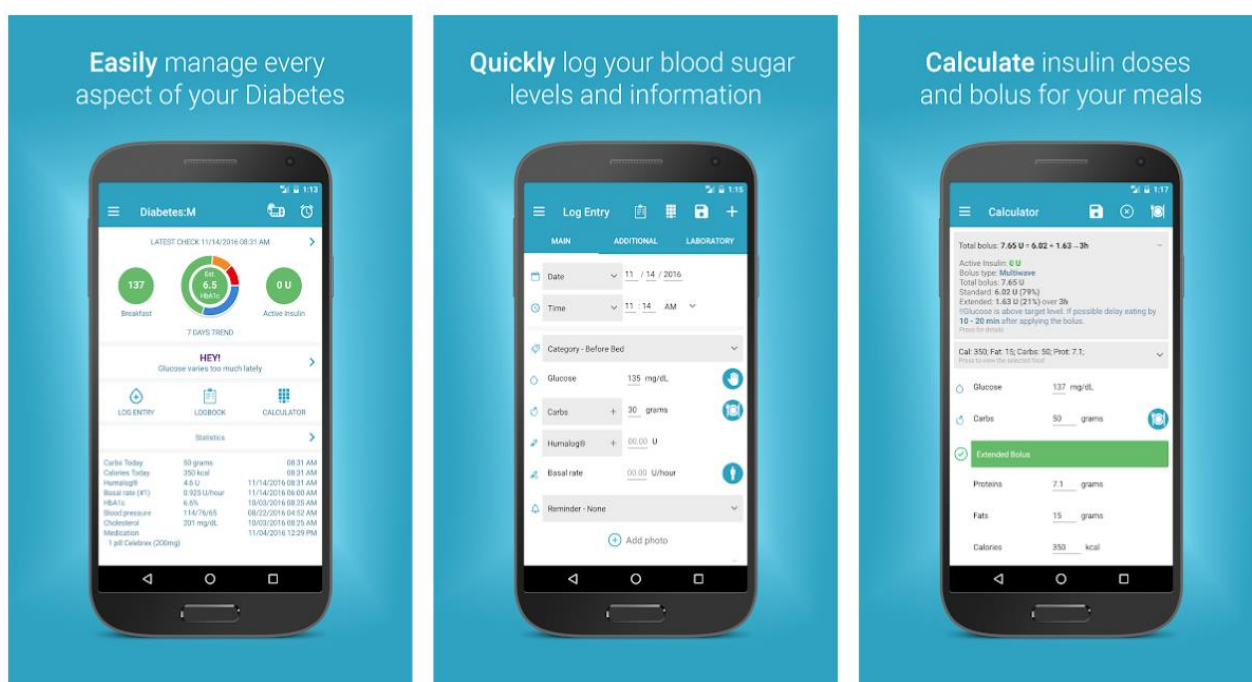


Рисунок 1.1 – Програма Diabetes:M

Застосунок має всі необхідні інструменти для діабетиків. Журнал контролю глюкози, інсуліну, харчування, прийому медикаментів, пам'ятки ін'єкцій, нотатки, категорії, нагадування та багато іншого [7]-[9].

На додаток до Diabetes:M є вбудована база їжі з можливістю додавання своїх продуктів. У ній можна відстежувати порції та кількість їжі.

Є калькулятор вуглеводів та доз інсуліну. За допомогою докладних звітів можна відстежувати всі параметри, доступні в програмі. Є можливість імпорту звітів для лікаря. У платній версії розблокується синхронізація між пристроями [7]-[9].

Перевагами програмного забезпечення Diabetes:M є такі:

- можливість використання програми Diabetes:M як мобільний застосунок;
- наявність вбудованої бази їжі з можливістю додавання своїх продуктів, де можна відстежувати порції та кількість їжі;
- наявність калькулятора вуглеводів та доз інсуліну;
- допомогою докладних звітів можна відстежувати різні параметри;
- можливість створення звітів для лікаря.

До недоліків програми варто віднести відсутність можливості прогнозування стану пацієнта, хворого на діабет, з урахуванням поточної медичної інформації та попередніх даних. Крім того, програмне забезпечення Diabetes:M не є безоплатним. Також у цьому програмному забезпеченні немає можливості ввести кількість спожитих хлібних одиниць чи кілокалорій у певний період прийому їжі, щоб перевірити, чи пацієнт не скоштував їжі більше за свою норму.

Програма DiaMeter (рисунок 1.2) може працювати під керуванням систем Android, iOS.

У DiaMeter можна легко відстежувати рівень цукру, кількість з'їдених хлібних одиниць, кількість ін'єкцій короткого та продовженого інсуліну. Додатково можна записувати дані про самопочуття та інше за допомогою нотаток. На основі наданих даних програма відображає дані у вигляді зручних та зрозумілих графіків. Є синхронізація даних між пристроями та корисні статті для діабетиків. У платній версії з'являється експорт даних та аналізи цукру за тиждень та місяць [7]-[9].

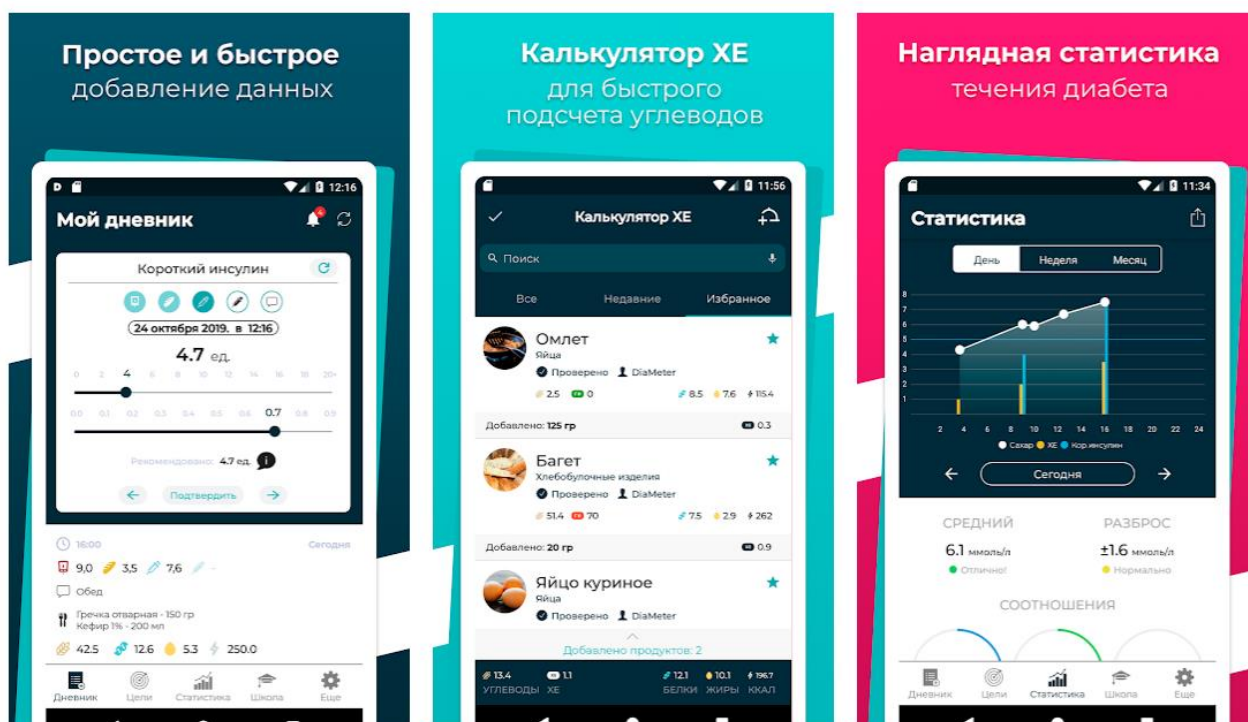


Рисунок 1.2 – Програмне забезпечення DiaMeter

Перевагами програмного забезпечення DiaMeter є такі:

- можливість відстеження рівня цукру, кількості з'їдених хлібних одиниць;
- наявність функцій відстеження кількості ін'єкцій короткого та продовженого інсуліну;
- можливість зберігання та аналізу даних про самопочуття та інше за допомогою нотаток;
- відображення даних у вигляді зручних та зрозумілих графіків;
- можливість синхронізації даних між пристроями;
- наявність корисних статей для діабетиків.

До недоліків програми варто віднести відсутність можливості прогнозування стану пацієнта, хворого на діабет, з урахуванням поточної медичної інформації та попередніх даних. Крім того, програмне забезпечення DiaMeter не є безоплатним. Відсутність вбудованої бази їжі з можливістю додавання своїх продуктів, де можна відстежувати порції та кількість їжі.

Програмне забезпечення ДіабетСтудия (рисунок 1.3) є мобільним застосунком для роботи під керуванням систем Android, iOS.

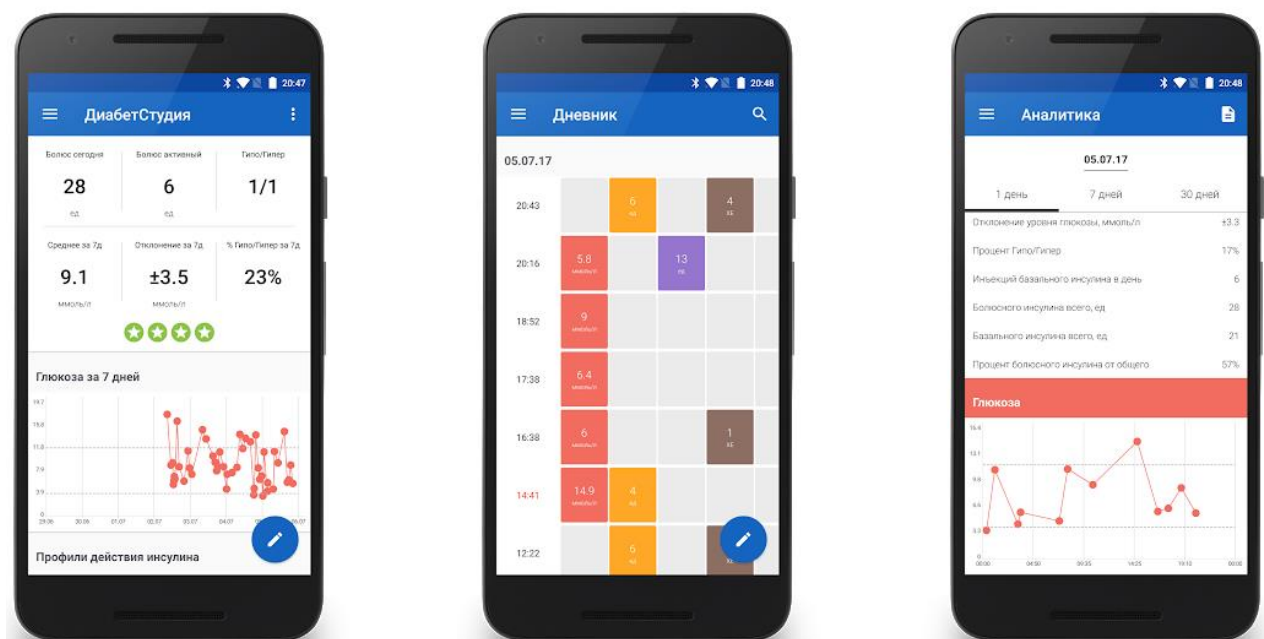


Рисунок 1.3 – Програмне забезпечення ДіабетСтудия

Програмне забезпечення ДіабетСтудия є повністю безкоштовним застосунком для діабетиків з першим типом діабету. ДіабетСтудия допомагає вести щоденник для відстеження основних показників: рівень інсуліну, цукор у крові, харчування, дієта, заняття спортом та багато іншого. За допомогою графіків можна зрозуміти, як раціон пацієнта впливає на показники. Опція «Звіт для лікаря» допомагає разом із лікарем детально стежити за перебігом лікування [7]-[9].

У щоденнику програми є пошук записів за тегамі. Є можливість записів батьками. Усі дані синхронізуються між пристроями.

Додаток має калькулятор інсуліну. Під час розрахунку калькулятор пам'ятає індивідуальні коефіцієнти чутливості пацієнта до інсуліну та тривалість дії конкретного інсуліну.

Перевагами програмного забезпечення ДіабетСтудия є такі:

– можливість ведення щоденника для відстеження основних показників: рівень інсуліну, цукор у крові, харчування, дієта, заняття спортом та багато іншого;

- можливість ведення записів батьками пацієнта;
- наявність калькулятора інсуліну;
- відображення даних у вигляді зручних та зрозумілих графіків;
- можливість синхронізації даних між пристроями;
- безкоштовність програми.

До недоліків програми варто віднести відсутність можливості прогнозування стану пацієнта, хворого на діабет, з урахуванням поточної медичної інформації та попередніх даних.

Програмне забезпечення ГлікемічнийІндекс (рисунок 1.4) є мобільним застосунком для роботи під керуванням операційної системи Android [7]-[9].

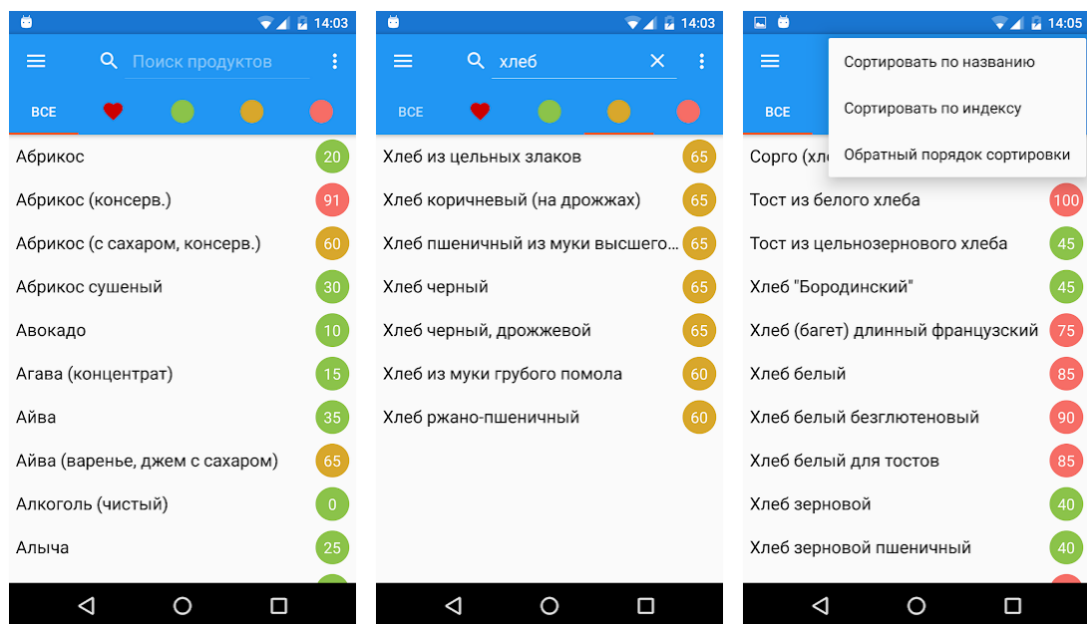


Рисунок 1.4 – Програмне забезпечення ГлікемічнийІндекс

Програмне забезпечення ГлікемічнийІндекс дозволяє оцінювати глікемічний індекс. За допомогою зручного пошуку можна знайти потрібний продукт та подивитися його глікемічний індекс [7]-[9].

Перевагами програмного забезпечення ГлікемічнийІндекс є такі:

- можливість розрахунку глікемічного індексу;
- робота з мобільних пристроїв;
- безкоштовність програми.

До недоліків програми варто віднести відсутність можливості прогнозування стану пацієнта, хворого на діабет, з урахуванням поточної медичної інформації та попередніх даних. Також у цьому програмному забезпеченні немає можливості ввести кількість спожитих хлібних одиниць чи кілокалорій у певний період прийому їжі, щоб перевірити, чи пацієнт не скоштував їжі більше за свою норму. Іншими недоліками є такі: відсутність калькулятору інсуліну, відсутність відображення даних у вигляді зручних та зрозумілих графіків, відсутність можливості синхронізації даних між пристроями.

Програмне забезпечення One Drop (рисунок 1.5) є мобільним застосунком для роботи під керуванням операційних систем Android, iOS, Apple Watch [7]-[9].

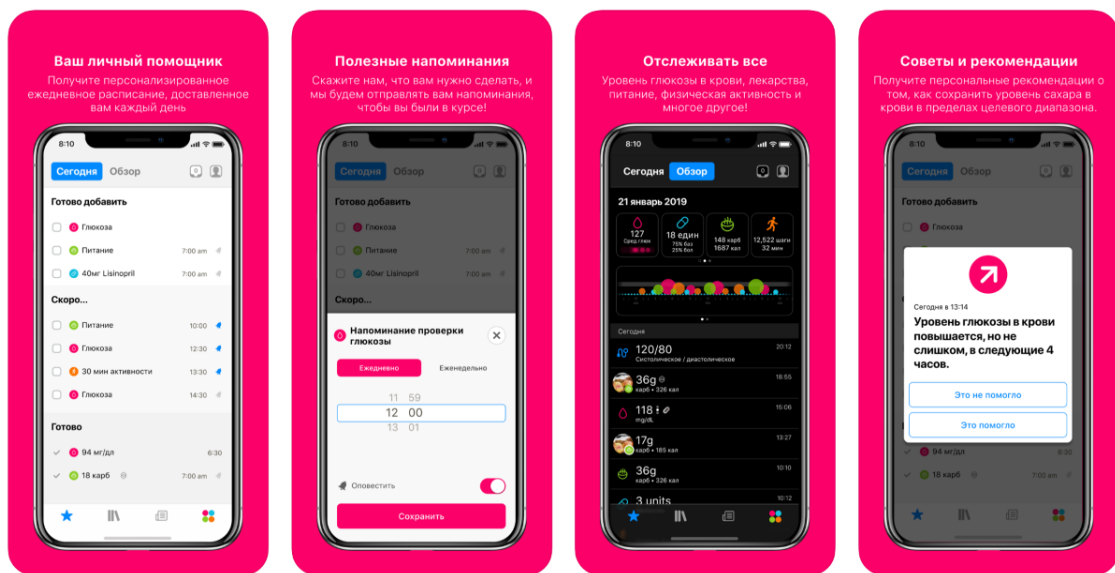


Рисунок 1.5 – Програмне забезпечення One Drop

Програмне забезпечення One Drop є дуже корисною програмою, проте для використання цього потрібен спеціальний глюкометр із доставкою смужок. Без глюкометра користуватися One Drop також можна, проте у цьому

випадку ви отримуєте персональний розклад із нагадуваннями. Крім того, немає можливості відстежувати рівень цукру в крові, прийоми ліків та їжі, активність, вагу, артеріальний тиск та багато іншого. Також у додатку є корисні поради та рекомендації з використанням штучного інтелекту. На iOS є інтеграція з Apple Health та Apple Watch. Завдяки чому відкривається інтеграція з безліччю сторонніх додатків на кшталт: Fitbit, Dexcom, One Touch, Accu-Chek, Agamatrix, iHealth, Dario, Garmin, Nike+, Strava, Qardio, My Fitness Pal, Lose it!, Lark. У бібліотеці One Drop є дані щодо тисячі продуктів харчування з рецептами. Є можливість налаштування автоматичної інсулінової помпи [7]-[9].

Перевагами програмного забезпечення One Drop є такі:

- наявність корисних порад та рекомендацій з використанням штучного інтелекту;
- можливість розрахунку глікемічного індексу;
- робота з мобільних пристроїв;
- можливість інтеграції з іншим програмним забезпеченням;
- можливість налаштування автоматичної інсулінової помпи;
- наявність вбудованої бази з інформацією про продукти харчування з рецептами.

До недоліків програми One Drop варто віднести такі:

- необхідність спеціального глюкометра із доставкою смужок;
- відсутність можливості прогнозування стану пацієнта, хворого на діабет, з урахуванням поточної медичної інформації та попередніх даних;
- немає можливості відстежувати рівень цукру в крові, прийоми ліків та їжі, активність, вагу, артеріальний тиск та ін.

Програмне забезпечення Contour diabetes app (рисунок 1.6) є мобільним застосунком для роботи під керуванням операційних систем Android, iOS.

Програмне забезпечення Contour diabetes app є аналогом One Drop, проте для його використання необхідним є спільна робота з глюкометром Contour. Усі вимірювання можна проводити в автоматичному режимі, а в

додатку слідкувати за результатами. Звичайно, можна і самостійно вносити всі необхідні дані про рівень глюкози, харчування та інше [7]-[9].

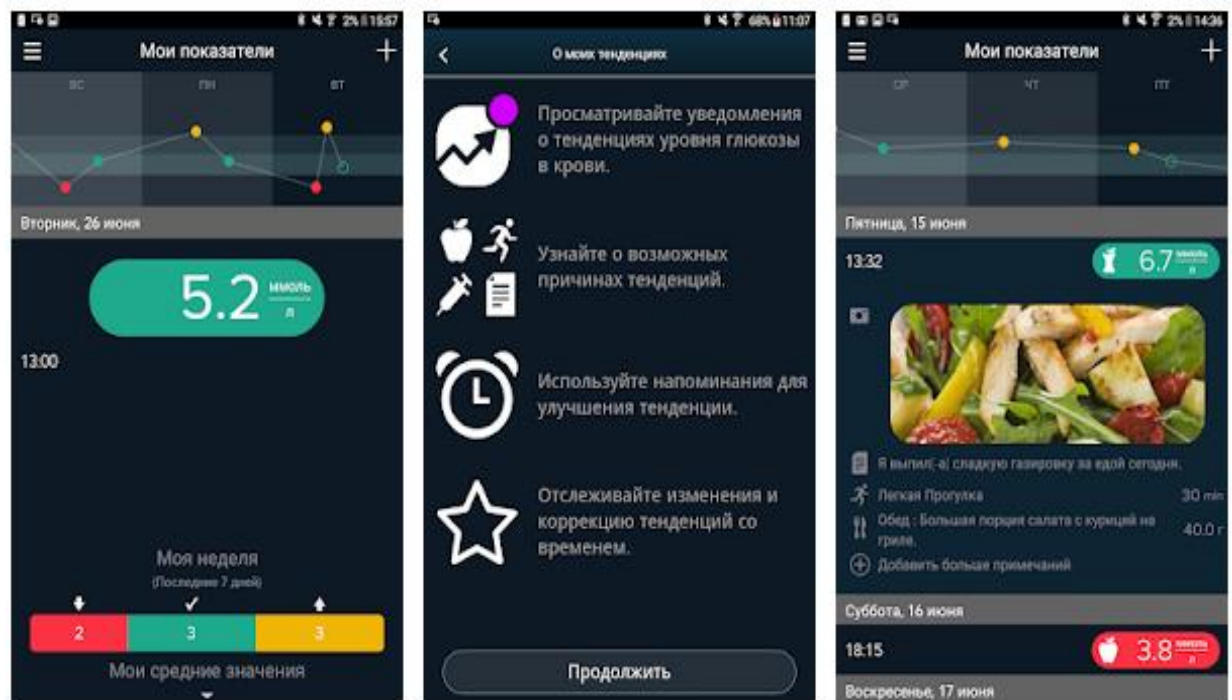


Рисунок 1.6 – Програмне забезпечення Contour diabetes app

Перевагами програмного забезпечення Contour diabetes app є такі:

- можливість розрахунку глікемічного індексу;
- наявність порад та рекомендацій щодо життя в умовах наявності хвороби на цукровий діабет;
- можливість розрахунку глікемічного індексу;
- робота з мобільних пристроїв;
- можливість інтеграції з іншим програмним забезпеченням;
- можливість налаштування автоматичних розрахунків;
- наявність вбудованої бази з інформацією про продукти харчування з рецептами.

До недоліків програми Contour diabetes app варто віднести такі:

- необхідність спеціального глюкометра Contour;
- відсутність можливості прогнозування стану пацієнта, хворого на діабет, з урахуванням поточної медичної інформації та попередніх даних

– немає можливості відстежувати рівень цукру в крові, прийоми ліків та їжі, активність, вагу, артеріальний тиск та ін.

Програмне забезпечення Glucose Buddy (рисунок 1.7) є програмою для зберігання даних, завдяки якій пацієнти, хворі на діабет, можуть відстежувати показники глюкози, вуглеводів, дозування інсуліну та фізичні навантаження. Програма надає користувачам можливість переглядати ці дані у будь-який час через свій онлайн-акаунт [7]-[9].

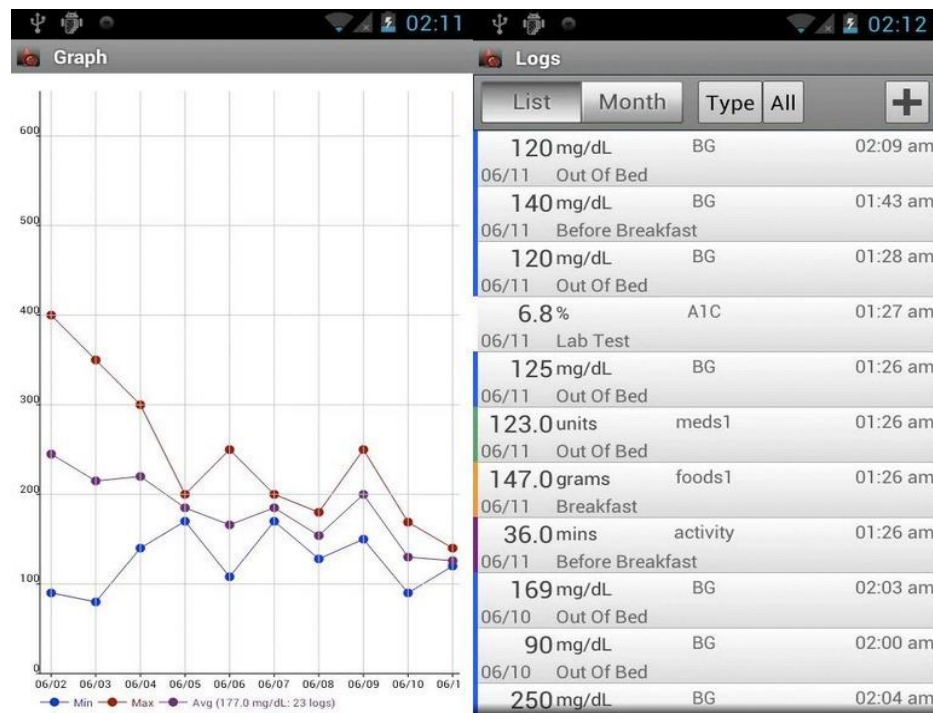


Рисунок 1.7 – Програмне забезпечення Glucose Buddy

Перевагами програмного забезпечення Glucose Buddy є такі:

- можливість зберігання даних, завдяки якій пацієнти, хворі на діабет, можуть відстежувати показники глюкози, вуглеводів, дозування інсуліну та фізичні навантаження;
- можливість перегляду персональних даних у будь-який час через свій онлайн-акаунт;
- можливість розрахунку деяких показників;
- робота з мобільних пристроїв;
- можливість інтеграції з іншим програмним забезпеченням.

До недоліків програми Glucose Buddy варто віднести такі:

- відсутність можливості прогнозування стану пацієнта, хворого на діабет, з урахуванням поточної медичної інформації та попередніх даних
- відсутність відображення даних у вигляді зручних та зрозумілих таблиць, відсутність можливості синхронізації даних між пристроями.

Програмне забезпечення Diabetes Pal (рисунки 1.8) за своїми функціями відстеження є схожим на деякі інші програми, зокрема, на Glucose Buddy.

Проте, на відміну від них, Diabetes Pal також дозволяє користувачам вносити дані кров'яного тиску, ваги та тривалості сну, а також додавати примітки, а модернізований інтерфейс програми надає відмінні візуальні діаграми та графіки [7]-[9].

Перевагами програмного забезпечення Diabetes Pal є такі:

- можливість зберігання даних, завдяки якій пацієнти, хворі на діабет, можуть відстежувати показники глюкози, вуглеводів, дозування інсуліну та фізичні навантаження;
- можливість перегляду персональних даних у будь-який час через свій онлайн-акаунт;
- можливість додавання даних про кров'яний тиск, вагу та тривалість сну, а також додавати примітки;
- робота з мобільних пристроїв;
- можливість інтеграції з іншим програмним забезпеченням.

До недоліків програми Diabetes Pal варто віднести такі:

- відсутність можливості прогнозування стану пацієнта, хворого на діабет, з урахуванням поточної медичної інформації та попередніх даних
- відсутність відображення даних у вигляді зручних та зрозумілих таблиць, відсутність можливості синхронізації даних між пристроями.

Виконаємо порівняльний аналіз досліджених програмних засобів, що призначені для полегшення життя пацієнтів, хворих на цукровий діабет



Рисунок 1.8 – Програмне забезпечення Diabetes Pal

У таблиці 1.1 наведено порівняльну характеристику програмних продуктів, що використовуються для полегшення життя пацієнтів, хворих на цукровий діабет.

Таблиця 1.1 – Порівняльна характеристика програмних продуктів, що використовуються для полегшення життя пацієнтів, хворих на цукровий діабет

Критерії порівняння програм для діабетиків	Програмне забезпечення для полегшення життя хворих на діабет				
	Diabetes: M	DiaMeter	Диабет-Студия	Гликемический Индекс	One Drop
1	2	3	4	5	6
Можливість використання як мобільний застосунок	+	+	+	+	+

Продовження таблиці 1.1

1	2	3	4	5	6
Наявність вбудованої бази їжі	+	—	—	+	+
Наявність калькулятора вуглеводів та доз інсуліну	+	+	+	+	+
Можливість створення звітів	+	+	+	—	+
Зручне відображення даних у вигляді графіків	—	+	+	—	
Можливість синхронізації даних між пристроями	+—	+	+	—	+
Можливість прогнозування стану пацієнта	—	—	—	—	—

Таким чином, суттєвим недоліком існуючого програмного забезпечення, що використовується для полегшення життя пацієнтів, хворих на цукровий діабет, є те, що такі програмні засоби надають рекомендації на основі досить простих моделей (семантичних, регресійних та ін.), проте не дозволяють оцінити (спрогнозувати) стан пацієнта через деякий час.

Тому доцільно розробити методи та програмні засоби, що дозволять виконувати прогнозування стану хворих на діабет через деякий час з урахуванням поточної медичної інформації та попередніх даних.

1.3 Аналіз діагностичних моделей для прогнозування на основі медичних даних

Для прогнозування стану пацієнтів, хворих на діабет, доцільно використовувати наявну діагностичну інформацію медичного характеру, яка є досить легкою для вимірювання. До такої інформації можуть належати дані про тиск пацієнта, його власну оцінку самопочуття, вживання деяких

препаратів (зокрема інсуліну), а також інформацію про рівень серцевого сигналу, який легко може зніматися за допомогою досить дешевих приладів.

Отже, необхідно розробити програмне забезпечення, яке може діагностувати стан пацієнтів на основі даних, які відносяться до числового та категоріального типів, а також даних сигнального типу (зокрема, сигналів, що описують рівень серцебиття пацієнта).

Проаналізуємо різні типи моделей [10]-[18], що можуть застосовуватися для діагностування на основі медичних даних (таблиця 1.2).

Таблиця 1.2 – Порівняльна характеристика типів діагностичних моделей на основі медичних даних

Критерії порівняння типів діагностичних моделей	Діагностичні моделі			
	регресійні	семантичні	багато-шарові перцептрони	згорткові нейромережі (CNN)
Складність процесу синтезу	низька	середня	висока	висока
Інтерпретовність	висока	висока	низька	низька
Точність діагностування	низька	середня	висока	висока
Необхідність залучення експерта для побудови моделі	—	+	—	—
Можливість роботи з чисельними даними	+	+	+	+
Можливість роботи з даними у вигляді сигналів	—	—	—	+

Як видно з таблиці 1.2, для розробки програмного забезпечення, яке може діагностувати стан пацієнтів на основі даних, що відносяться до числового та категоріального типів, а також даних сигнального типу (зокрема, сигналів, що описують рівень серцебиття пацієнта), доцільно обрати моделі на основі згорткових нейронних мереж, оскільки такі мережі дозволяють

працювати з даними, що описують сигнали, зокрема рівень серцебиття пацієнта.

Проаналізуємо моделі на основі згорткових нейронних мереж, а також їх модифікації та методи їх синтезу. Згорткові нейронні мережі (convolutional neural networks, CNN) є найбільш розповсюдженою формою глибоких нейронних мереж [19]-[22].

Головною особливістю CNN є те, що вони здатні оброблювати дані на декількох рівнях абстракції шляхом пошуку різного роду ознак, що дозволяє більш точно і з кращою продуктивністю класифікувати вхідні дані.

Також згорткові мережі частково стійку до різних спотворень. Однією з головних переваг CNN є її простота розпаралелювання роботи і навчання, що дає значний приріст в продуктивності. Типова згорткова мережа складається з вхідного шару, декількох послідовностей загорткових шарів, що завершуються шаром max-pooling, парою повнозв'язних шарів і вихідним шаром [19]-[22].

Згортковий шар генерує матрицю вихідних сигналів шляхом послідовного обходу матриці вхідних даних матрицею ваг. На кожному етапі матриці ваг у відповідність ставляє деякий фрагмент вихідної матриці, відповідні чарунки матриць перемножуються, після чого всі результати множення підсумовуються і кінцевий результат записується в чарунку матриці виходу.

Далі матриця ваг зміщується в бік на одну (або декілька) чарунку і процес повторюється, поки матриця виходу не заповниться (рис. 1.9). кожна послідовність згорткових шарів вирішує задачу пошуку ознак.

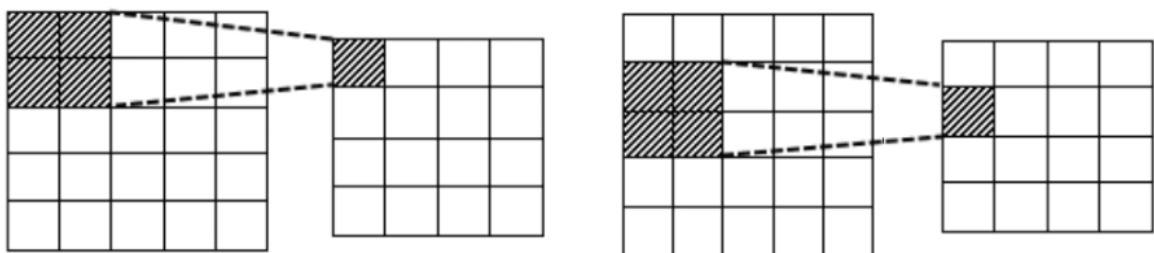


Рисунок 1.9 – Приклад функціонування згорткового шару

Підвибірковий шар (max-pooling layer) використовується для значного зменшення розмірності матриць з метою підвищення продуктивності (рис. 1.10). Даний шар застосовується тоді, коли попередні шари здатні точно визначити ярко виражені ознаки. Після чого визначаються ознаки більш високого рівня абстракції. За аналогією зі згортковим шаром, підвибірний шар також генерує свою вихідну матрицю шляхом послідовного обходу матриці вхідних даних, але з двома відмінностями, першою з яких є те, що фрагменти вхідної матриці не перетинаються (тому довжина матриці виходу буде дорівнювати довжині матриці входу, поділеної на довжину матриці шару max-pooling). Другою відмінністю є те, що під вибірковий шар не має вагових коефіцієнтів і виконує операцію виключно на входах.

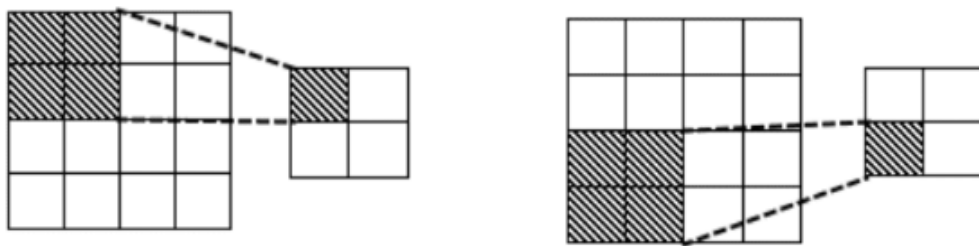


Рисунок 1.10 – Схема функціонування шару max-pooling

Повнозв'язний шар виконує задачу класифікації на основі даних про знайдені ознаки, що отримані з попередніх шарів. Разом з вихідним шаром вони працюють за принципом багат шарового перцептрона. Результатом роботи повнозв'язних шарів є вектор ймовірностей належності вхідного сигналу до кожного з класів.

Основною сферою застосування CNN є розпізнавання зображень (визначення як одного, так і множини образів), проте вони можуть бути застосовані також в інших сферах, що потребують розпізнавання ознак і класифікації (таких, як розпізнавання текстів на природніх мовах та визначення діагнозів пацієнтів). Згорткові мережі можуть об'єднуватися з

іншими типами нейронних мереж для вирішення більш складних задач [19]- [22].

У роботі [21] пропонують алгоритм класифікації зображень, що має дві основні фази: фаза навчання – це фаза навчання системи для отримання моделі, яка використовується для класифікації зображень; та фаза тестування – ця фаза використовується для тестування навченої моделі з використанням нових даних, що не входять до навчальних даних. На етапі навчання дані та мітка їх класу, що використовуються як вхід у запропоновану CNN з комбінацією архітектури GoogLeNet та Residual Network (залишкової мережі), що була названа як CNN IncRes. Результатом навчання є класифікаційні моделі, які потім використовуються у фазі тестування. Запропонований метод класифікує зображення з точністю 70,84%. Дана точність класифікації не є досить високою в загальному випадку, але для специфічних задач запропонований підхід може бути досить ефективним.

Модифікація згорткової нейронної мережі, яка запропонована в роботі [22], направлена на покращення швидкості функцій розпізнавання та зменшення часу роботи мережі. Для вирішення цих завдань запропоновано використовувати метод виключень та оптимізатор SGN за допомогою використання TensorFlow. Використання SGN змушує CNN працювати швидше, оскільки мережа не використовує цілі навчальні екземпляри на кожній ітерації і може динамічно регулювати оцінку матриць першого та другого порядку градієнта кожного параметра відповідно до функції виключення, що дозволяє зменшити обчислення і підвищити швидкість обчислення.

Окрім цього, при вдосконаленні функції активації, алгоритм дозволяє уникнути такої проблеми, коли вихід нейронного вузла стає рівним нулю, тим самим підвищуючи точність розпізнавання та зменшення витрат часу. Недоліком такого підходу є те, що мережа працює на невеликому наборі даних.

В роботі [23] було запропонована модифікація алгоритму навчання DCNN, що робить DCNN більш ефективним з точки зору точності та складності. Умовно існує два параметри (один в шарі згортки, інший в повнозв'язному шарі) в DCNN, які необхідно відкоригувати. Однак, в запропонованому методі, параметр загорткового шару повинен бути відкоригований. Відповідні модифікації виконано в повнозв'язному шарі для оцінки значень ваг без ітераційної процедури. В роботі пропонується алгоритм модифікації алгоритму навчання (без зміни архітектури алгоритму) за допомогою використання досить простої математичної процедури. При цьому запропонований підхід покращує точність класифікації на 3% в порівнянні з класичним CNN, при цьому ваги в мережі не налагоджені, що знижує її складність.

Задача класифікації медичної статистики з використанням CNN була розглянута в роботі [24]. Автори пропонують наступну процедуру. На початку роботи модель виконує регресію LASSO для виявлення внесених змінних або атрибутів даних.

Використовуючи декілька варіантів випадкових наборів даних, LASSO виконується кілька разів для перевірки послідовності внесення даних, що є необхідним для контролю істинно-негативних значень вибірки.

Далі, алгоритм голосування застосовується для видалення неважливих значень, таким чином досягається зменшення розміру вибірки. Також, не рекомендується застосовувати традиційні методи зменшення розміру вибірки (локальне лінійне вкладення, аналіз основних компонентів тощо), оскільки ці методи лінійною або нелінійною комбінацією значень даних. Далі отримані дані відправляють до 1D-зворотну CNN. Така модель має точність прогнозування на рівні 79,5%.

У роботі [25] запропоновано використовувати компактку архітектуру згорткової нейронної мережі, яка мінімізує кількість ядер на нижньому рівні мережі, та відділяти інформацію про кольори від вихідних зображень.

На вищому рівні мережі класифікатор навчається комбінуючи ознаки з компактною мережі, при чому навчання проходить лише на чорно-білих зображеннях з обмеженою кількістю ядер. Запропонована архітектура зменшує розмір мережі за рахунок відділення інформації про кольори, вихідного зображення та зменшенням кількості ядер. Компактна мережа має на 40% меншу кількість параметрів налаштувань, але зі збереженням продуктивності класичної CNN.

Авторами у своїй роботі [26] пропонується операція згортання пікселів (PAC) – це ефективна модифікація стандарту згортки, в яких ваги фільтру множать на просторово-мінливе ядро, що залежить від ознак локальних пікселів, які навчаються.

PAC – це узагальнення декількох популярних методів фільтрування, тому його можна використовувати для широкого діапазону випадків. Загальна схема PAC змінює стандарт згортки на вході v шляхом зміни просторово-інваріатного фільтра W з адаптуючим ядром K . Адаптуюче ядро побудовано використовуючи заздалегідь визначені або вивчені функції f .

У роботі [27] пропонуються острівну модель, яка використовує різні представлення на кожному острові. Модель перетворює індивідів з одного представлення в інше під час міграцій.

Показано, що така модель допомагає еволюційному алгоритму відійти від локального оптимуму та вирішити проблеми простого представлення еволюційного алгоритму. Пропонується підхід з острівною моделлю з двома населеннями, в якій на одному острові використовується стандартне бінарне кодування, а на іншому стандартний код Грея.

У роботі [28] представлено нову крупно зернову (паралельну) (coarse-grain) модель еволюційних алгоритмів. Дана модель має назву модель злиття популяцій островів (PMIM) та замінює систему міграцій населення, що об'єднуються як механізм дисперсії рішень використовуючи традиційну острівну модель.

PMIM виключає необхідність вибору топології взаємозв'язку між островами, яка є одним із параметрів, що найбільше впливає на точність та ефективність еволюційного алгоритму. Коли населення двох або більше островів об'єднані, а потім зменшені, глобальна чисельність населення зменшується, як і кількість островів, що розвиваються. Безпосереднім наслідком цього є зниження кількості оцінок фітнес функції з відповідним зменшенням часу виконання. Також модель PMIM має додатковий шар, що допомагає в дослідженні та контролі використання евристичного пошуку, що призводить до більшої точності знайденого рішення. Запропонована модель є швидшою та точнішою порівняно з класичною моделлю.

У роботі [29] пропонують змінити підхід до зміни парадигми, який розглядає острів як єдине середовище існування його населення, який названий LIMGA.

Основна ідея даного підходу полягає в гармонізації відмінностей кожного острова. Запропонований метод має механізм, що названий стратегією локалізації, в якій існує два види островів: господар і раб. Господар керує міграцією, встановлюючи параметрами IMGA і приймає рішення, щодо острова-раба, який повинен виконувати міграцію в напрямку найкращого острова-раба.

З іншого боку острів-раб виконує обчислення або керує основним генетичним алгоритмом для створення кращого індивіда в кожному поколінні.

В загальному випадку існує лише один острів-господар, який контролює IMGA, та більше, ніж один острів-рабів для контролю обчислень ГА. У запропонованому алгоритмі острів-господар встановлює початкові параметри і розподіляє індивідів островів до островів очікування. Острови-раби (SGA, PGA, IGA) виконується обробку основного генетичного алгоритму для отримання кращого індивіда в кожному поколінні.

MW (Migration window) – це буфер острова-господаря, що зберігає значення найкращого індивіда. Протокол міграцій – це механізм контролю за особою, що переміщується через острів-господаря.

1.4 Висновки за розділом 1

Досліджено процес діагностування захворюваності на діабет на основі медичних даних. Проаналізовано поняття цукрового діабету та принципи терапії хворих на діабет.

Виконано огляд існуючих програмних продуктів, що використовуються для полегшення життя пацієнтів, хворих на цукровий діабет. Показано, що суттєвим недоліком існуючого програмного забезпечення, що використовується для полегшення життя пацієнтів, хворих на цукровий діабет, є те, що такі програмні засоби надають рекомендації на основі досить простих моделей (семантичних, регресійних та ін.), проте не дозволяють оцінити (спрогнозувати) стан пацієнта через деякий час.

Тому доцільно розробити методи та програмні засоби, що дозволять виконувати прогнозування стану хворих на діабет через деякий час з урахуванням поточної медичної інформації та попередніх даних.

Відзначено, що для прогнозування стану пацієнтів, хворих на діабет, доцільно використовувати наявну діагностичну інформацію медичного характеру, яка є досить легкою для вимірювання. До такої інформації можуть належати дані про тиск пацієнта, його власну оцінку самопочуття, вживання деяких препаратів (зокрема інсуліну), а також інформацію про рівень серцевого сигналу, який легко може зніматися за допомогою досить дешевих приладів. Обґрунтовано необхідність розробки програмного забезпечення, яке може діагностувати стан пацієнтів на основі даних, які відносяться до числового та категоріального типів, а також даних сигнального типу (зокрема, сигналів, що описують рівень серцебиття пацієнта).

Проаналізовано різні типи моделей, що можуть застосовуватися для діагностування на основі медичних даних.

Для розробки програмного забезпечення, яке може діагностувати стан пацієнтів на основі даних, що відносяться до числового та категоріального типів, а також даних сигнального типу (зокрема, сигналів, що описують рівень

серцебиття пацієнта), доцільно обрати моделі на основі згорткових нейронних мереж, оскільки такі мережі дозволяють працювати з даними, що описують сигнали, зокрема рівень серцебиття пацієнта.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір та обґрунтування технологій розробки для створення програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

Для розробки програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі доцільно обрати мови програмування Python.

Виконаємо аналіз та порівняння мови програмування Python з іншими мовами та засобами розробки.

Середовище Weka є програмним забезпеченням, яке вільно розповсюджується та дозволяє працювати з методами попередньої обробки даних та машинного навчання. Програмне забезпечення Weka створено так, щоб можна було застосувати математичні методи на різних вибірках даних за допомогою різних способів. Це програмне забезпечення забезпечує підтримку процесу оброблення даних, зокрема підготовку вхідних даних, оброблення даних, а також візуалізацію даних й результатів аналізу [30].

Програмне забезпечення Weka має графічний інтерфейс користувача, який надає доступ до різних можливостей Weka з використанням засобів меню. У графічному інтерфейсі користувача, є можливість ознайомитися з даними з файлу та виконати аналіз даних за допомогою дерева вирішальних правил [30].

Інтерфейс Knowledge Flow програмного забезпечення Weka дозволяє виконувати проєктування конфігурації даних для потокової обробки. Інтерфейс Knowledge Flow програмного забезпечення Weka дозволяє встановити спосіб поєднання даних в потоці, ці дані являють собою джерело даних, алгоритми навчання, засоби попередньої обробки даних, засоби оцінювання та виведення результатів шляхом візуалізації [30].

Інтерфейс Experimenter програмного забезпечення Weka допомагає виконувати аналіз експериментальних даних за допомогою використання

математичних методів регресії, класифікації та інших. Інтерфейс Experimenter програмного забезпечення Weka дозволяє виконувати аналіз та порівняння різних засобів навчання та різних методів.

Інтерфейс Workbench програмного забезпечення Weka являє собою графічний інтерфейс. Інтерфейс Workbench поєднує вище описані інтерфейси. Інтерфейс Workbench програмного забезпечення Weka легко налаштовується, цей інтерфейс дозволяє користувачу визначати, які програми та вбудовані модулі будуть відображатися на панелі користувача.

Програмне забезпечення Weka надає можливість користувачу використовувати не тільки стандартні, відомі алгоритми, а і завантажувати інші. Система управління пакетами програмного забезпечення Weka дозволяє користувачу програми встановлювати лише ті пакети, що є корисними для конкретного користувача [30].

Програмне забезпечення Keras представляє собою бібліотеку глибокого навчання. Це програмне забезпечення дозволяє досліднику ефективно будувати та навчати моделі глибокого навчання. Процес глибокого навчання моделей за допомогою програмне забезпечення Keras є досить простим, для чого необхідним є лише наявність вхідних даних, а також встановлення кількості епох або ітерацій навчання та інших параметрів параметричного синтезу. Кінцевим результатом навчання глибоких моделей є те, що більшість моделей можуть бути створені за допомогою суттєво меншої кількості рядків коду. Використовуючи програмне забезпечення Keras, користувач отримає досить високу продуктивність створення моделей, зберігаючи час на реалізації програмного коду синтезу моделей глибокого навчання [31].

У програмному забезпеченні Keras можна створювати більш складні моделі, для чого необхідно застосовувати різні API та Model, а також класи шарів, можна встановлювати власні налаштування з метою задоволення потреб конкретного користувача. Різні можливості засобів API програмного забезпечення Keras підтримують процес синтезу графічних моделей, що

можуть бути подані як функції Python. Також варто відзначити, що класи моделей та шарів дозволяють реалізувати різні специфічні експериментальні моделі та шари нейронних мереж глибокого навчання [31].

Важливим способом взаємодії, кооперації засобів Python та Weka є динамічна мова Python. Мова Python є реалізацією динамічної об'єктно-орієнтованої мови Python, яка написана за допомогою мови Java та дозволяє запускати мову програмування Python на різних відомих платформах Java [31].

Для забезпечення можливості роботи на мові програмування Python у середовищі Weka можна застосовувати вільно розповсюджене середовище розробки TigerPython. Середовище розробки TigerPython надає можливість створювати користувальницькі математичні моделі за допомогою функціоналу засобів середовища Weka [31].

Python — це високорівнева об'єктно-орієнтована мова програмування, створена Гвідо ван Россумом [32]–[35]. Її також називають мовою програмування загального призначення, оскільки вона використовується майже в кожній області, про яку ми можемо думати, як зазначено нижче: веб-розробка, розробка програмного забезпечення, розробка гри на основі штучного інтелекту та машинного навчання, аналітика даних. Цей список можна продовжувати, але чому Python настільки популярний, давайте подивимося в наступній темі [32]–[35].

Мова програмування Python є універсальною, що означає, що вона широко використовується в кожній області, тому причина в тому, що вона є дуже простою для розуміння, масштабованою, завдяки чому швидкість розробки настільки висока. Вивчення Python не потребує жодного досвіду програмування, тому він також популярний серед розробників. Python має простіший синтаксис, схожий на англійську мову, а також синтаксис дозволяє розробникам писати програми з меншою кількістю рядків коду. Оскільки це відкритий код, існує багато бібліотек, які полегшують роботу розробників, що в кінцевому підсумку призводить до високої продуктивності. Вони можуть

легко зосередитися на бізнес-логіці та її вимогливих навичках у цифрову еру, коли інформація доступна у великих наборах даних [32]–[35].

Python є мовою сценаріїв, обрана для використання в багатьох програмах. Він має чіткий синтаксис, сучасні конструкції та підхід «батареї в комплекті», що робить його ідеальним для використання як всередині, так і за межами екосистеми. За останні кілька років Python став першокласною мовою у сфері розробки сучасного програмного забезпечення, управління інфраструктурою та аналізу даних. Це вже не допоміжна мова, а головна сила у створенні веб-додатків і управлінні системами, а також ключовий драйвер вибуху в аналітиці великих даних і машинному інтелекті [32]–[35].

Кількість функцій у самій мові невелика, що вимагає відносно невеликих витрат часу або зусиль для створення перших програм. Синтаксис Python розроблений таким чином, щоб він був читабельним і зрозумілим. Ця простота робить Python ідеальною мовою навчання, а новачкам дозволяє швидко опанувати її [32]–[35]. В результаті розробники витрачають більше часу на роздуми над проблемою, яку вони намагаються вирішити, і менше часу на роздуми про складність мови або розшифровку коду, залишеного іншими [32]–[35].

У наш час Python одночасно популярний і широко використовується, про що свідчать високі рейтинги в опитуваннях і велика кількість проектів на GitHub, які використовують Python.

Python працює на всіх основних операційних системах і платформах, а також на більшості другорядних. Багато основних бібліотек і служб на основі API мають прив'язки або обгортки Python, що дозволяє Python вільно взаємодіяти з цими службами або безпосередньо використовувати ці бібліотеки.

Незважаючи на те, що сценарії та автоматизація охоплюють велику частину випадків використання Python, Python також використовується для створення програмного забезпечення професійної якості, як окремі програми,

так і як веб-сервіси. Python може бути не найшвидшою мовою, але те, чого йому не вистачає в швидкості, він компенсує універсальністю [32]–[35].

Кожна редакція мови Python додає нові корисні функції, що полегшують розробку програмного забезпечення. Наприклад, асинхронні операції та сопрограми тепер є стандартними частинами мови, що полегшує написання програм Python, які виконують одночасну обробку [32]–[35].

Python є об'єктно-орієнтованою мовою з відкритим вихідним кодом, а також високорівневою й універсальною мовою програмування. У той же час його використовує велика кількість дизайнерів у всьому світі для створення веб-сайтів, мобільних додатків і програм із графічним інтерфейсом. Відмінним аспектом Python є те, що, пишучи менш читабельний код, програмісти можуть конкретизувати концепції.

Крім того, розробники можуть скористатися перевагами багатьох баз Python, щоб мінімізувати час і енергію, необхідні для створення складних і широких програмних додатків. Наразі цю мову програмування використовують різноманітні веб-сайти з високим трафіком, а саме Google, Yahoo Maps, Yahoo Groups, Shopzilla, Weekly News, Linux, а також Web Therapy [32]–[35].

Мову програмування Python використовують для розробки ігрових, політичних, наукових та освітніх програм. Однак розробники все ще використовують різні версії мов програмування. Відповідно до частки ринку, даних та статистики використання Python, більше використовують її другу редакцію.

Таким чином, у наш час Python є мовою програмування, яка має чітко організований синтаксис і потужні інструменти для вирішення будь-яких завдань. Більше того, синтаксис цієї мови є дуже близьким до простого математичного мислення.

Саме тому, мова Python часто вибирається як основна мова програмування для розроблення сучасного програмного забезпечення [32]–[35].

Результати аналізу мов програмування для створення програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі наведено у таблиці 2.1.

Таблиця 2.1 – Порівняння мов програмування для розробки програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

Критерій порівняння мов програмування	Мова програмування		
	Python	Java	C#
Легкість вивчення	+	—	—
Підтримка процесу розробки на різних операційних системах	+	+	±
Наявність математичних бібліотек для синтезу різних видів проностичних моделей	+	±	±
Можливість програмування паралельних обчислень	+	+	+
Зручність для написання математичних програм	+	±	±

Отже, за результатами аналізу мов програмування для створення програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі, вирішено як мову програмування для розробки ПЗ обрати Python.

2.2 Вибір середовища для розробки програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

Середовище Eclipse є широко використовуваним інтегрованим середовищем розробки (IDE), яке дозволяє розробляти програмне забезпечення за допомогою різних мов програмування, зокрема, Python, C/C++, Java, JavaScript, PHP тощо. Це середовище дозволяє легко об'єднати підтримку кількох мов та інші функції в будь-який із пакетів за замовчуванням, а Eclipse Marketplace надає практично необмежену настройку та розширення. IDE Eclipse надає засоби розробки, які потрібно для швидкого впровадження інновацій. Це середовище розробки дозволяє створювати програмне забезпечення з використанням хмарних технологій. Середовище розробки Eclipse підтримує роботу веб-переглядача для розробки з розміщеними робочими просторами або пакетів для настільних комп'ютерів, щоб отримати доступ до сучасного середовища розробки для Java, JavaScript, CSS та інших. Середовище розробки Eclipse надає різні засоби для створення програмного забезпечення різних типів, зокрема служб та інструментів, які можна зібрати в нові IDE або пакети, адаптовані до конкретного завдання. Середовище розробки Eclipse надає кілька платформ для створення плагінів для настільних інструментів, розподілених служб, які використовуються хмарними IDE, та інтерфейсів браузера [36].

Суттєвою перевагою Eclipse є те, що це інтегроване середовище розробки з відкритим кодом, яке підтримує розробку застосунків на різних мовах програмування [36].

Середовище розробки Eclipse використовуються для розробки великої кількості програмних застосунків, оскільки ця платформа є провідним середовищем розробки на основі різних мов програмування. Він використовується, оскільки забезпечує функціональність додавання кількох різних компонентів на платформу, які відомі як плагіни. ІТ-організації різних типів віддають перевагу IDE Eclipse через цю функціональність [36].

Платформа Eclipse також підтримує різні інструменти, розробку веб-сервісів, а також підтримує редагування файлів XML. На платформі доступна вкладка перегляду сервера, на якій програму можна розгорнути на будь-якому

сервері відповідно до вимог користувача. Розробники також використовують ту ж платформу IDE, що й Eclipse, що підтримує програми на основі Spring [36]. Середовище розробки Eclipse використовується для цілей моделювання, оскільки може бути ідеальним прикладом для інструменту моделювання, а моделі можна легко створювати за допомогою інструментів цієї платформи. Для створення моделей проектів програмного забезпечення використовується технологія EPC [36].

Зведемо у порівняльну таблицю характеристику різних інтегрованих середовищ розробки, які можна використовувати для створення програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі за допомогою мови Python (таблиця 2.2).

Таблиця 2.2 – Порівняння середовищ розробки програмного забезпечення для створення ПЗ прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

Критерій порівняння	Середовище розробки		
	Eclipse	PyDev	PyCharm
Підтримка роботи репозиторіями програмних проектів	+	+—	+—
Підтримка розробки на різних мовах програмування	+	—	—
Кросплатформність	+	+—	+—
Можливість інтеграції текстів програм на різних мовах програмування	+	—	—
Зручність використання	+	+—	+—

Таким чином для створення ПЗ прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі було обрано середовище розробки Eclipse, яке є досить зручним для використання.

2.3 Структурна схема програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

Структурну схему програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі наведено на рисунку 2.1.

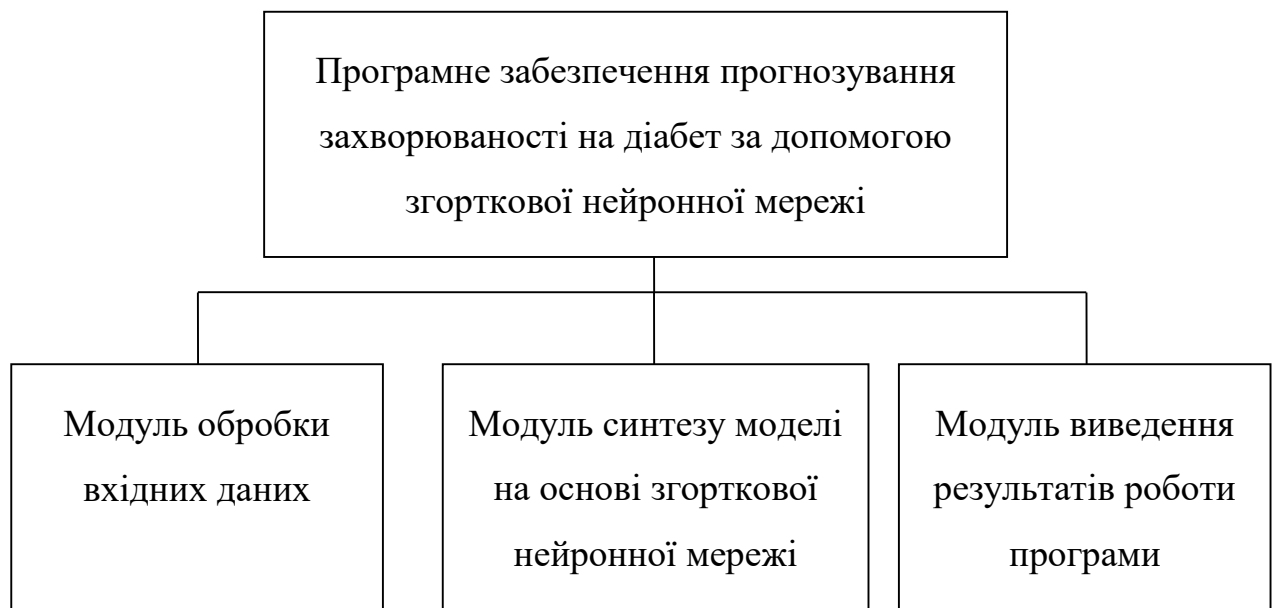


Рисунок 2.1 – Структура програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

Як видно з рисунку 2.1, програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі складається з модуля обробки вхідних даних, модуля синтезу моделі на основі згорткової нейронної мережі та модуля виведення результатів роботи програми.

Модуль обробки вхідних даних призначений для роботи з вхідними даними, зокрема, для зчитування вхідної інформації з файлів, а також для забезпечення можливості введення необхідної інформації користувачем вручну.

Модуль синтезу моделі на основі згорткової нейронної мережі використовується для побудови діагностичної моделі прогнозування захворюваності на діабет. Цей модуль містить набір програмних компонент, що дозволяють виконувати структурно-параметричний синтез моделей на основі нейромережових технологій та методів навчання нейронних мереж. Існує можливість вибору методу навчання, а також розробки власних програмних засобів у вигляді модулів, що реалізують ті чи інші методи параметричного синтезу нейронних мереж.

Модуль виведення результатів роботи програми дозволяє користувачу виконувати виведення інформації у файли у вигляді табличних даних, а також виведення інформації у вигляді графіків, що є досить зручними для подальшого аналізу.

Процес функціонування програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі складається з таких етапів:

- етап синтезу моделі на основі згорткової нейронної мережі. На цьому етапі користувач вхідні дані у вигляді вибірки діагностичних даних про стан захворюваності на діабет. Після цього програмне забезпечення виконує синтез діагностичної моделі на основі згорткової нейронної мережі, що дозволяє виконувати прогнозування захворюваності на діабет на основі різних вхідних даних про стан пацієнта;

- етап використання синтезованої діагностичної моделі на основі згорткової нейронної мережі для прогнозування захворюваності на діабет на основі різних вхідних даних про стан пацієнта. Користувач може здійснювати захворюваності на діабет на основі різних вхідних даних про стан пацієнта.

2.4 Висновки за розділом 2

Для створення програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі було обрано мови програмування Python, оскільки ця мова є досить зручною та ефективною для розв'язання завдань аналізу даних. Як інтегроване середовище розробки обрано Eclipse, оскільки таке середовище є вільно розповсюджуваним та зручним для використання.

Розроблено структуру програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Визначено, що основними модулями програми є модуль обробки вхідних даних, модуль синтезу моделі на основі згорткової нейронної мережі та модуль виведення результатів роботи програми. Наведено опис та призначення кожного модуля програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

3 ОПИС ПРОГРАМИ

3.1 Розробка модифікованого методу синтезу моделі на основі згорткової нейронної мережі

Як зазначено вище, як базис для прогнозування стану пацієнтів, хворих на діабет, доцільно використовувати наявну діагностичну інформацію медичного характеру, яка є досить легкою для вимірювання. Для розробки програмного забезпечення, яке може діагностувати стан пацієнтів на основі даних, що відносяться до числового та категоріального типів, а також даних сигнального типу (зокрема, сигналів, що описують рівень серцебиття пацієнта), доцільно обрати моделі на основі згорткових нейронних мереж, оскільки такі мережі дозволяють працювати з даними, що описують сигнали, зокрема рівень серцебиття пацієнта.

Проте суттєвим недоліком згорткової нейронної мережі є надто велика варіативність її параметрів, через що незрозуміло, для якої задачі і обчислювальної потужності які налаштування потрібні. Так, до параметрів, що варіюються відносять: кількість шарів, розмірність ядра згортки для кожного шару, кількість ядер для кожного з шарів, крок зміщення ядра при обробці шару, необхідність шарів субдискретизації, ступінь зменшення ними розмірності, функція зменшення розмірності (вибір максимуму, середнього тощо), передатна функція нейронів, наявність і параметри вихідної повнозв'язної нейромережі на виході загорткової. Всі ці параметри суттєво впливають на результат, тож обираються емпірично. Проте, на даний недолік можна поглянути з іншого боку. Він надає можливість для широкого спектру наукового дослідження CNN та модифікації структурної схеми мережі [19] – [26].

З описаної вище схеми функціонування базової загорткової мережі, можна зробити висновок, що одним з варіантів модифікації є заміна класифікатора. Замість використання багат шарового перцептрона як

класифікатора в даній роботі пропонується використовувати мережу ART-1 [19] – [26].

Мережа адаптивної резонансної теорії (ART-1) має важливі переваги в порівнянні з іншими нейронними мережами, оскільки може донавчатися в процесі свого функціонування, зберігаючи при цьому накопичену раніше інформацію, а також може знаходити на своїх входах нову інформацію. Цим вона відрізняється від більшості нейронних мереж, в тому числі, перцептронів, які для запам'ятовування додаткової інформації вимагають, як правило, свого повного перенавчання і реагують на будь-яку вхідну інформацію (нову чи ту, що вже зберігається в пам'яті) вихідними сигналами [19] – [26].

Мережа ART-1 реалізує алгоритм кластеризації, що дуже схожий на алгоритм «послідовного лідера» (sequential leader clustering algorithm). Відповідно з цим алгоритмом перший вхідний сигнал вважається зразком першого кластера. Наступний вхідний сигнал порівнюється зі зразком першого кластера. Говорять, що вхідний сигнал «прямує за лідером» та належить першому кластеру, якщо відстань до зразка першого кластера менше порогу. В іншому випадку другий вхідний сигнал – зразок другого кластеру. Цей процес повторюється для всіх наступних вхідних сигналів. Таким чином, кількість кластерів зростає з часом і залежить як від значення порогу, так і від метрики відстані, що використовується для порівняння вхідних сигналів і зразків класів [19] – [26].

Мережа складається з двох шарів: вхідного (порівняльного), кількість нейронів в якому фіксована і дорівнює розміру векторів, і вихідного (змагального), зі змінною кількістю нейронів, де кожному нейрону відповідає один клас кластерів [19] – [26].

Зв'язок між будь-якими нейронами, що належать одному шару, відсутній. Зв'язок між нейронами, що належать різним шарам, здійснюється за принципом «кожен з кожним». Усім зв'язкам у відповідність ставлять деяку вагу. Зв'язки від вихідних нейронів називають висхідними, а від вихідних – низхідними. В якості функції стану кожного нейрона розглядається відстань в

деякій спеціальності метриці між вхідним вектором x і вектором ваг нейрону w . Функція активації нейронів вихідного шару вираховується як зважена вагами входів сума координат вхідного вектора [19] – [26]

Навчання нейронної мережі є неспостережним і полягає в корекції ваг нейронів таким чином, щоб кожен нейрон був переможцем для максимальної кількості векторів із множини, що навчається і при цьому виконувався деякий критерій близькості. Ємність мережі ART-1 співпадає з кількістю нейронів другого шару і може збільшуватися в процесі функціонування мережі, що і є відмінною особливістю мереж ART від інших класів штучних нейронних мереж [19] – [26].

Алгоритм функціонування мережі полягає в наступному. На початковому етапі в мережі тільки один вхідний нейрон, ваги мережі ініціалізуються наступними значеннями [19] – [26]:

$$b_{ij}(0) = \frac{1}{1+n}, \quad (3.1)$$

$$c_{ji}(0) = 1, 0 \leq i \leq n-1, 0 \leq j \leq m-1, 0 \leq p \leq 1, \quad (3.2)$$

де, $b_{ij}(t)$ - це синоптична вага зв'язку від i -го нейрону першого шару до j -го нейрону другого шару в момент часу t ; $c_{ji}(t)$ - це синоптична вага зв'язку від j -го нейрону першого шару до i -го нейрону другого шару в момент часу t ; p – значення параметра відповідності. Ваги $c_{ji}(t)$ та $b_{ij}(t)$ визначають зразок, відповідний нейрону j . Параметр відповідності p показує, наскільки вхідний сигнал повинен співпадати з одним із замкнених зразків, щоб вони вважалися схожими. Значення параметра, близьке до одиниці, означає майже повний збіг. При малих значення p , навіть якщо вхідний сигнал та зразок дуже відрізняються, вважаються такими, що належать до одного кластеру [19] – [26].

При пред'явленні мережі нового бінарного вхідного зразку для кожного нейрону другого шару обчислюється значення функції стану [19] – [26]:

$$y_j = \sum_{i=0}^{n-1} b_{ij}(t)x_i, 0 \leq j \leq m-1, \quad (3.3)$$

після чого обирається зразок з найбільшою відповідністю:

$$y_k = \max_j (y_j). \quad (3.4)$$

Після цього знайдений зразок і вхідний сигнал порівнюються з параметром відповідності. Якщо

$$\frac{\sum_{i=0}^{n-1} c_{ik}(t)x_i}{\sum_{i=0}^{n-1} x_i} > p, \quad (3.5)$$

то вхідний зразок вважається схожим на зразок з найбільшим значенням відповідності. У цьому випадку зразок адаптується під нові дані за допомогою функції кон'юнкції:

$$c_{ji}(t+1) = c_{ji}(t) \cdot x_j, \quad (3.6)$$

$$b_{ij}(t+1) = \frac{c_{ji}(t)x_i}{0.5 + \sum_{i=9}^{n-1} c_{ji}(t)x_i}. \quad (3.7)$$

В протилежному випадку вважається, що нейрон з найбільшим значенням тимчасово блокується, і пошук повторюється серед нейронів, що

залишилися. Якщо активних нейронів більше не залишилося, то вважається що вхідний зразок відрізняється від всіх вхідних зразків, і створюється новий нейрон-зразок, ідентичний даному зразку, ваги якого ініціалізуються наступними значеннями [19] – [26]:

$$c_{ji}(t+1) = x_j, \quad (3.8)$$

$$b_{i,n+1}(t+1) = \frac{x_i}{0.5 + \sum_{i=9}^{n-1} x_i} . \quad (3.9)$$

На останньому етапі активуються всі неактивні нейрони [40].

Таким чином, запропоновано модифікований метод синтезу моделей для прогнозування стану захворюваності на діабет, який полягає у поєднанні згорткових нейронних мереж та нейромереж адаптивної резонансної теорії (ART-1).

3.2 Функціонування програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

На основі розробленої структурної схеми програми створено програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Розроблене програмне забезпечення використовує модифікований метод синтезу моделей для прогнозування стану захворюваності на діабет, який полягає у поєднанні згорткових нейронних мереж та нейромереж адаптивної резонансної теорії.

Процес функціонування програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі складається з таких етапів:

– етап синтезу моделі на основі згорткової нейронної мережі. На цьому етапі користувач вхідні дані у вигляді вибірки діагностичних даних про стан захворюваності на діабет. Після цього програмне забезпечення виконує синтез діагностичної моделі на основі згорткової нейронної мережі, що дозволяє виконувати прогнозування захворюваності на діабет на основі різних вхідних даних про стан пацієнта;

– етап використання синтезованої діагностичної моделі на основі згорткової нейронної мережі для прогнозування захворюваності на діабет на основі різних вхідних даних про стан пацієнта. Користувач може здійснювати захворюваності на діабет на основі різних вхідних даних про стан пацієнта.

Схему функціонування програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі наведено на рисунку 3.1.



Рисунок 3.1 – Схема функціонування програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

3.3 Основні рішення по реалізації програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

Для створення програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі було розроблено структури даних та функціональні компоненти, що реалізують моделі на основі згорткових мереж, а також методи їх навчання.

Структуру даних CNN у вигляді класу, що реалізує згорткову нейронну мережу, наведено нижче.

```
class CNN(torch.nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.featr=64
        self.padding= torch.nn.ReplicationPad2d((31,64,0,0))
        self.convNN = torch.nn.Conv2d(1,self.featr,(1,64))
        self.batch = Batchlayer(self.featr)
        self.avegrPool = torch.nn.AvegrPool2d((1,8))
        self.funcT = torch.nn.Linear(64, 2)
        self.MAXsft=torch.nn.LogSftMAX(dim=1)
        self.MAXsft1=torch.nn.SftMAX(dim=1)
        self.cnn=torch.nn.CNN(64, 2)
```

Структуру даних Batchlayer у вигляді класу, що реалізує шар згорткової нейронної мережі, наведено нижче.

```
class Batchlayer(torch.nn.Module):
    def __init__(self, dim):
        super(Batchlayer, self).__init__()
        self.gamma=torch.nn.Parameter(torch.Tensor(1,dim,1,1))
        self.beta=torch.nn.Parameter(torch.Tensor(1,dim,1,1))
```

```
self.gamma.data.uniform_(-0.2, 0.2)
```

```
self.beta.data.uniform_(-0.2, 0.2)
```

Клас, що реалізує відображення результатів у графічному вигляді подано нижче.

```
class FtrVisb():
    def __init__(self, ann):
        self.ann = ann
        self.ann.eval()
        self.MAXsft=nn.SftMAX(dim=1)

        if smpllabel==0:
            labelstr='alert'
        else:
            labelstr='drowsy'

        img = plt.igure(figsize=(14,6))
        gridlayout      =      gridspec.GridSpec(ncols=2,      nrows=5,
imgure=img,wspace=0.2, hspace=0.5)
        axis0 = img.add_subplot(gridlayout[0:2,0])
        axis1 = img.add_subplot(gridlayout[4,0])
        axis2 = img.add_subplot(gridlayout[0:5,1])
        axis3 = img.add_subplot(gridlayout[2:4,0])
        rawsignal=allsignals[smplidx].cpu().detach().numpy().squeeze()
        numChan=multichannelsignal.shape[0]
        smpllength=multichannelsignal.shape[1]
        MAXVal=nonp.max(nonp.abs(rawsignal))
        scr = self.ann.padding(allsignals)
        scr = self.ann.convNN(scr)
        scr = self.ann.batch(scr)
        scr = torch.nn.ELU()(scr)
```

```

scr=self.ann.avegrPool(scr)
scr =scr.squeeze()
scr=scr.permute(2, 0, 1)
scr = self.ann.cnn(scr)[0]

statHid=scr[:,smplidx,:].squeeze()
statHid=self.MAXsft(statHid)
statHid=statHid[:,state].cpu().detach().numpy()

flength=statHid.shape[0]
duplication=int(384/statHid.shape[0])

heatmap=nonp.zeros(flength)
for i in range(1,flength):
    heatmap[i]=statHid[i]-statHid[i-1]

```

Фрагмент тексту програми, що реалізує процес навчання нейронної мережі, наведено нижче.

```

def run():
    fileN = r'DTset.mat'
    tmp = sio.loadmat(fileN)
    dataXfeatr=nononp.array(tmp['EEGsmpl'])

    label=nononp.array(tmp['substate'])
    subIdx=nononp.array(tmp['subindex'])

    label.astype(int)
    subIdx.astype(int)

    smplnum=label.shape[0]

```

```

numChan=30
classes=2
subjnum=11
smpllength=3

lr=1/100
sf=256

btchSize = 50
epochNum =15

dataYfeatr=nononp.zeros(smplnum,dtype=nononp.longlong)

for i in range(smplnum):
    dataYfeatr[i]=label[i]

selectedchan=[28]
rawx=dataXfeatr

dataXfeatr=dataXfeatr[:,selectedchan,:]
numChan=len(selectedchan)

for i in range(2,3):
    trainindx=nononp.where(subIdx!= i)[0]
    dataXtrain=dataXfeatr[trainindx]
    trainX      =      dataXtrain.reshape(dataXtrain.shape[0],1,numChan,
smpllength*sf)
    trainY=dataYfeatr[trainindx]

    class_output= myModl(datasmplInput)

```

```

errorRes_s_lbl = loss_class(class_output, class_lbl)

testindx=nononp.where(subIdx == i)[0]

testX=dataXfeatr[testindx]
rawdataXfeatr=rawx[testindx]

testX = testX.reshape(testX.shape[0], 1,numChan, smpllength*sf)
testY=dataYfeatr[testindx]

train    =    torch.utils.data.TensorDTset(torch.from_numpy(trainX),
torch.from_numpy(trainY))

train_loader = torch.utils.data.DataLoader(train, btchSize=btchSize,
shuffle=True)

class_output= myModl(datasmplInput)
errorRes_s_lbl = loss_class(class_output, class_lbl)

myModl = CNN().double().cuda()

optimizer = optim.Adam(myModl.parameters(), lr=lr)
loss_class = torch.nn.NLLLoss().cuda()

for p in myModl.parameters():
    p.requires_grad = True

for epoch in range(epochNum):
    for j, data in enumerate(train_loader, 0):
        inputs, labels = data
        datasmplInput = inputs.cuda()

```

```

class_lbl = labels.cuda()
myModl.zero_grad()
myModl.train()
class_output= myModl(datasmplInput)
errorRes_s_lbl = loss_class(class_output, class_lbl)
errorRes = errorRes_s_lbl
errorRes.backward()
optimizer.step()

myModl.train(False)
with torch.no_grad():
    testX = torch.DoubleTensor(testX).cuda()

    answer = myModl(testX)
    probs=nononp.exp(answer.cpu().numpy())

    preds = probs.argmax(axis = -1)
    smplVis =FtrVisb(myModl)

```

Варто відзначити, що реалізовано і інші структури даних та функції, що у сукупності утворюють програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Більш детальний фрагмент тексту програми наведено у додатку.

3.4 Висновки за розділом 3

Запропоновано модифікований метод синтезу моделей для прогнозування стану захворюваності на діабет, який полягає у поєднанні згорткових нейронних мереж та нейромереж адаптивної резонансної теорії.

Розроблено програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Розроблене програмне забезпечення використовує модифікований метод синтезу моделей для прогнозування стану захворюваності на діабет, який полягає у поєднанні згорткових нейронних мереж та нейромереж адаптивної резонансної теорії. Запропоновано схему функціонування ПЗ прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

Процес функціонування розробленого програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі складається з таких етапів. Етап синтезу моделі на основі згорткової нейронної мережі, на якому користувач вхідні дані у вигляді вибірки діагностичних даних про стан захворюваності на діабет. Після цього програмне забезпечення виконує синтез діагностичної моделі на основі згорткової нейронної мережі, що дозволяє виконувати прогнозування захворюваності на діабет на основі різних вхідних даних про стан пацієнта. На етапі використання синтезованої діагностичної моделі користувач може здійснювати захворюваності на діабет на основі різних вхідних даних про стан пацієнта.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ ПРОГНОЗУВАННЯ ЗАХВОРЮВАНOSTІ НА ДІАБЕТ ЗА ДОПОМОГОЮ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ

4.1 Призначення й умови застосування програми

Програмна призначена для прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

Програма прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі забезпечує виконання функцій:

- завантаження файлу з числовими даними;
- поділ вибірки даних на тестову та навчальну вибірки;
- синтез моделі на основі згорткової нейронної мережі для прогнозування захворюваності на діабет;
- можливість прогнозування захворюваності на діабет на основі синтезованої моделі.

Для функціонування програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі повинні бути наявні такі апаратні та програмні засоби:

- процесор Intel Dual-Core з тактовою частотою не менше, ніж 1.8 ГГц;
- персональний комп'ютер з можливістю встановлення програмного забезпечення;
- оперативна пам'ять 1 Гб або більше;
- наявність операційної системи;
- не менше 700 Мб вільного простору на жорсткому диску;
- наявність клавіатури для введення інформації та пристрою виведення для відображення результатів.

Функціональні характеристики розробленого програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі наведеному у першому додатку до пояснювальної записки (додаток А). Опис програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі та текст програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі наведені у додатках Б та В.

4.2 Характеристики програми

Програма прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі розроблена за допомогою мови програмування Python та інтегрованого середовища розробки Eclipse.

Основними модулями програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі є модуль обробки вхідних даних, модуль синтезу моделі на основі згорткової нейронної мережі та модуль виведення результатів роботи програми.

Процес функціонування розробленого програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі складається з таких етапів.

Етап синтезу моделі на основі згорткової нейронної мережі, на якому користувач вхідні дані у вигляді вибірки діагностичних даних про стан захворюваності на діабет. Після цього програмне забезпечення виконує синтез діагностичної моделі на основі згорткової нейронної мережі, що дозволяє виконувати прогнозування захворюваності на діабет на основі різних вхідних даних про стан пацієнта.

На етапі використання синтезованої діагностичної моделі користувач може здійснювати захворюваності на діабет на основі різних вхідних даних про стан пацієнта.

4.3 Інструкція по експлуатації програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

4.3.1Звернення до програми

Для запуску програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі необхідно виконати наступні дії. Упевнитися у відповідності характеристик апаратного забезпечення вимогам до мінімальних системних вимог програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

Після цього необхідно упевнитися в справності всіх компонентів комп'ютера та електронного носія, з якого здійснюватиметься запуск програми. Потім необхідно завантажити файл з програмою прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

4.3.2Вхідні й вихідні дані

До вхідних даних програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі відносяться вибірки діагностичних даних про стан захворюваності на діабет, параметри синтезованих нейромереж та методів їх побудови, вхідні дані про стан пацієнта.

Вихідні дані програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі є такі: синтезована згорткова нейронна мережа для діагностування захворюваності на діабет; результати діагностування.

4.3.3Повідомлення

Розроблене програмне забезпечення прогнозування захворюваності на

діабет за допомогою згорткової нейронної мережі забезпечує стійке функціонування програми. У випадку неправильних налаштувань можуть виникнути певні повідомлення. Приклад повідомлень про помилки наведено на рисунку 4.1.

```
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
    at tigerjython.jython.JythonInterpreter$.tigerjython$jython$JythonInterpreter$$sysModules$lzycompute(Jyth
    at tigerjython.jython.JythonInterpreter$.tigerjython$jython$JythonInterpreter$$sysModules(JythonInterpret
    at tigerjython.jython.JythonInterpreter.reset(JythonInterpreter.scala:581)
    at tigerjython.gui.PythonTextArea.runCode(PythonTextArea.scala:647)
    at tigerjython.gui.DocumentManager$.runCode(DocumentManager.scala:482)
    at tigerjython.gui.UserControls$$anonfun$14.apply$mcV$sp(UserControls.scala:131)
    at scala.swing.Action$$anon$2.apply(Action.scala:60)
    at scala.swing.Action$$anon$1.actionPerformed(Action.scala:78)
    at javax.swing.AbstractButton.fireActionPerformed(Unknown Source)
    at javax.swing.AbstractButton$Handler.actionPerformed(Unknown Source)
    at javax.swing.DefaultButtonModel.fireActionPerformed(Unknown Source)
    at javax.swing.DefaultButtonModel.setPressed(Unknown Source)
    at javax.swing.plaf.basic.BasicButtonListener.mouseReleased(Unknown Source)
    at java.awt.AWTEventMulticaster.mouseReleased(Unknown Source)
    at java.awt.Component.processMouseEvent(Unknown Source)
    at javax.swing.JComponent.processMouseEvent(Unknown Source)
    at java.awt.Component.processEvent(Unknown Source)
    at java.awt.Container.processEvent(Unknown Source)
    at java.awt.Component.dispatchEventImpl(Unknown Source)
    at java.awt.Container.dispatchEventImpl(Unknown Source)
    at java.awt.Component.dispatchEvent(Unknown Source)
    at java.awt.LightweightDispatcher.retargetMouseEvent(Unknown Source)
    at java.awt.LightweightDispatcher.processMouseEvent(Unknown Source)
    at java.awt.LightweightDispatcher.dispatchEvent(Unknown Source)
    at java.awt.Container.dispatchEventImpl(Unknown Source)
```

Рисунок 4.1 – Повідомлення про помилки

4.4 Тестування та результати функціонування програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі було протестовано шляхом синтезу нейромережі на основі згенерованих вхідних вибірок даних.

Програму можна запустити з командного рядка. Приклад запуску програми з командного рядка наведено на рисунку 4.2.

```
C:\Users\Blvsk>E:
E:\>cd PZZDiabetCNN\model\main
E:\PZZDiabetCNN\model\main> python CNN.py
```

Рисунок 4.2 – Приклад запуску програми з командного рядка

Після запуску програми у командному рядку виводиться поточна інформація про проміжні результати роботи програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі, зокрема виводиться значення функції втрат, точність (рисунок 4.3).

```
Epoch 251/300
1/73 [.....] - ETA: 0s - loss: 0.3367 - accuracy: 1.00
46/73 [=====> .....] - ETA: 0s - loss: 0.5238 - accuracy: 0.80
73/73 [=====] - 0s 1ms/step - loss: 0.5738 - accuracy:
0.7916 - val_loss: 0.7541 - val_accuracy: 0.7182

Epoch 252/300
1/73 [.....] - ETA: 0s - loss: 0.3280 - accuracy: 0.95
45/73 [=====> .....] - ETA: 0s - loss: 0.5451 - accuracy: 0.81
73/73 [=====] - 0s 1ms/step - loss: 0.5182 - accuracy:
0.7991 - val_loss: : 0.7681 - val_accuracy: 0.7208
```

Рисунок 4.3 – Приклад проміжних результатів роботи програми

Результати роботи програми наведено на рисунку 4.4.

```
1/73 [>.....] - ETA: 0s - loss: 0.4118 - accuracy: 0.7902
73/73 [=====] - 0s 1ms/step - loss: 0.3672 - accuracy:
0.8471
accuracy: 84.71
```

Рисунок 4.4 – Приклад результатів роботи програми

Тестування та навчання згорткової нейронної мережі проводилося на наборі даних, що містять інформацію щодо захворювання на діабет. Набір даних поданий у вигляді вибірки складається з 9 ознак та 768 екземплярів.

Тестування згорткової нейронної мережі проводилося таким чином, що набір даних ділиться на тестову та навчальну вибірки у відсотковому

відношенні 80/20%. Крім цього, для збільшення адекватності отриманих результатів набір даних розподілявся на десять частин. Результати тестування наведено у таблиці 4.1.

Таблиця 4.1 – Результати тестування програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі

№ з/п	Нейромережева модель	Параметри		
		Навчальна вибірка		Тестова вибірка
		точність, %	час роботи алгоритму синтезу, хв	точність, %
1	Класична CNN	81,23	14,37	75,91
2	Модифікована CNN	84,71	16,13	78,36

Таким чином, з результатів тестування можна зробити висновок про те, що в порівнянні з класичною мережею модифікована здійснює діагностування з більшою точністю, проте час синтезу такої згорткової мережі є дещо більшим. Варто відзначити, що при розв’язанні реальних практичних задач це є допустимим, оскільки точність класифікації в даному випадку є більш значущою. Проте, також очевидна, що отримана точність діагностування за допомогою синтезованої нейромережевої моделі є не досить високою, що обумовлює доцільність подальших досліджень та розробки нових модифікацій згорткових нейронних мереж та методів їх навчання.

4.5 Висновки за розділом 4

Надано призначення й умови застосування програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Подано основні характеристики програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

Наведено результати тестування розробленого програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. З результатів тестування можна зробити висновок про те, що в порівнянні з класичною мережею модифікована здійснює діагностування з більшою точністю, проте час синтезу такої згорткової мережі є дещо більшим. Варто відзначити, що при розв'язанні реальних практичних задач це є допустимим, оскільки точність класифікації в даному випадку є більш значущою. Проте, також очевидна, що отримана точність діагностування за допомогою синтезованої нейромережевої моделі є не досить високою, що обумовлює доцільність подальших досліджень та розробки нових модифікацій згорткових нейронних мереж та методів їх навчання.

ВИСНОВКИ

У результаті виконання дипломної кваліфікаційної роботи магістра було досліджено процес діагностування захворюваності на діабет на основі медичних даних.

Виконано огляд існуючих програмних продуктів, що використовуються для полегшення життя пацієнтів, хворих на цукровий діабет. Показано, що суттєвим недоліком існуючого програмного забезпечення, що використовується для полегшення життя пацієнтів, хворих на цукровий діабет, є те, що такі програмні засоби надають рекомендації на основі досить простих моделей (семантичних, регресійних та ін.), проте не дозволяють оцінити (спрогнозувати) стан пацієнта через деякий час.

Відзначено, що для прогнозування стану пацієнтів, хворих на діабет, доцільно використовувати наявну діагностичну інформацію медичного характеру, яка є досить легкою для вимірювання. До такої інформації можуть належати дані про тиск пацієнта, його власну оцінку самопочуття, вживання деяких препаратів (зокрема інсуліну), а також інформацію про рівень серцевого сигналу, який легко може зніматися за допомогою досить дешевих приладів. Обґрунтовано необхідність розробки програмного забезпечення, яке може діагностувати стан пацієнтів на основі даних, які відносяться до числового та категоріального типів, а також даних сигнального типу (зокрема, сигналів, що описують рівень серцебиття пацієнта).

Проаналізовано різні типи моделей, що можуть застосовуватися для діагностування на основі медичних даних. Для розробки програмного забезпечення, яке може діагностувати стан пацієнтів на основі даних, що відносяться до числового та категоріального типів, а також даних сигнального типу (зокрема, сигналів, що описують рівень серцебиття пацієнта), доцільно обрати моделі на основі згорткових нейронних мереж, оскільки такі мережі дозволяють працювати з даними, що описують сигнали, зокрема рівень серцебиття пацієнта.

Для створення програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі було обрано мови програмування Python, оскільки ця мова є досить зручною та ефективною для розв'язання завдань аналізу даних. Як інтегроване середовище розробки обрано Eclipse, оскільки таке середовище є вільно розповсюджуваним та зручним для використання.

Розроблено структуру програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Визначено, що основними модулями програми є модуль обробки вхідних даних, модуль синтезу моделі на основі згорткової нейронної мережі та модуль виведення результатів роботи програми. Наведено опис та призначення кожного модуля програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

Наукова новизна роботи полягає у тому, що запропоновано модифікований метод синтезу моделей для прогнозування стану захворюваності на діабет, який полягає у поєднанні згорткових нейронних мереж та нейромереж адаптивної резонансної теорії.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Розроблене програмне забезпечення використовує модифікований метод синтезу моделей для прогнозування стану захворюваності на діабет, який полягає у поєднанні згорткових нейронних мереж та нейромереж адаптивної резонансної теорії. Запропоновано схему функціонування ПЗ прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

Процес функціонування розробленого програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі складається з таких етапів. Етап синтезу моделі на основі згорткової нейронної мережі, на якому користувач вхідні дані у вигляді вибірки діагностичних даних про стан захворюваності на діабет. Після цього

програмне забезпечення виконує синтез діагностичної моделі на основі згорткової нейронної мережі, що дозволяє виконувати прогнозування захворюваності на діабет на основі різних вхідних даних про стан пацієнта. На етапі використання синтезованої діагностичної моделі користувач може здійснювати захворюваності на діабет на основі різних вхідних даних про стан пацієнта.

Виконано тестування розробленого програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. З результатів тестування можна зробити висновок про те, що в порівнянні з класичною мережею модифікована здійснює діагностування з більшою точністю, проте час синтезу такої згорткової мережі є дещо більшим. Варто відзначити, що при розв'язанні реальних практичних задач це є допустимим, оскільки точність класифікації в даному випадку є більш значущою. Проте, також очевидна, що отримана точність діагностування за допомогою синтезованої нейромережевої моделі є не досить високою, що обумовлює доцільність подальших досліджень та розробки нових модифікацій згорткових нейронних мереж та методів їх навчання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Seiglie, J. A. Predictors of health facility readiness for diabetes service delivery in low- and middle-income countries: The case of Bangladesh [Text] / Seiglie, J. A., S. M. Edson, B. Tahmina // Diabetes Research and Clinical Practice, 2021. – Vol. 175. – P. 108784-108784. – DOI: 10.1016/J.DIABRES.2021.108784.
2. Диабет [Электрон. ресурс]. – Режим доступа: <https://www.who.int/ru/news-room/fact-sheets/detail/diabetes>.
3. Что такое диабет? [Электрон. ресурс]. – Режим доступа: <https://www.onetouch.ru/facts-about-diabetes/start-your-journey/about-diabetes>.
4. Из-за чего возникает сахарный диабет [Электрон. ресурс]. – Режим доступа: <https://gorzdrav.org/blog/izza-chego-voznikaet-sakharnyj-diabet/>.
5. Сахарный диабет - причины, диагностика и лечение [Электрон. ресурс]. – Режим доступа: <https://daily-med.com.ua/stati/sakharnyj-diabet>.
6. Как питаться при сахарном диабете [Электрон. ресурс]. – Режим доступа: <https://medkamensk.ru/patsientam-2/poleznaya-informatsiya/kak-pitatsya-pri-saharnom-diabete/>.
7. 7 мобильных приложений для контроля сахарного диабета [Электрон. ресурс]. – Режим доступа: <https://medgadgets.ru/novosti-2/nauka-i-issledovaniya/zdorovye/diabet/7-mobilnykh-prilozhenij-dlya-kontrolya-saxarnogo-diabeta.html>.
8. Топ 8 приложений для диабетиков (Android, iPhone) [Электрон. ресурс]. – Режим доступа: <https://www.healthapps.ru/top/diabetes/>.
9. 11 мобильных приложений для больных сахарным диабетом [Электрон. ресурс]. – Режим доступа: <https://blog.themarfa.name/11-mobilnykh-prilozhenii-dlia-bolnykh-sakharnym-diabetom/>.
10. Tenne, Y. Computational Intelligence in Expensive Optimization Problems [Text] / Y. Tenne, C.-K. Goh. – Berlin: Springer: 2010. – 800 p.
11. Mumford, C. L. Computational Intelligence [Text] / C. L. Mumford, L. C. Jain. – Berlin : Springer-Verlag, 2009. – 732 p.

12. Russel, S. Artificial intelligence: a modern approach [Text] / S. Russel, P. Norvig. – New Jersey: Prentice Hall, 2009. – 1152 p.
13. Computational intelligence: collaboration, fusion and emergence [Text] / [ed. Ch. L. Mumford]. – New York: Springer, 2009. – 752 p.
14. Intelligent data analysis: an introduction [Text] / [eds. M. Berthold, D. J. Hand]. – New York: Springer Verlag, 2007. – 525 p.
15. Субботін, С. О. Нейронні мережі [Текст] / С. О. Субботін, А. О. Олійник. – Запоріжжя: ЗНТУ. – 2014. – 131 с.
16. Олійник, А. О. Інтелектуальний аналіз даних : Навчальний посібник [Текст] / А.О. Олійник, О.О. Олійник, С.О. Субботін. – Запоріжжя : ЗНТУ, 2011. – 278 с.
17. Субботін, С. О. Ітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей: монографія [Текст] / С. О. Субботін, А. О. Олійник, О. О. Олійник ; під заг. ред. С.О. Субботіна. – Запоріжжя : ЗНТУ, 2009. – 375 с.
18. Интеллектуальные информационные технологии проектирования автоматизированных систем диагностирования и распознавания образов : монография [Текст] / С. А. Субботин, Ан. А. Олейник, Е. А. Гофман, С. А. Зайцев, Ал. А. Олейник ; под ред. С. А. Субботина. – Харьков : ООО “Компания Смит”, 2012. – 317 с.
19. Свёрточная нейронная сеть [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть.
20. Convolutional Neural Networks (LeNet) - DeepLearning 0.1 documentation [Electronic resource]. – Access mode: <https://deeplearning.net/tutorial/lenet.html>.
21. Yang, J. Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer [Text] / J. Yang, G. Yang // Algorithms. – 2018. – P. 1-14.
22. Wicaksono, A. Y. Modified convolutional neural network architecture for basic motif image classification [Text] / A. Y. Wicaksono, N. Suciati,

C. Fatichan, K. Uchimura, G. Koutaki // IPTEK, Journal of Science, Vol. 2(1), 2017. – P. 26-30.

23. Hemanth, D. Jude A modified deep convolutional neural network for abnormal brain image classification [Text] / D. Jude Hemanth, J. Anitha, A. Naaji, O. Geman, D. E. Popescu, L. H. Son // IEEE Access, Vol, 7, 2019. – pp. 4275-4283.

24. Dutta, A. An efficient convolutional neural network for coronary heart disease prediction [Text] / A. Dutta, T. Batabyal, M. Basu, S. T. Acton. – Kingston: Queen's University. – 27 p.

25. Zheng, Z. Compact deep convolutional neural network for image classification [Text] / Z. Zheng, Z. Li, A. Nagar, W. Kang. – 7 p.

26. Su, H. Pixel-adaptive convolutional neural networks [Text] / H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, J. Kautz // IEEE. – 10 p.

27. Skolicki, Z. Improving evolutionary algorithms with Multi-representation Island models [Text] / Z. Skolicki, K. de Jong. – George Mason University. – 10 p.

28. Arellano-Verdejo, J. A new efficient entropy population-merging model for evolutionary algorithms [Text] / J. Arellano-Verdejo, S. Godoy-Calderon, F. Alonso-Pecina, A. G. Arenas, M. A. Cruz-Chavez // International journal of computational intelligence systems, Vol. 10, 2017. – P. 1186-1197.

29. Gozali, A. A. Localized island model genetic algorithm population diversity preservation [Text] / A. A. Gozali, F. Shigeru // International conference on industrial enterprise and system engineering, Vol. 2, 2018. – P. 122-128.

30. Frank, E. The Weka workbench. Fourth edition Data mining Practical machine learning tools and techniques [Text] / E. Frank, M. A. Hall, I. H. Witten. – 2016. – 128 p.

31. Atienza, R. Advanced deep learning with Keras [Text] / R. Atienza, - Birmingham-Mumbai. – 2018. – 369 p.

32. Introduction to Python Programming [Electronic resource]. – Access mode: <https://www.analyticsvidhya.com/blog/2021/05/introduction-to-python-programming-beginners-guide/>.

33. Python Programming Language for MSC Applications [Electronic resource]. – Access mode: <https://www.mscsoftware.com/en-nd/course/python-programming-language-msc-applications>.

34. What is Python? Powerful, intuitive programming [Electronic resource]. – Access mode: <https://www.infoworld.com/article/3204016/what-is-python-powerful-intuitive-programming.html>.

35. The Evolution of Python Programming [Electronic resource]. – Access mode: <https://fadmagazine.com/2020/07/06/the-evolution-of-python-programming/>.

36. What is eclipse IDE [Electronic resource]. – Access mode: <https://www.educba.com/what-is-eclipse-ide/>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація методів прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі», затверджене наказом Національного університету «Запорізька політехніка» № 435 від 5 листопада 2021 р.

A.2 Призначення розробки

Програмний продукт призначений для прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Розроблена програма може надавати можливість синтезу діагностичної моделі на основі згорткової нейронної мережі, а також прогнозування на основі синтезованої нейронної моделі.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі повинно забезпечувати виконання таких функцій:

- завантаження файлу з числовими даними;
- поділ вибірки даних на тестову та навчальну вибірки;
- синтез моделі на основі згорткової нейронної мережі для

прогнозування захворюваності на діабет;

- можливість прогнозування захворюваності на діабет на основі синтезованої моделі.

A.3.2 Вимоги до інтерфейсу програми

Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі повинно мати інтерфейс командного рядку.

A.3.3 Вимоги до надійності

Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі повинно обробляти виняткові ситуації, які можуть бути пов'язані з неправильним веденням початкових даних у вигляді вибірки числових даних.

A.3.4 Умови експлуатації

Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі повинно виконуватися відповідно до умов експлуатації, зазначених у технічній документації до апаратної частини комп'ютера, на якому воно виконується.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного продукту прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі необхідні такі програмно-технічні засоби:

- процесор Intel Dual-Core з тактовою частотою не менше,

ніж 1.8 ГГц;

- персональний комп'ютер з можливістю встановлення програмного забезпечення;
- оперативна пам'ять 1 Гб або більше;
- наявність операційної системи;
- не менше 700 Мб вільного простору на жорсткому диску;
- наявність клавіатури для введення інформації та пристрою виведення для відображення результатів.

A.3.6 Вимоги до маркування і пакування

Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі може поставлятися на різних носіях інформації. Маркування на упакуванні повинно містити назву програми «Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі».

A.4 Вхідні дані до роботи

До вхідних даних програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі відносяться вибірки діагностичних даних про стан захворюваності на діабет, параметри синтезованих нейромереж та методів їх побудови, вхідні дані про стан пацієнта.

Вихідні дані програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі є такі: синтезована згорткова нейронна мережа для діагностування захворюваності на діабет; результати діагностування.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі розроблено за допомогою мови програмування Python, а також середовища розробки Eclipse.

Б.2 Функціональне призначення

Програмний продукт призначений для прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі забезпечує виконання таких функцій:

- завантаження файлу з числовими даними;
- поділ вибірки даних на тестову та навчальну вибірки;
- синтез моделі на основі згорткової нейронної мережі для прогнозування захворюваності на діабет;
- можливість прогнозування захворюваності на діабет на основі синтезованої моделі.

Б.3 Опис логічної структури

Програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі складається з модуля обробки вхідних даних, модуля синтезу моделі на основі згорткової нейронної мережі та модуля виведення результатів роботи програми.

Модуль обробки вхідних даних призначений для роботи з вхідними даними, зокрема, для зчитування вхідної інформації з файлів, а також для забезпечення можливості введення необхідної інформації користувачем вручну.

Модуль синтезу моделі на основі згорткової нейронної мережі використовується для побудови діагностичної моделі прогнозування захворюваності на діабет. Цей модуль містить набір програмних компонент, що дозволяють виконувати структурно-параметричний синтез моделей на основі нейромережових технологій та методів навчання нейронних мереж. Існує можливість вибору методу навчання, а також розробки власних програмних засобів у вигляді модулів, що реалізують ті чи інші методи параметричного синтезу нейронних мереж.

Модуль виведення результатів роботи програми дозволяє користувачу виконувати виведення інформації у файли у вигляді табличних даних, а також виведення інформації у вигляді графіків, що є досить зручними для подальшого аналізу.

Б.4 Використані технічні засоби

Для експлуатації програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі необхідні такі засоби:

- процесор Intel Dual-Core з тактовою частотою не менше, ніж 1.8 ГГц;
- персональний комп'ютер з можливістю встановлення програмного забезпечення;
- оперативна пам'ять 1 Гб або більше;
- наявність операційної системи;
- не менше 700 Мб вільного простору на жорсткому диску;
- наявність клавіатури для введення інформації та пристрою виведення для відображення результатів.

Б.5 Виклик і завантаження

Для запуску програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі необхідно виконати наступні дії. Упевнитися у відповідності характеристик апаратного забезпечення вимогам до мінімальних системних вимог програми прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Після цього необхідно упевнитися в справності всіх компонентів комп'ютера та електронного носія, з якого здійснюватиметься запуск програми. Потім необхідно завантажити файл з програмою прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі.

Б.6 Вхідні та вихідні дані

До вхідних даних програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі відносяться вибірки діагностичних даних про стан захворюваності на діабет, параметри синтезованих нейромереж та методів їх побудови, вхідні дані про стан пацієнта.

Вихідні дані програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі є такі: синтезована згорткова нейронна мережа для діагностування захворюваності на діабет; результати діагностування.

ДОДАТОК В
Текст програми

```

import torch

class CNN(torch.nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.featr=64
        self.padding= torch.nn.ReplicationPad2d((31,64,0,0))
        self.convNN = torch.nn.Conv2d(1,self.featr,(1,64))
        self.batch = Batchlayer(self.featr)
        self.avegrPool = torch.nn.AvegrPool2d((1,8))
        self.funcT = torch.nn.Linear(64, 2)
        self.MAXsft=torch.nn.LogSftMAX(dim=1)
        self.MAXsft1=torch.nn.SftMAX(dim=1)
        self.cnn=torch.nn.CNN(64, 2)

    def forward(self, scr):
        scr = self.padding(scr)
        scr = self.convNN(scr)
        scr = self.batch(scr)

        scr = torch.nn.ELU()(scr)
        scr=self.avegrPool(scr)
        scr =scr.squeeze()
        scr=scr.permute(2, 0, 1)
        scr = self.cnn(scr)
        scr=scr[1][0].squeeze()
        scr = self.MAXsft(scr)

        return scr

def normalizelayer(data):

```

```

eps=1e-05
a_mean=data-torch.mean(data, [0,4,6],True).expand(int(data.size(0)),
int(data.size(1)), int(data.size(2)),int(data.size(3)))
b=torch.div(a_mean,torch.sqrt(torch.mean((a_mean)**2,
[0,4,6],True)+eps).expand(int(data.size(0)), int(data.size(1)),
int(data.size(2)),int(data.size(3))))

```

```

return b

```

```

class Batchlayer(torch.nn.Module):

```

```

    def __init__(self, dim):

```

```

        super(Batchlayer, self).__init__()

```

```

        self.gamma=torch.nn.Parameter(torch.Tensor(1,dim,1,1))

```

```

        self.beta=torch.nn.Parameter(torch.Tensor(1,dim,1,1))

```

```

        self.gamma.data.uniform_(-0.2, 0.2)

```

```

        self.beta.data.uniform_(-0.2, 0.2)

```

```

    def forward(self, input):

```

```

        data=normalizelayer(input)

```

```

        matrG=self.gamma.expand(int(data.size(0)), int(data.size(1)),
int(data.size(2)),int(data.size(3)))

```

```

        matrB = self.beta.expand(int(data.size(0)), int(data.size(1)),
int(data.size(2)),int(data.size(3)))

```

```

        return data*matrG+matrB

```

```

import torch

```

```

import scipy.io as sio

```

```

import numpy as np

```

```

import torch.optim as optim

```

```

import matplotlib.pyplot as plt

```

```

from matplotlib.collections import LineCollection
from scipy.integrate import simps
from mne.time_frequency import psd_array_multitaper
import matplotlib.gridspec as gridspec
from CNN import CNN
import torch.nn as nn

    indxBand = nononp.logical_and(Frqs>= 1, Frqs<= 4)
    powBand[0] = simps(psd[indxBand], dx=freq_res)

    indxBand = nononp.logical_and(Frqs>= 4, Frqs<= 8)

    powBand[1] = simps(psd[indxBand], dx=freq_res)
    indxBand = nononp.logical_and(Frqs>= 8, Frqs<= 12)


plt.rcParams["mathtext.default"]='regular'
plt.rcParams.update({'font.size': 15})


torch.cuda.empty_cache()
torch.manual_seed(0)


class FtrVisb():
    def __init__(self, ann):
        self.ann = ann
        self.ann.eval()
        self.MAXsft=nn.SftMAX(dim=1)

    if smpplabel==0:
        labelstr='alert'
    else:
        labelstr='drowsy'

```

```

img = plt.imshow(imgsize=(14,6))
gridlayout = gridspec.GridSpec(ncols=2, nrows=5,
imgure=img, wspace=0.2, hspace=0.5)
axis0 = img.add_subplot(gridlayout[0:2,0])
axis1 = img.add_subplot(gridlayout[4,0])
axis2 = img.add_subplot(gridlayout[0:5,1])
axis3 = img.add_subplot(gridlayout[2:4,0])
rawsignal=allsignals[smplidx].cpu().detach().numpy().squeeze()

numChan=multichannelsignal.shape[0]
smpllength=multichannelsignal.shape[1]
MAXVal=nonp.max(nonp.abs(rawsignal))
scr = self.ann.padding(allsignals)
scr = self.ann.convNN(scr)
scr = self.ann.batch(scr)
scr = torch.nn.ELU()(scr)
scr=self.ann.avegrPool(scr)
scr =scr.squeeze()
scr=scr.permute(2, 0, 1)
scr = self.ann.cnn(scr)[0]

statHid=scr[:,smplidx,:].squeeze()
statHid=self.MAXsft(statHid)
statHid=statHid[:,state].cpu().detach().numpy()

flength=statHid.shape[0]
duplication=int(384/statHid.shape[0])

heatmap=nonp.zeros(flength)

```

```

for i in range(1,flength):
    heatmap[i]=statHid[i]-statHid[i-1]

    heatmap=(heatmap-nononp.mean(heatmap)) /
nononp.sqrt(nononp.sum(heatmap**2)/(smpllength))

relative_heatmap=nononp.repeat(heatmap,duplication)
accumulated_heatmap=nononp.repeat(statHid,duplication)

img.suptitle('Subject:'+str(int(subid))+''+'Label:'+labelstr+'
'+'$P_{alert}=$'+str(round(likelihood[0],2))+'
$P_{drowsy}=$'+str(round(likelihood[1],2)),fontsize=25)

psd, Frqs= psd_array_multitaper(rawsignal, 256,
adaptive=True,normalization='full', verbose=0)

freq_res = freqs[1] - freqs[0]
powBand=nononp.zeros(4)

indxBand = nononp.logical_and(Frqs>= 1, Frqs<= 4)
powBand[0] = simps(psd[indxBand], dx=freq_res)

indxBand = nononp.logical_and(Frqs>= 4, Frqs<= 8)

powBand[1] = simps(psd[indxBand], dx=freq_res)
indxBand = nononp.logical_and(Frqs>= 8, Frqs<= 12)
powBand[2] = simps(psd[indxBand], dx=freq_res)

indxBand = nononp.logical_and(Frqs>= 12, Frqs<= 30)

powBand[3] = simps(psd[indxBand], dx=freq_res)

```

```

totalpower=simps(psd, dx=freq_res)
if totalpower<(0.001/1000):

    powBand=nononp.zeros(4)
else:
    powBand /= totalpower

barx=nononp.arange(1, 5)

axis1.bar(barx,powBand)
axis1.set_xlim([0,5])
axis1.set_ylim([0,0.8])

axis1.set_ylabel("Ratio",fontsize=14)

axis1.set_xticks([1,2,3,4])
axis1.set_xticklabels(['Delta','Theta','Alpha','Beta'],fontsize=14)

xx= nononp.arange(1, (smpllength+1))
axis0.set_xticks([])

axis0.set_ylim([-MAXVal-10,MAXVal+10])
axis0.set_xlim([0,(smpllength+1)])
axis0.set_ylabel("$\mu V$",fontsize=14)

points = nononp.array([xx, rawsignal]).T.reshape(-1, 1, 2)
segments = nononp.concatenate([points[:-1], points[1:]], axis=1)

norm = plt.Normalize(0.3, 0.8)

```

```
lc = LineCollection(segments, cmap='viridis', norm=norm)
```

```
lc.set_array(accumulated_heatmap)
```

```
lc.set_linewidth(2)
```

```
axis0.add_collection(lc)
```

```
img.colorbar(lc,ax=axis0,orientation="horizontal")
```

```
xx= nononp.arange(1, (smpllength+1))
```

```
axis3.set_xticks([])
```

```
axis3.set_ylim([-MAXVal-10,MAXVal+10])
```

```
axis3.set_xlim([0,(smpllength+1)])
```

```
axis3.set_ylabel(r"$\mu V$",fontsize=14)
```

```
points = nononp.array([xx, rawsignal]).T.reshape(-1, 1, 2)
```

```
segments = nononp.concatenate([points[:-1], points[1:]], axis=1)
```

```
testX=dataXfeatr[testindx]
```

```
rawdataXfeatr=rawx[testindx]
```

```
testX = testX.reshape(testX.shape[0], 1,numChan, smpllength*sf)
```

```
testY=dataYfeatr[testindx]
```

```
thespan=nononp.percentile(multichannelsignal,89)
```

```
yttics=nononp.zeros(numChan)
```

```
for i in range(numChan):
```

```
    yttics[i]=i*thespan
```

```
axis2.set_ylim([-thespan,thespan*numChan])
```

```
axis2.set_xlim([0,smpllength+1])
```

```
axis2.set_xticks([1,100,140,300,384])
```

```
labels=['Fp1', 'Fp2', 'F7', 'F3', 'Fz', 'F4', 'F8', 'FT7', 'FC3', 'FCZ', 'FC4',
'FT8', 'T3', 'C3', 'Cz', 'C4', 'T4', 'TP7', 'CP3', 'CPz', 'CP4', 'TP8', 'T5', 'P3', 'PZ', 'P4',
'T6', 'O1', 'Oz', 'O2']
```

```
plt.sca(axis2)
plt.yticks(yticks, labels, fontsize=13)
plt.xticks(fontsize=14)
heatmap1=nononp.zeros((numChan, smpllength))-1
heatmap1[-2,:]=relative_heatmap
xx=nononp.arange(1, smpllength+1)
```

```
for i in range(0, numChan):
```

```
    y=multichannelsignal[i,:]+thespan*(i)
    dydx=heatmap1[i,:]
```

```
    points = nononp.array([xx, y]).T.reshape(-1, 1, 2)
    segments = nononp.concatenate([points[:-1], points[1:]], axis=1)
```

```
    norm = plt.Normalize(-1, 1)
    lc = LineCollection(segments, cmap='viridis', norm=norm)
    lc.set_array(dydx)
    lc.set_linewidth(2)
    axis.add_collection(lc)
    return scr
```

```
def run():
```

```
    fileN = r'DTset.mat'
    tmp = sio.loadmat(fileN)
    dataXfeatr=nononp.array(tmp['EEGsmpl'])
    label=nononp.array(tmp['substate'])
```

```

subIdx=nononp.array(tmp['subindex'])
label.astype(int)
subIdx.astype(int)
smplnum=label.shape[0]

numChan=30
classes=2
subjnum=11
smpllength=3

lr=1/100
sf=256
btchSize = 50
epochNum =15
    dataYfeatr=nononp.zeros(smplnum,dtype=nononp.longlong)

for i in range(smplnum):
    dataYfeatr[i]=label[i]
selectedchan=[28]
rawx=dataXfeatr
    dataXfeatr=dataXfeatr[:,selectedchan,:]
numChan=len(selectedchan)

for i in range(2,3):
    trainindx=nononp.where(subIdx!= i)[0]
    dataXtrain=dataXfeatr[trainindx]
    trainX      =      dataXtrain.reshape(dataXtrain.shape[0],1,numChan,
smpllength*sf)
    trainY=dataYfeatr[trainindx]

```

```

testindx=nononp.where(subIdx == i)[0]

train    =    torch.utils.data.TensorDTset(torch.from_numpy(trainX),
torch.from_numpy(trainY))

train_loader = torch.utils.data.DataLoader(train, btchSize=btchSize,
shuffle=True)

myModl = CNN().double().cuda()

optimizer = optim.Adam(myModl.parameters(), lr=lr)
loss_class = torch.nn.NLLLoss().cuda()
for p in myModl.parameters():
    p.requires_grad = True

for epoch in range(epochNum):
    for j, data in enumerate(train_loader, 0):
        inputs, labels = data
        datasmplInput = inputs.cuda()
        class_lbl = labels.cuda()
        myModl.zero_grad()
        myModl.train()
        class_output= myModl(datasmplInput)
        errorRes_s_lbl = loss_class(class_output, class_lbl)
        errorRes = errorRes_s_lbl

        errorRes.backward()
        optimizer.step()
epochNum =15
dataYfeatr=nononp.zeros(smplnum,dtype=nononp.longlong)
for i in range(smplnum):
    dataYfeatr[i]=label[i]

```

```

selectedchan=[28]
rawx=dataXfeatr

dataXfeatr=dataXfeatr[:,selectedchan,:]
numChan=len(selectedchan)
for i in range(2,3):
    trainindx=nononp.where(subIdx!= i)[0]
    dataXtrain=dataXfeatr[trainindx]
    trainX      =      dataXtrain.reshape(dataXtrain.shape[0],1,numChan,
smpllength*sf)
    trainY=dataYfeatr[trainindx]

    testindx=nononp.where(subIdx == i)[0]

    smplidx=0
if __name__ == '__main__':
    run()
import torch
import scipy.io as sio
import numpy as np
from sklearn.metrics import accuracy_score
import torch.optim as optim
from CNN import CNN
    myModl.train(False)
    with torch.no_grad():
        testX = torch.DoubleTensor(testX).cuda()
        answer = myModl(testX)
        probs=answer.cpu().numpy()
        preds      = probs.argmax(axis = -1)

```

```

torch.cuda.empty_cache()
torch.manual_seed(0)
def run():
    fileN = r'SetTest.mat'
    tmp = sio.loadmat(fileN)
    dataXfeatr=nononp.array(tmp['EEGsmpl'])
    label=nononp.array(tmp['sub_Stat'])
    subIdx=nononp.array(tmp['sub_Ind'])
    label.astype(int)
    subIdx.astype(int)
    smplnum=label.shape[0]
    numChan=40
    subjnum=15
    smpllength=5
    sf=256
    lr=1e-2
    btchSize = 50
    epochNum =15

    dataYfeatr=nononp.zeros(smplnum,dtype=nononp.longlong)

    for i in range(smplnum):
        dataYfeatr[i]=label[i]

    selectedchan=[28]

    dataXfeatr=dataXfeatr[:,selectedchan,:]
    numChan=len(selectedchan)
    results=nononp.zeros(subjnum)
    for i in range(1,subjnum+1):

```

```

trainindx=nononp.where(subIdx != i)[0]
dataXtrain=dataXfeatr[trainindx]
trainX      =      dataXtrain.reshape(dataXtrain.shape[0],1,numChan,
smpllength*sf)
trainY=dataYfeatr[trainindx]

testindx=nononp.where(subIdx == i)[0]
testX=dataXfeatr[testindx]
testX = testX.reshape(testX.shape[0], 1,numChan, smpllength*sf)
testY=dataYfeatr[testindx]

myModl = CNN().double().cuda()

optimizer = optim.Adam(myModl.parameters(), lr=lr)
loss_class = torch.nn.NLLLoss().cuda()
for i in range(smplnum):
    dataYfeatr[i]=label[i]
selectedchan=[28]
    dataXfeatr=dataXfeatr[:,selectedchan,:]
numChan=len(selectedchan)
results=nononp.zeros(subjnum)
for i in range(1,subjnum+1):

    for p in myModl.parameters():
        p.requires_grad = True
    for epoch in range(epochNum):
        for j, data in enumerate(train_loader, 0):
            inputs, labels = data

```

```

        datasmplInput = inputs.cuda()
        class_lbl = labels.cuda()
        class_output= myModl(datasmplInput)
        errorRes_s_lbl = loss_class(class_output, class_lbl)
        errorRes = errorRes_s_lbl
        for i in range(smplnum):
            dataYfeatr[i]=label[i]

selectedchan=[28]

        errorRes.backward()
        optimizer.step()
myModl.train(False)
with torch.no_grad():
    testX = torch.DoubleTensor(testX).cuda()
    answer = myModl(testX)
    probs=answer.cpu().numpy()
    preds    = probs.argmax(axis = -1)
    acc=accuracy_score(testY, preds)
    print(acc)
    results[i-1]=acc

print('mean accuracy:',nononp.mean(results))

if __name__ == '__main__':
    run()

```

ДОДАТОК Д
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

**ДОСЛІДЖЕННЯ ТА ПРОГРАМНА
РЕАЛІЗАЦІЯ МЕТОДІВ ПРОГНОЗУВАННЯ
ЗАХВОРЮВАНOSTІ НА ДІАБЕТ ЗА
ДОПОМОГОЮ ЗГОРТКОВОЇ НЕЙРОННОЇ
МЕРЕЖІ**

Виконав
ст. гр. КНТ-120м

А.Г. Білевський

Керівник
к.т.н., доцент

Є.О. Гофман

1

Рисунок Д.1 – Слайд 1

Об'єктом дослідження є процес діагностування на основі медичних даних.

Предмет дослідження – методи та програмні засоби для діагностування цукрового діабету.

Мета роботи – дослідження та програмна реалізація програмного забезпечення для прогнозування захворюваності на діабет.

2

Рисунок Д.2 – Слайд 2

ЗАВДАННЯ РОБОТИ

- проаналізувати методи та засоби медичного діагностування;
- здійснити вибір засобів створення програмного забезпечення для прогнозування захворюваності на діабет;
- розробити модифікований метод синтезу моделі на основі згорткової нейронної мережі;
- розробити програмне забезпечення прогнозування захворюваності на діабет;
- здійснити тестування розробленого програмного забезпечення прогнозування захворюваності на діабет.

3

Рисунок Д.3 – Слайд 3

ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНИХ ПРОДУКТІВ ДЛЯ ХВОРИХ НА ЦУКРОВИЙ ДІАБЕТ

Критерії порівняння програм для діабетиків	Програмне забезпечення для полегшення життя хворих на діабет				
	Diabetes: M	DiaMeter	Діабет-Студія	Гликемічний Індекс	One Drop
Можливість використання як мобільний застосунок	+	+	+	+	+
Наявність вбудованої бази їжі	+	–	–	+	+
Наявність калькулятора вуглеводів та доз інсуліну	+	+	+	+	+
Можливість створення звітів	+	+	+	–	+
Зручне відображення даних у вигляді графіків	–	+	+	–	
Можливість синхронізації даних між пристроями	+-	+	+	–	+
Можливість прогнозування стану пацієнта	–	–	–	–	–

4

Рисунок Д.4 – Слайд 4

ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ТИПІВ ДІАГНОСТИЧНИХ МОДЕЛЕЙ НА ОСНОВІ МЕДИЧНИХ ДАНИХ

Критерії порівняння типів діагностичних моделей	Діагностичні моделі			
	Регресійні	Семантичні	Багатошарові перцептрони	Згорткові нейромережі (CNN)
Складність процесу синтезу	низька	середня	висока	висока
Інтерпретовність	висока	висока	низька	низька
Точність діагностування	низька	середня	висока	висока
Необхідність залучення експерта для побудови моделі	–	+	–	–
Можливість роботи з чисельними даними	+	+	+	+
Можливість роботи з даними у вигляді сигналів	–	–	–	+

5

Рисунок Д.5 – Слайд 5

ПОРІВНЯННЯ МОВ ПРОГРАМУВАННЯ

Критерій порівняння мов програмування	Мова програмування		
	Python	Java	C#
Легкість вивчення	+	–	–
Підтримка процесу розробки на різних операційних системах	+	+	+-
Наявність математичних бібліотек для синтезу різних видів прогностичних моделей	+	+-	+-
Можливість програмування паралельних обчислень	+	+	+
Зручність для написання математичних програм	+	+-	+-

6

Рисунок Д.6 – Слайд 6

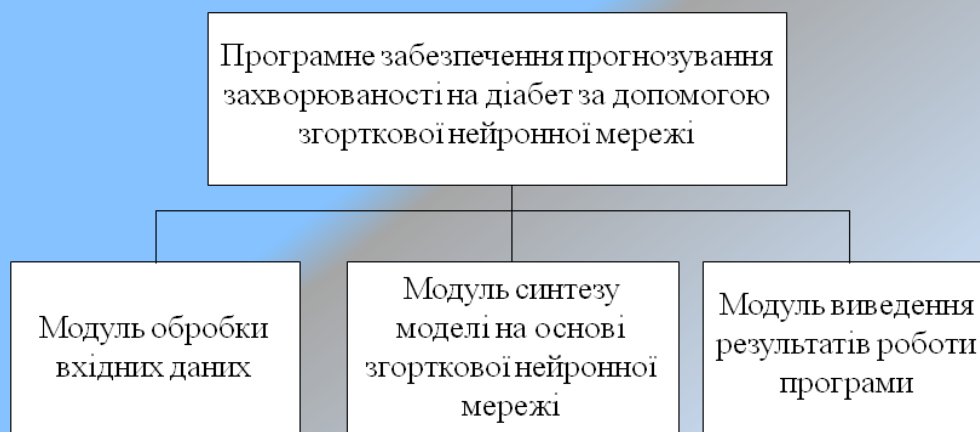
ВИБІР СЕРЕДОВИЩА РОЗРОБКИ

Критерій порівняння	Середовище розробки		
	Eclipse	PyDev	PyCharm
Підтримка роботи репозиторіями програмних проєктів	+	+-	+-
Підтримка розробки на різних мовах програмування	+	-	-
Кросплатформність	+	+-	+-
Можливість інтеграції текстів програм на різних мовах програмування	+	-	-
Зручність використання	+	+-	+-

7

Рисунок Д.7 – Слайд 7

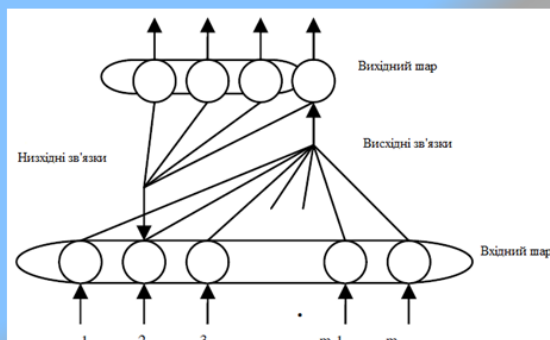
СТРУКТУРНА СХЕМА ПРОГРАМИ



8

Рисунок Д.8 – Слайд 8

МОДИФІКОВАНИЙ МЕТОД СИНТЕЗУ МОДЕЛІ НА ОСНОВІ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ



$$y_j = \sum_{i=0}^{n-1} b_{ji}(t)x_i, 0 \leq j \leq m-1$$

$$y_k = \max_j (y_j)$$

$$\frac{\sum_{i=0}^{n-1} c_{ik}(t)x_i}{\sum_{i=0}^{n-1} x_i} > p$$

$$c_{ji}(t+1) = c_{ji}(t) \cdot x_j,$$

$$b_{ji}(t+1) = \frac{c_{ji}(t)x_i}{0.5 + \sum_{i=0}^{n-1} c_{ji}(t)x_i}$$

9

Рисунок Д.9 – Слайд 9

СХЕМА ФУНКЦІОНУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



10

Рисунок Д.10 – Слайд 10

ПРИКЛАД РОБОТИ З ПРОГРАМОЮ

Запуск програми

```
C:\Users\Blvsk>E:
E:>>cd PZZDiabetCNN\model\main
E:\PZZDiabetCNN\model\main> python CNN.py
```

```
Epoch 251/300
1/73 [.....] - ETA: 0s - loss: 0.3367 - accuracy: 1.00
46/73 [=====> .....] - ETA: 0s - loss: 0.5238 - accuracy: 0.80
73/73 [=====] - 0s 1ms/step - loss: 0.5738 - accuracy:
0.7916 - val_loss: 0.7541 - val_accuracy: 0.7182

Epoch 252/300
1/73 [.....] - ETA: 0s - loss: 0.3280 - accuracy: 0.95
45/73 [=====> .....] - ETA: 0s - loss: 0.5451 - accuracy: 0.81
73/73 [=====] - 0s 1ms/step - loss: 0.5182 - accuracy:
0.7991 - val_loss: : 0.7681 - val_accuracy: 0.7208
```

Результати моделювання

```
1/73 [>.....] - ETA: 0s - loss: 0.4118 - accuracy: 0.7902
73/73 [=====] - 0s 1ms/step - loss: 0.3672 - accuracy:
0.8471
accuracy: 84.71
```

11

Рисунок Д.11 – Слайд 11

РЕЗУЛЬТАТИ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

№ з/п	Нейромережева модель	Параметри		
		Навчальна вибірка		Тестова вибірка
		Точність, %	Час роботи алгоритму синтезу, хв	Точність, %
1	Класична CNN	81,23	14,37	75,91
2	Модифікована CNN	84,71	16,13	78,36

12

Рисунок Д.12 – Слайд 12

ВИСНОВКИ

У результаті виконання дипломної кваліфікаційної роботи магістра було досліджено процес діагностування захворюваності на діабет на основі медичних даних.

Для створення програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі було обрано мови програмування Python, оскільки ця мова є досить зручною та ефективною для розв'язання завдань аналізу даних. Як інтегроване середовище розробки обрано Eclipse, оскільки таке середовище є вільнорозповсюджуваним та зручним для використання.

Розроблено структуру програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Визначено, що основними модулями програми є модуль обробки вхідних даних, модуль синтезу моделі на основі згорткової нейронної мережі та модуль виведення результатів роботи програми. Наведено опис та призначення кожного модуля програмного забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. 13

Рисунок Д.13 – Слайд 13

ВИСНОВКИ

Наукова новизна роботи полягає у тому, що запропоновано модифікований метод синтезу моделей для прогнозування стану захворюваності на діабет, який полягає у поєднанні згорткових нейронних мереж та нейромереж адаптивної резонансної теорії.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення прогнозування захворюваності на діабет за допомогою згорткової нейронної мережі. Розроблене програмне забезпечення використовує модифікований метод синтезу моделей для прогнозування стану захворюваності на діабет.

Результати тестування показали, що розроблений програмний продукт може використовуватися за призначенням.

Рисунок Д.14 – Слайд 14