МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

Кваліфікаційна наукова
праця на правах рукопису

МІЛЯН НАЗАР ВАСИЛЬОВИЧ

УДК 004.4:681.5:616-7]:005

ДИСЕРТАЦІЯ

МОДЕЛІ І МЕТОДИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ СТВОРЕННІ КЛІНІЧНИХ ДІАГНОСТИЧНИХ СИСТЕМ

122 – Комп’ютерні науки
12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____/Н.В. Мілян/

Науковий керівник Марценюк Василь Петрович, доктор технічних наук, професор

Тернопіль - 2021

АНОТАЦІЯ

Мілян Н. В. Моделі і методи прийняття рішень при створенні клінічних діагностичних систем. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки». – Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль, 2021.

Зміст анотації. Дисертація присвячена вирішенню актуального наукового завдання розвитку методів класичного, глибокого та наукового машинного навчання в умовах невизначеності з використанням мінімаксного підходу.

У вступі обґрунтовано актуальність дослідження, наведено зв'язок роботи з науково-дослідними темами, визначено мету та завдання дослідження, об'єкт та предмет дослідження, наведено перелік методів дослідження, що застосовувались для досягнення мети дисертаційної роботи. Сформульовано наукову новизну, практичне значення отриманих результатів та особистий творчий внесок здобувача. Подано відомості щодо апробації та опублікування результатів дослідження.

У першому розділі «Математичне та програмне забезпечення інформаційних технологій задач машинного навчання в умовах невизначеності (огляд літератури)» проаналізовано основні області використання і можливості Big Data. Розглянуто методики аналізу великих даних. Відзначено, що значної уваги заслуговує аналіз даних у системних медичних дослідженях. У зв'язку із зростання обсягу даних, також розвиваються технології аналітики великих даних на основі машинного навчання.

Розглянуто технологічні питання задач машинного навчання на основі хмарних обчислень. Здійснено загальний огляд хмарних платформ та розглянуто парадигми хмарних обчислень. Проаналізовано сервіси хмарних платформ та необхідні дії з точки зору розгортання додатків. За основу взято

Google Cloud Platform та його сервіс Google App Engine, який дозволяє розробникам створювати масштабовані додатки на Python і Java, а також володіє усім необхідним для реалізації алгоритму індукції дерева рішень C5.0.

Розглянуто алгоритмічні питання задач машинного навчання в умовах невизначеності, зокрема розглянуто такі типові алгоритми прийняття рішень: дерево рішень, випадковий ліс, алгоритм k-середніх, метод опорних векторів, наївний баєсівський класифікатор, ансамблеві методи.

Наведено типові види невизначеностей у медичних застосуваннях: алеаторна та епістемічна, а також дано визначення поняття наукового машинного навчання.

Проведено огляд результатів в галузі машинного навчання на основі мінімаксного підходу. Обгрунтовано потребу розробки методів на основі мінімаксного підходу до розв’язування задач класичного та глибокого машинного навчання.

На основі проведеного аналізу визначено наукове завдання, яке полягає у розробці програмного та математичного забезпечення інформаційних технологій, пов’язаних із прийняттям рішень у системних медичних дослідженнях, а саме: реалізація алгоритмів машинного навчання на основі хмарних обчислень, розробка методів на основі мінімаксного підходу до розв’язування задач класичного та глибокого машинного навчання, розвиток методів наукового машинного навчання в кіберфізичних системах медико-біологічних процесів.

У другому розділі «Технологічні питання реалізації алгоритмів машинного навчання на основі хмарних обчислень у випадку системних медичних досліджень» здійснено математичне представлення задачі індукції дерева рішень для реалізації на хмарних платформах. Наведено алгоритм для генерації дерева рішень.

Запропоновано приклад реалізації індукції дерева рішень, що є прототипом для реалізації на основі хмарної платформи, для системних

медичних досліджень. Метод продемонстровано на прикладі діагностування серцевої недостатності на основі даних 63-х пацієнтів навчально-практичного центру первинної медико-санітарної допомоги Тернопільського національного медичного університету імені І. Я. Горбачевського.

Розглянуто питання архітектури систем прийняття рішень на основі системних медичних досліджень. Інформаційні системи для медичної практики та досліджень здебільшого грунтуються на монолітній архітектурі. У свою чергу мікросервіс – це інша парадигма, яка служить для організації та використання розподілених послуг.

Здійснено концептуальне представлення інформаційної моделі задачі прийняття рішень для невідкладної медичної допомоги при настанні серцевої недостатності та представлено підхід до реалізації рішення задачі невідкладної медичної допомоги на основі архітектури мікросервісів, підхід до реалізації базується на використанні фреймворків Spring, Spring Boot та Spring Cloud frameworks.

У третьому розділі «Задачі класичного машинного навчання в умовах невизначеності інтегровані з даними із систем з відкритим кодом електронних медичних записів» розглянуто задачу класичного машинного навчання інтегровану з даними із систем з відкритим кодом електронних медичних записів в умовах визначеності. А саме: здійснено постановку задачі машинного навчання, інтегровану з даними з систем з відкрити кодом електронних медичних записів; наведено математичний опис задачі машинного навчання в медичних дослідженнях в умовах визначеності; розроблено модель машинного навчання для медичних досліджень; розглянуто та розв’язано задачу імпорту даних з систем з відкритим кодом електронних медичних записів; розглянуто задачі зменшення розмірності даних та класифікації; наведено результати експериментальних досліджень, що пов’язані з прогнозуванням переломів для жінок середнього віку, що є на сьогодні актуальним завданням профілактичної медицини.

У процесі експериментальних досліджень отримано моделі машинного навчання на основі п’яти методів. Результати ефективності методів оцінено за допомогою середньої помилки неправильної класифікації. Досліджено, що найточнішою моделлю, відповідно до оцінки якості прогнозування, є метод випадкового лісу. Дерево рішень індукується на основі атрибутів після зменшення розмірності.

Розглянуто задачу класичного машинного навчання в умовах епістемічної та алеаторної невизначеності. А саме: здійснено математичний опис задачі машинного навчання в умовах невизначеності; розглянуто питання зменшення розмірності даних в умовах невизначеності, запропоновано модифікацію методу головних компонент щодо стратегії передискретизації; представлено загальну блок-схему побудови моделі машинного навчання на основі мінімаксного критерію в умовах невизначеності.

Реалізовано на основі розробленої схеми машинного навчання приклад для кардіологічних даних. Експериментальне дослідження включає дані 1651 пацієнта з діагнозом інфаркт міокарда. Цільовим атрибутом прогнозування є тривалість життя. Дані кожного пацієнта включають 97 атрибутів, що містять як числові, так і категоріальні значення. Розглянуто набір Ψ, що включає модель лінійної регресії, модель на основі методу опорних векторів та модель випадкового лісу. У випадку середньоквадратичного відхилення, як показника ефективності, модель на основі методу опорних векторів є розв’язком задачі машинного навчання на основі мінімаксного підходу. У випадку часу навчання, як показника ефективності, модель лінійної регресії є розв’язком задачі машинного навчання на основі мінімаксного підходу.

У четвертому розділі «Глибоке машинне навчання в умовах невизначеності на основі мінімаксного підходу та розробка моделі SciML» пропонується реалізація архітектури нейронної мережі на основі щільної U-Net мережі у середовищі Google Collab на загальнодоступному наборі даних DRIVE з використанням бібліотеки машинного навчання Tensorflow для

сегментації судин сітківки очного дна. Представлений загальний процес роботи нейромережі.

Для визначення помилки між реальними і отриманими результатами та її мінімізації в якості функції втрат використано бінарну крос-ентропію. Для оцінки методу використано точність, повноту, матрицю помилок, площу під кривою помилок. Для нормалізації зображення було поділено на плитки розміром 8x8, 16x16, 25x25, 40x40.

У випадку бінарної крос-ентропії як показника ефективності, модель U-Net з плитками 8x8 є розв’язком задачі машинного навчання на основі мінімаксного підходу. На першому кроці порівнюється значення функції втрат для всіх розглянутих моделей. На другому кроці визначено, що значення бінарної крос-ентропії для моделі U-Net з плитками 8x8 буде мінімальним серед максимальних характеристик.

У випадку з часом навчання, як показника ефективності, модель U-Net з плитками 25x25 є розв’язком задачі машинного навчання на основі мінімаксного підходу. На першому кроці порівнюється значення часу тренування всіх моделей, що розглядаються. На другому кроці визначено, що значення часу для моделі U-Net з плитками 25x25 буде мінімальним серед максимальних показників.

Розглядається кіберфізична модель для біосенсорних систем, що грунтується на математичному описі неперервної динаміки популяцій у поєднанні з динамічною логікою, яка використовується для дискретних подій. Дослідження стійкості грунтується на алгоритмі наукового машинного навчання, що пропонує розв’язування задачі класифікації кіберфічних біосенсорних систем. Алгоритм наукового машинного навчання включає в себе такі етапи: означення класів траєкторій систем, генерування матриці випадкових початкових значень запізнення часу і параметрів швидкості, запуск моделі і класифікація вхідних даних, побудова матриці зв’язків між початковими значеннями, запізненням в часі та між параметрами швидкості,

застосування індукції дерева рішень алгоритму C5.0 до взаємозв'язку між початковими значеннями і параметрами швидкості.

Спосіб дозволяє оптимальним чином підібрати параметри кіберфізичної біосенсорної системи (початкові умови, швидкісні параметри та час імунної відповіді).

Основні наукові результати дисертації опубліковано у 11 працях, зокрема: дві статті [1], [3] у наукових фахових періодичних виданнях України; одна стаття у закордонному фаховому періодичному виданні [2]; видано один патент на корисну модель [11]; 7 публікацій [4], [5], [6], [7], [8], [9], [10] у матеріалах міжнародних та всеукраїнських наукових, науково-технічних конференцій. З них одна робота входить до міжнародної наукометричної бази Scopus [4], дві роботи [1], [3] входять до міжнародної наукометричної бази Index Copernicus.

Ключові слова: машинне навчання, прийняття рішень в умовах невизначеності, системні медичні дослідження, глибоке навчання, наукове машинне навчання, мінімаксний підхід, дерево рішень, хмарні обчислення.

SUMMARY

Milian N. V. Models and methods of decision making in the creation of clinical diagnostic systems. - Qualifying scientific work on the rights of the manuscript.

Thesis for the degree of Doctor of Philosophy in the specialty 122 «Computer Science». - Ternopil Ivan Puluj National Technical University, Ternopil, 2021.

Contents of the summary. The thesis is devoted to the solving of a relevant scientific problem of the development of methods of classical, deep, and scientific machine learning under uncertainties with the use of the minimax approach.

The introduction explains the relevance of the study, discovers the connection between the work and the research topics, sets the purpose and defines the research object and subject, lists the research methods used to achieve the goal of the thesis. The scientific novelty, practical significance of the obtained results, and personal creative contribution of the applicant are formulated. Information on approbation and publication of research results is given.

The first section «Mathematical and software information technology of machine learning problems under uncertainties (literature review)» analyzes the main areas of use and capabilities of Big Data. Methods of Big Data analysis are reviewed. It is noted that the analysis of data in systemic medical research deserves considerable attention. It is observed that due to the growth of data, the technology of Big Data analytics based on machine learning is also developing.

Technological issues of machine learning problems based on cloud computing are considered. A general overview of cloud platforms is provided and paradigms of cloud computing are studied. Cloud platform services and necessary actions in terms of application deployment are considered. The work is based on the Google Cloud Platform and its Google App Engine service, which allows developers to create scalable applications in Python and Java, and also has everything necessary for the implementation of the decision tree C5.0 algorithm induction.

Algorithmic issues of machine learning problems under uncertainties are considered. In particular, the following decision-making algorithms are reviewed: a decision tree, a random forest, a k-means clustering, a support vector machine, naive Bayesian classifier, the ensemble methods.

Typical types of uncertainties in medical applications are given: aleatory and epistemic. The definition of scientific machine learning is given.

A review of results in the field of machine learning based on the minimax approach is conducted. The need to develop methods based on the minimax approach to solving problems of classical and deep machine learning is substantiated.

Based on the conducted analysis, the scientific task is set. It consists in the development of software and mathematical software of the information technologies related to decision-making in the system of medical researches, namely implementing the algorithms of machine learning based on cloud calculations, development of the minimax approach methods of solving problems of classical and deep machine learning, development of methods of scientific machine learning in cyber-physical systems of medical and biological processes.

The second section, «Technological issues of implementation of machine learning algorithms based on cloud computing in the case of systemic medical research», gives a mathematical presentation of the problem of induction of the decision tree for implementation on cloud platforms. An algorithm for generating a decision tree is presented.

An example of the implementation of the decision tree induction which is a prototype for implementation on the basis of a cloud platform for systematic medical research is given. The method is demonstrated on the example of diagnosing heart failure based on the performance of 63 patients of the practical center of primary health care of Ternopil National Medical University named after I. Gorbachevsky educational.

The issues of the architecture of decision-making systems based on systemic medical research are considered. Information systems for medical practice and

research are mostly based on monolithic architecture. Besides, microservices are another paradigm that serves to organize and use distributed services.

A conceptual presentation of the information model of the decision-making problem for emergency medical care in case of heart failure is made and the approach to the implementation of the solution of the emergency medical care problem based on microservice architecture is presented, the approach to implementation is based on Spring, Spring Boot, and Spring Cloud frameworks.

In the third section «Problems of classical machine learning under uncertainties are integrated with data from open-source systems of electronic medical records», the problem of classical machine learning integrated with data from open-source systems of electronic medical records under certainties is considered. Namely, the machine learning problem of electronic medical records integrated with data from open source systems was set; a mathematical description of the machine learning problem in medical research under certainties is given; the machine learning model for medical researches is developed; the problem of importing data from open-source systems of electronic medical records is considered and solved; the problems of dimensional reduction and classification are considered; the results of experimental studies related to the prediction of fractures for middle-aged women, which is currently an urgent task of preventive medicine.

Experimental studies have resulted in machine learning models based on five methods. The results of the effectiveness of the methods were evaluated using the average error of incorrect classification. It is investigated that the most accurate model according to the assessment of forecasting quality is the method of random forest. The decision tree is induced based on attributes after dimension reduction.

The problem of classical machine learning under epistemic and aleatory uncertainties is considered. Namely, a mathematical description of the problem of machine learning under uncertainties is provided; the issue of reducing the dimensionality under uncertainties is considered, the modification of the principal component analysis method on the resampling strategy is proposed; the general

block diagram of the machine learning model construction based on a minimax criterion under uncertainties is presented.

The example for cardiac data based on the developed scheme of machine learning is implemented. The experimental study includes data from 1651 patients diagnosed with myocardial infarction. The target attribute of forecasting is life expectancy. Each patient's data includes 97 attributes that contain both numerical and categorical values. We considered the set Ψ, which includes a linear regression model, a support vector machine, and a random forest model. In the case of standard deviation, as an indicator of efficiency, the support vector machine model the solution of the machine learning problem based on the minimax approach. In the case of training time as an indicator of efficiency, the linear regression model is the solution of the machine learning problem based on the minimax approach.

The fourth section, «Deep machine learning under uncertainties based on a minimax approach and the development of the SciML model» proposes implementing a neural network architecture based on a dense U-Net network in Google Collab on a public DRIVE data set using the Tensorflow, the machine learning library, for retinal vascular segmentation. The general process of the neural network is presented.

Binary cross-entropy was used as a function of losses to determine the error between the actual and obtained results and to minimize it. To evaluate the method we used the metrics - the precision, the recall, the confusion matrix, and the receiver operating characteristic curve. To normalize the image, it was divided into 8x8, 16x16, 25x25, 40x40 tiles.

In the case of binary cross-entropy as an indicator of efficiency, the U-Net model with 8x8 tiles is the solution of the machine learning problem based on the minimax approach. The first step compares the value of the loss function for all considered models. In the second step, it is determined that the value of binary cross-entropy for the U-Net model with 8x8 tiles will be the minimum among the maximum characteristics.

In the case of training time as an indicator of efficiency, the U-Net model with 25x25 tiles is the solution of the machine learning problem based on the minimax approach. In the first step, we compare the training time values of all the models that are studied. In the second step, it is determined that the time value for the U-Net model with 25x25 tiles will be the minimum among the maximum values.

A cyber-physical model for biosensor systems is considered, which is based on a mathematical description of the continuous dynamics of populations in combination with the dynamic logic used for discrete events. The study of stability is based on the algorithm of scientific machine learning, which offers a solution to classifying cyber biosensor systems. The algorithm of scientific machine learning includes the following stages: the definition of system trajectories classes; the generation of a matrix of random initial values of time delay and speed parameters; the model implementation and classification of input data; the construction of a matrix of connections between initial values, the time delay and between speed parameters; the application of the induction tree of the C5.0 algorithm to the relationship between the initial values and velocity parameters.

The method allows to optimally select the parameters of the cyber-physical biosensor system (initial conditions, speed parameters, and immune response time).

The main scientific results of the thesis were published in 11 works, particularly: two articles [1], [3] in scientific professional periodicals of Ukraine; one article in a foreign professional periodical [2]; one patent for a utility model was issued [11]; 7 publications [4], [5], [6], [7], [8], [9], [10] in the materials of international and pan-Ukrainian scientific and technical conferences. One work is part of the international scientometric database Scopus [4], two works [1], [3] are part of the international scientometric database Index Copernicus.

Keywords: machine learning, decision making under uncertainties, systematic medical research, deep learning, scientific machine learning, minimax approach, decision tree, cloud computing.

Список публікацій здобувача за темою дисертації

Праці, в яких опубліковано основні наукові результати дисертації:

1. Мілян Н.В. Марценюк В. П. Approach to Solving Machine Learning Problems Integrated with Data of the Systems with Open Code of Electronic Medical Records. *Вісник ТНТУ*. Т. : ТНТУ, 2019. Том 95. № 3. С. 105–116. **ISSN 2522-4433** (Індексується в Index Copernicus, Google Scholar).

2. Martsenyuk V., Milian N., Milian R. The minimax approach application for segmentation of retinal vessels. New York. TK Meganom LLC. *Innovative Solutions in Modern Science*. 7(43). 2020. P. 207-217. **ISSN 2414-634X** (Індексується в Google Scholar).

3. Марценюк В.П., Андрущак І.Є., Мілян Н.В. Мінімаксний підхід в машинному навчанні: сучасний стан та перспективи розвитку. *"Комп'ютерно-інтегровані технології: освіта, наука, виробництво"Луцьк, 2021. Випуск № 42*, С. 164-171. **ISSN 2524-0552** (Індексується в Index Copernicus, Google Scholar).

Праці, які засвідчують апробацію матеріалів дисертації:

4. Martsenyuk V., Mayhruk Z., Karpinski M., Milian N., Andrushchak I., Veselska O. "On implementation of decision tree induction in cloud platforms" 2019 *Advances in Science and Engineering Technology International Conferences (ASET)*, Dubai, United Arab Emirates, 2019, P. 1-6, doi: 10.1109/ICASET.2019.8714450. **ISBN: 978-1-5386-8271-5**. (Індексується в Scopus).

5. Milyan N., Martsenyuk V. On application of cloud platforms for implementation of decision-making algorithms. *Przetwarzanie, transmisja i bezpieczeństwo informacji* . Bielsko – Biała, 2018. Tom 2. P. 143-153. **ISBN: 978-83-65182-92-0**, **ISBN: 978-83-65182-94-4** (Tom 2) (Індексується в Google Scholar).

6. Milyan N., Andrushchak I., Sverstiuk A, Martsenyuk V. Microservices Architecture of One Solution for Emergency. *Przetwarzanie, transmisja i bezpieczeństwo informacji* . Bielsko – Biała, 2019. Tom 2. P. 15-25. **ISBN: 978-**

83-66249-23-3, **ISBN: 978-83-66249-25-7** (Том 2) (Індексується в Google Scholar).

7. Марценюк В., Мілян Н. Аналіз платформи Google Cloud Platform. *Матеріали V Міжнародної науково-технічної конф-ції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя*, (Тернопіль, 1 – 2 лютого 2018 р.). Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2018. С. 73 – 74.

8. Марценюк В., Мілян Н. Роль і місце хмарних обчислень серед сучасних інформаційних технологій. *Фундаментальні та прикладні проблеми сучасних технологій: матеріали Міжнародної науково-технічної конференції до 100-річчя з дня заснування НАН України та на вшанування пам'яті Івана Пулюя (100-річчя з дня смерті)*, (Тернопіль, 23–24 травня 2018 року). Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2018. С. 219 – 221. ISBN: 978-966-305-093-5.

9. Марценюк В., Мілян Н. Використання алгоритмів індукції дерева рішень для аналізу великих обсягів даних. *Актуальні задачі сучасних технологій : зб. тез доповідей міжнар. наук.-техн. конф. Молодих учених та студентів*, (Тернопіль, 28–29 листоп. 2018.). М-во освіти і науки України, Терн. націон. техн. ун-т ім. І. Пулюя [та ін]. Тернопіль : ТНТУ, 2018. С. 114 – 116.

10. Мілян Н. Аналіз методів машинного навчання з вчителем. *Збірник тез Міжнародної студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання"*, 26-27 квітня 2018 року. Т. : ТНТУ, 2018. Том 1. С. 51–52.

Праці, які додатково відображають наукові результати дисертації:

11. Спосіб кіберфізичного моделювання сенсорного масиву біопікселів. Патент України на корисну модель №139048 МПК G01N33/483 / Н.В. Мілян, В.П. Марценюк, А.С. Сверстюк, А.М. Клос-Вітковська. Заявка № u201901590. Заявл. 18.02.2019. Опубл. 26.12.2019. Бюл.№ 24.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ..... 18

ВСТУП..... 19

РОЗДІЛ 1. МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ЗАДАЧ МАШИННОГО НАВЧАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ (огляд літератури)..... 27

1.1 Основні області використання і можливості Big Data. Методики аналізу великих даних..... 27

1.2 Технологічні питання задач машинного навчання на основі хмарних обчислень 32

1.2.1 Технології хмарних обчислень. Загальний огляд хмарних платформ 32

1.2.2 Головні складові Google Cloud Platform..... 43

1.2.3 Аналіз сервісу Google App Engine та його аналогів..... 46

1.3 Основні типи невизначеностей у медичних застосуваннях 51

1.4 Алгоритмічні питання задач машинного навчання в умовах невизначеності..... 52

1.4.1 Загальний огляд типових алгоритмів прийняття рішень..... 52

1.4.2 Наукове машинне навчання..... 57

1.4.3 Прийняття рішень в умовах невизначеності на основі мінімаксного підходу 57

1.5 Постановка наукового завдання дисертаційного дослідження..... 64

Висновки до першого розділу..... 65

РОЗДІЛ 2. ТЕХНОЛОГІЧНІ ПИТАННЯ РЕАЛІЗАЦІЇ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ НА ОСНОВІ ХМАРНИХ ОБЧИСЛЕНЬ У ВИПАДКУ СИСТЕМНИХ МЕДИЧНИХ ДОСЛІДЖЕНЬ 67

2.1 Задача індукції дерева рішень на основі хмарних платформ 67

2.1.1 Математичне представлення задачі індукції дерева рішень 67

2.1.2 Прототип реалізації індукції дерева рішень на основі хмарної платформи для системних медичних досліджень 70

2.2 Питання архітектури систем прийняття рішень на основі системних медичних досліджень 74

2.2.1 Питання архітектури систем прийняття рішень: розробка архітектури на основі мікросервісів 74

2.2.2 Концептуальне представлення інформаційної моделі задачі прийняття рішень для невідкладної медичної допомоги при виникненні серцевої недостатності 77

2.2.3 Реалізація рішення задачі невідкладної медичної допомоги на основі архітектури мікросервісів 78

Висновки до другого розділу 84

РОЗДІЛ 3. ЗАДАЧІ КЛАСИЧНОГО МАШИННОГО НАВЧАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ ІНТЕГРОВАНІ З ДАНИМИ ІЗ СИСТЕМ З ВІДКРИТИМ КОДОМ ЕЛЕКТРОННИХ МЕДИЧНИХ ЗАПИСІВ..... 86

3.1 Задача класичного машинного навчання інтегрованого з даними із систем з відкритим кодом електронних медичних записів в умовах визначеності 86

3.1.1 Постановка задачі 86

3.1.2 Математичний опис задачі машинного навчання в системних медичних дослідженнях в умовах визначеності 87

3.1.3 Розробка моделі машинного навчання 89

3.1.4 Задача імпорту даних з систем з відкритим кодом електронних медичних записів 90

3.1.5 Задачі зменшення розмірності та класифікації 94

3.1.6 Результати експериментальних досліджень на основі клінічних даних 96

3.2 Задача класичного машинного навчання в умовах невизначеності 100

3.2.1 Математичний опис задачі машинного навчання в умовах епістемічної та алеаторної невизначеності 100

3.2.2 Зменшення розмірності в умовах невизначеності 102

3.2.3Загальна блок-схема побудови моделі машинного навчання на основі мінімаксного критерію в умовах епістемічної та алеаторної невизначеності 106

3.2.4 Приклад на основі кардіологічних даних 108

Висновки до третього розділу 114

РОЗДІЛ 4. ГЛИБОКЕ МАШИННЕ НАВЧАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ НА ОСНОВІ МІНІМАКСНОГО ПІДХОДУ ТА РОЗРОБКА МОДЕЛІ SCIML 116

4.1 Застосування згорткових мереж для завдань класифікації 116

4.2 Реалізація U-Net моделі з використанням TensorFlow то оцінка результатів з використанням мінімаксного підходу. 121

4.3 Опис біологічних процесів, що лежать в основі кіберфізичних біосенсорних систем 126

4.4 Алгоритм наукового машинного навчання кіберфізичної біосенсорної системи 128

Висновки до четвертого розділу 130

ВИСНОВКИ 132

Список використаних джерел 135

ДОДАТОК А 157

ДОДАТОК Б 160

ДОДАТОК В 175

ДОДАТОК С 182

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

МН – машинне навчання

ЕКГ – електрокардіографія

ЕЕГ – електроенцефалографія

КТ – комп'ютерна томографія

МРТ – магнітно-резонансна томографія

ОВМ – метод опорних векторів (SVM)

МГК – метод головних компонент (PCA)

ГН – глибоке навчання

ОПМ – обробка природної мови (NLP)

ВСТУП

Актуальність теми. Із зростанням обсягу даних відбувається швидкий розвиток галузі машинного навчання (МН), пов'язаний із розробкою теорії та застосуванням в різних галузях. Важливою складовою МН є задачі оптимізації, оскільки суть більшості алгоритмів МН полягає у побудові моделі оптимізації та визначенні параметрів цільової функції на основі наборів навчальних даних. У час появи великих наборів даних Big Data ефективність алгоритмів чисельної оптимізації суттєво впливає на популярність використання моделей МН. Протягом розвитку теорії МН було розроблено ряд важливих методів оптимізації, які покращують продуктивність та ефективність методів.

МН успішно застосовується в медицині для вирішення широкого класу задач. На сьогодні алгоритми МН використовуються не лише з метою аналізу даних об’єктивного обстеження та моніторингу стану пацієнта, а й аналізу результатів обстежень та аналізів з допомогою медичного обладнання, наприклад, ЕКГ, ЕЕГ, КТ, МРТ та інші. При цьому використовуються різноматні моделі, починаючи від регресії, методу опорних векторів (ОВМ), випадкових лісів для навчання з вчителем та методу головних компонент (МГК) для навчання без вчителя [150], [116], [121]. Глибоке навчання (ГН) дозволяє розробити інтелектуальну систему моніторингу стану пацієнта, використовуючи дистанційні датачі, дані з соціальних мереж із підтримкою обробки природної мови (ОПМ). Застосування таких засобів підвищує ефективність роботи лікаря, позбавляючи його від виконання цілого ряду рутинних дій. Однією із важливих задач використання МН в медицині є підготовка коректних медичних даних, що надходять з різних джерел, з метою використання в алгоритмах МН з урахуванням невизначеності.

Невизначеність залишається важливим фактором, що впливає на процес прийняття рішень у всіх галузях. Вона суттєво впливає на точність, чутливість та специфічність застосовуваних алгоритмів. Різні аспекти

застосування МН вимагають врахування невизначеності [28]: прогнозування [140]; прийняття рішень [48]; навчання на основі обмежених, зашумлених та відсутніх даних [30]; моделювання та дослідження складних моделей [146]; стиснення даних [44]; автоматичне наукове моделювання [133] та планування експерименту [188].

Невизначеності виникають у процесі постановки та розв’язування задач МН в основних медичних процесах (діагностика, лікування, профілактика). Моделі, які описують перебіг захворювань та паталогічні процеси, повинні враховувати невизначеності [136]. Тому постає потреба аналізу різного типу задач МН в системних медичних дослідженнях як з вчителем, так і без вчителя, з точки зору впливу невизначеності.

Більшість даних, які використовуються в МН для системних медичних досліджень, надходять з різних сенсорних пристроїв. На основі отриманих спостережень метою є отримати певні висновки або прийняти рішення, передбачити певні показники. Очевидно, що у випадку одних і тих же спостереженнях, повинні бути прийняті однакові рішення або ж передбачені однакові значення. У той же час, можуть використовуватися різні типи вхідних даних в силу різних причин (наприклад, через зменшення розмірів даних). Крім того, для оцінки стану пацієнта можуть використовуватися різні діагностичні пристрої. Такого типу невизначеності називаються алеаторними і вони пов’язані з невизначеностями в даних.

Слід зазначити, що вибір моделі МН також впливає на прийняті рішення та прогнозовані медичні показники. Це пов’язано з поняттями чутливості та специфічності, подібно до того, яке це має місце в медичній диференціальній діагностиці, порівнюючи певні методи системного медичного дослідження. Такий тип невизначеності, який пов'язаний із моделями, відомий як епістемічний. Ці два типи невизначеності (алеаторна та епістемічна) не повинні впливати на прийняті рішення чи прогнозовані значення. Усе це вимагає розробки підходу МН з урахуванням невизначеностей.

У вітчизняній школі кібернетики та інформатики, котра створена В. М. Глушковим, досягнуто значних результатів у напрямку розвитку новітніх інформаційних технологій. У розвиток загальної теорії керування, математичного моделювання та оптимізації, теорії та методів системного аналізу та у розробку відповідних програмних засобів важливий внесок зробили О. Г. Наконечний, Ф. Г. Гаращенко, О. К. Закусило, Б. М. Бублик, Ю. М. Єрмольєв, М. З. Згуровський, Б. Н. Пшеничний та інші. З питань прогнозування процесів та прийняття рішень у складних системах при неточності, неповноті, нечіткості вихідних даних в умовах концептуальної невизначенності одержані фундаметальні та прикладні результати (О. Г. Наконечний, В. М. Кунцевич, Б. М. Бублик, Ю. М. Єрмольєв); методів оцінювання та керування нелінійними системами з розподіленими параметрами (Ю. К. Подлипенко, О. Г. Наконечний); чисельних методів нелінійного аналізу та оптимізації (А. П. Власюк, О. Ю. Грищенко, Б. М. Бублик, І. В. Бейко). У роботах О. С. Коваленка, І. М. Ляшенка, В. П. Марценюка, Ю. Г. Антомонова, М. М. Амосова, М. Б. Славіна, Дж. Снейда, М. Мекі, Н. Бейлі відображено результати застосування теорії автоматичного керування, методів математичного моделювання до медицини та біології. У роботах вчених В. С. Михалевича, І. В. Сергієнка, В. М. Глушкова, О. Г. Наконечного, В. Л. Волковича, В. А. Заславського, І. А. Ушакова, В. В. Акіменка розвинуто підходи до проектування систем МН. Проте навіть зараз є багато невирішених завдань в інформаційних технологіях процесів прийняття клінічних рішень в умовах невизначеності.

Зв’язок роботи з науковими програмами, планами, темами. Дисертаційне дослідження проведено в рамках науково-дослідної роботи Тернопільського національного медичного університету імені І. Я. Горбачевського, зокрема наукової теми «Кібер-фізичне моделювання в дослідженнях медико-біологічних процесів», номер держреєстрації №0119U000509, де здобувачу належить метод розв’язування задачі

класифікації кібер-фізичних систем медико-біологічних процесів на основі алгоритму наукового МН.

Також дисертаційне дослідження виконане в межах програми Європейського Союзу «Еразмус +» для освіти в рамках гранту KA2 (проєкт № 2020-1-PL01-KA203-082197 «Інновації для великих даних у реальному світі»), де здобувачу належить метод розв'язування задач класичного і глибокого МН на основі мінімаксного критерію в умовах епістемічної і алеаторної невизначеностей.

Мета і завдання дослідження. Метою дисертаційного дослідження є розвиток методів класичного, глибокого та наукового МН в умовах невизначеності з використанням мінімаксного підходу, що полягає у розробці та реалізації загальної схеми побудови моделі машинного навчання на основі системних медичних досліджень. Досягнення поставленої мети вимагає виконання таких завдань:

1. Проаналізувати математичне та програмне забезпечення інформаційних технологій задач МН в умовах невизначеності.
2. Запропонувати технологічні рішення до МН на основі системних медичних досліджень на платформах хмарних обчислень.
3. Розробити загальну схему побудови моделі МН на основі системних медичних досліджень.
4. Розв'язати задачі класичного МН в умовах невизначеності інтегровані з даними із систем з відкрити кодом електронних медичних записів.
5. Здійснити постановку задачі глибокого машинного навчання в умовах невизначеності та запропонувати метод її розв'язування на основі мінімаксного підходу.
6. Запропонувати метод розв'язування задачі класифікації кіберфізичих систем медико-біологічних процесів на основі алгоритму наукового МН.

Об'єкт дослідження. Процеси прийняття клінічних рішень в умовах невизначеності.

Предмет дослідження. Інформаційні технології на основі методів класичного і глибокого МН для системних медичних досліджень.

Методи дослідження. Для досягнення поставленої в дисертаційній роботі мети використані наступні методи дослідження: методи системного та порівняльного аналізу для обґрунтування актуальності та постановки наукового завдання дисертаційної роботи; методи алгоритмів прийняття рішень, методи класичного машинного навчання, методи глибокого машинного навчання.

Наукова новизна отриманих результатів.

1. Вперше здійснено постановку задач класичного і глибокого МН в умовах епістемічної і алеаторної невизначеностей, запропоновано метод розв'язування на основі мінімаксного критерію. Це дозволило розробити алгоритм побудови моделі на класах розподілів навчальних та тестових даних на множині ряду методів МН.

2. Вперше розроблено метод зменшення розмірності даних із втратою найменшої кількості інформації на основі аналізу головних складових (МГК) в умовах алеаторної невизначеності. Це дало змогу враховувати лише атрибути з найбільшими значеннями поясненої варіації на множині стратегій передискретизації, що розглядаються.

3. Отримали подальший розвиток задачі наукового МН, запропоновано метод розв'язування задачі класифікації кіберфічних біосенсорних систем. Це дозволило розробити алгоритм якісного аналізу динамічних моделей медико-біологічних процесів, що зводиться до розв'язування задачі МН. На відміну від існуючих методів якісного аналізу динамічних систем на основі теорії стійкості даний підхід дозволяє прогнозувати можливі стани кіберфізичної біосенсорної системи, кількість яких здатна зростати у випадку технології Big Data.

4. Отримали подальший розвиток методи МН на основі хмарних обчислень. Це дало змогу реалізувати прототип програми до розв'язування задач МН на віддалених платформах на основі даних електронних медичних записів з підтримкою технології Big Data для приросту даних.

5. Отримали подальший розвиток методи розробки архітектури інформаційних систем прийняття рішень на основі мікросервісів. Це дозволило одержати концептуальне представлення інформаційної моделі для прийняття рішень у системних медичних дослідженнях на основі архітектури мікросервісів, здатне бути використаним у проблемах профілактики, діагностики та лікування, та впровадженим у шпитальні та лабораторні інформаційні системи на хмарних платформах. Це дає змогу в запропонованому підході підтримувати технологію Big Data.

Практичне значення отриманих результатів. Розглянуто методи класичного МН в умовах невизначеності з використанням мінімаксного підходу, а також глибокого МН в умовах невизначеності з використанням мінімаксного підходу удосконалюють сучасне інформаційне забезпечення для обробки системних медичних досліджень. Реалізовано та представлено у дисертаційному дослідженні: мінімаксний підхід на основі класичного МН для кардіологічних даних, U-Net модель для сегментації судин очного дна засобами МН із застосуванням мінімаксного підходу. Результати роботи впроваджено у навчальний процес Тернопільського національного технічного університету імені Івана Пулюя (акт впровадження від 10.02.2021 р.) та Тернопільського національного медичного університету імені І. Я. Горбачевського (акти впровадження від 4.02.2021 р.).

Особистий внесок здобувача. Усі наукові результати дисертаційної роботи сформульовані та отримані автором самостійно. У наведених працях (Додаток А), опублікованих із співавторами, здобувачеві належать: зокрема у [1] розв'язано задачу класичного МН інтегрованих з даними з систем з відкритим кодом електронних медичних записів, [2] – розв'язано задачу глибокого МН із застосуванням критерію мінімаксу для бінарної крос-

ентропії та часу навчання, [3] – розглянуто сучасний стан та перспективи розвитку мінімаксного підходу в МН, [4] – досліджено алгоритм індукції дерева рішень та можливість його реалізації на хмарних платформах, [5] – проаналізовано можливості хмарних платформ для реалізації алгоритмів дерева рішень, [6] – запропоновано рішення для екстреної медицини на основі архітектури мікросервісів, [7] – здійснено аналіз платформи Google Cloud Platform, [8] – розглянуто роль і місце хмарних обчислень серед сучасних інформаційних технологій, [9] – охарактеризовано алгоритм індукції дерева рішень для аналізу великих обсягів даних, [10] – проаналізовано типові алгоритми машинного навчання з вчителем, [11] – запропоновано метод розв’язування задачі класифікації кіберфізичих систем медико-біологічних процесів.

З наукових робіт, опублікованих у співавторстві, у дисертаційній роботі використані результати особистих досліджень здобувача.

Апробація результатів дисертації. Результати дисертаційної роботи доповідались на конференціях: 2019 Advances in Science and Engineering Technology International Conferences (ASET). Dubai. United Arab Emirates. 2019; Przetwarzanie, transmisja i bezpieczeństwo informacji. Bielsko – Biała. 2018; Przetwarzanie, transmisja i bezpieczeństwo informacji. Bielsko – Biała. 2019; V Міжнародній науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя. Тернопіль. 1 – 2 лютого 2018; Фундаментальні та прикладні проблеми сучасних технологій: матеріали Міжнародної науково-технічної конференції до 100-річчя з дня заснування НАН України та на вшанування пам’яті Івана Пулюя (100-річчя з дня смерті). Тернопіль. 23–24 травня 2018; Конференції молодих учених та студентів. Тернопіль. 28–29 листопада 2018; Міжнародній студентської науково-технічної конференції «Природничі та гуманітарні науки. Актуальні питання». 26-27 квітня 2018.

Публікації. Основні наукові результати дисертації опубліковано в 11 працях, зокрема: дві статті [1], [3] у наукових фахових періодичних виданнях України; одна стаття у закордонному фаховому періодичному виданні [2]; видано один патент на корисну модель [11]; 7 публікацій [4], [5], [6], [7], [8], [9], [10] у матеріалах міжнародних та всеукраїнських наукових, науково-технічних конференцій. З них одна робота входить до міжнародної наукометричної бази Scopus [4], дві роботи [1], [3] входять до міжнародної наукометричної бази Index Copernicus.

Структура та обсяг дисертації. Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 195 найменувань та додатків. Повний обсяг дисертації складає 184 сторінки, основний зміст викладено на 116 сторінках, де наведено 25 рисунків та 8 таблиць.

РОЗДІЛ 1. МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ЗАДАЧ МАШИННОГО НАВЧАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ (огляд літератури)

У розділі здійснено огляд математичного та програмного забезпечення інформаційних технологій задач МН в умовах невизначеності. Описано поняття Big Data та методи аналізу Big Data. Розглянуто технологічні питання задач МН на основі хмарних обчислень, а також алгоритмічні питання задач МН в умовах невизначеності. Проведено огляд результатів у галузі МН на основі мінімаксного підходу. Описано важливість оптимізації, ефективність та дієвість алгоритмів чисельної оптимізації. На основі проведеного аналізу визначено наукове завдання, яке полягає у розробці програмного та математичного забезпечення інформаційних технологій, пов’язаних із прийняттям рішень в системних медичних дослідженнях.

Основні результати цього розділу опубліковано в роботах [8], [10], [12], [15].

1.1 Основні області використання і можливості Big Data. Методики аналізу великих даних

Разом зі стрімким накопиченням інформації швидкими темпами розвиваються і технології аналізу даних. Наприклад, якщо раніше була можливість лише сегментувати клієнтів на групи зі схожими особливостями, то тепер можливо будувати моделі для кожного клієнта в режимі реального часу, аналізуючи, наприклад, його переміщення по мережі Інтернет для пошуку конкретного товару. Інтереси споживача можуть бути проаналізовані і, відповідно до побудованої моделі, виведена відповідна реклама або конкретні пропозиції. Модель також може налаштовуватися і перебудовуватися в режимі реального часу, що було неможливо відносно недавно.

Big Data – популярний нині термін, що фігурує майже на всіх професійних конференціях, присвячених аналізу даних, прогностичній аналітиці, інтелектуальному аналізу даних (Data Mining). Термін використовується у сферах, де актуальна робота з якісно великими обсягами даних, де постійно відбувається збільшення швидкості потоку даних у процесах різних галузей: економіці, медицині, банківській діяльності, виробництві, маркетингу, телекомунікаціях, вебаналітиці та інших галузях.

Особливої уваги заслуговує аналіз даних у медицині, так як це одна із найбільш важливих галузей для людства і, відповідно, ефективне і правильне опрацювання даних у цій сфері є одним із найпріоритетніших завдань.

Варто відзначити, що у зв’язку із зростанням обсягу даних, паралельно активно розвиваються технології опрацювання великих даних. МН дало можливість здійснити багато теоретичних проривів і застосовується зараз фактично у всіх сферах діяльності людини, де відбувається опрацювання різного роду інформації. Більшість експертів погоджуються, що прискорення зростання обсягу даних є об'єктивною реальністю. Соціальні мережі, мобільні пристрої, дані з вимірювальних пристроїв, бізнес-інформація – ось лише кілька видів джерел, здатних генерувати гігантські обсяги інформації.

Значну частину інформації створюють не люди, а роботи, які взаємодіють як один з одним, так і з іншими мережами даних – такі, як, наприклад, сенсори й інтелектуальні пристрої. При таких темпах зростання кількість даних у світі, за прогнозами дослідників, щорічно подвоюватиметься. Кількість віртуальних і фізичних серверів у світі виросте в десятки разів за рахунок розширення і створення нових data-центрів. У зв'язку з цим зростає потреба в ефективному використанні і монетизації цих даних. Оскільки використання Big Data в бізнесі вимагає чималих інвестицій, то треба ясно розуміти ситуацію. А вона, по суті, проста: підвищити ефективність бізнесу можна скорочуючи витрати або/і збільшуючи обсяг продажів [21].

Парадигма Big Data визначає три основні типи завдань:

– Зберігання та управління обсягом даних у сотні терабайт або петабайт, які звичайні реляційні бази даних не дозволяють ефективно використовувати.

– Організація неструктурованої інформації, що складається з текстів, зображень, відео та інших файлів.

– Аналіз Big Data, який ставить питання про способи роботи з неструктурованою інформацією, генерацію аналітичних звітів, а також впровадження прогностичних моделей.

Ринок проєктів Big Data перетинається з ринком бізнес-аналітики (BA), обсяг якого у світі, за оцінками експертів, у 2012 році склав близько 100 млрд. доларів. Він включає в себе компоненти мережевих технологій, серверів, програмного забезпечення і технічних послуг.

Використання технологій Big Data також актуально для вирішення задач гарантування доходів (RA), призначених для автоматизації діяльності компаній. Сучасні системи гарантування доходів включають у себе інструменти виявлення невідповідностей і поглибленого аналізу даних, що дозволяють своєчасно виявити можливі втрати, або спотворення інформації, здатні привести до зниження фінансових результатів [21].

Існує багато [1] різноманітних методик аналізу масивів даних, в основі яких лежить інструментарій, запозичений із статистики та інформатики (наприклад, МН).

– *A/B-тестування*. Методика, у якій контрольна вибірка по черзі порівнюється з іншими. У такий спосіб вдається виявити оптимальну комбінацію показників для досягнення, наприклад, найкращої відповідної реакції споживачів на маркетингову пропозицію. Великі дані дозволяють провести величезну кількість ітерацій і таким чином отримати статистично достовірний результат.

– *Аналіз тональності тексту*. Набір методів контент-аналізу комп’ютерної лінгвістики, призначених для автоматизованого виявлення емоційно забарвленої лексики в тексті.

– *Аналіз часових рядів*. Набір, запозичених з статистики та цифрової обробки сигналів, методів аналізу повторюваних з часом послідовностей даних, наприклад, відстеження захворюваності пацієнтів.

– *Ансамблеве навчання*. Набір методик, що поєднують кілька моделей МН в одну прогностичну модель з метою зменшення дисперсії, підсилювання або покращення прогнозів.

– *Візуалізація*. Методи графічного представлення результатів аналізу великих даних у вигляді діаграм або анімації для спрощення інтерпретації полегшення розуміння отриманих результатів.

– *Генетичний алгоритм*. Алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації параметрів з використанням механізмів, що нагадують біологічну еволюцію.

– *Глибинний аналіз даних*. Набір методик для виявлення у великих масивах даних, що накопичуються, раніше невідомих, нетривіальних, практично корисних і доступних для інтерпретації знань, необхідних для прийняття рішень у різних сферах людської діяльності.

– *Злиття даних та інтеграція даних*. Набір методик, який дозволяє інтегрувати дані з різних джерел з метою проведення глибинного аналізу (Data Mining).

– *Класифікація*. Набір методик, який дозволяє передбачити поведінку споживачів у певному сегменті ринку (прийняття рішень про покупку, відтік, обсяг споживання та інше). Використовується в Data Mining.

– *Кластерний аналіз*. Статистичний метод класифікації об'єктів по групах за рахунок виявлення заздалегідь невідомих загальних ознак. Використовується в глибинному аналізі даних Data Mining.

– *Моделювання*. Моделювання поведінки складних систем, часто використовується для прогнозування, передбачення і опрацювання різних сценаріїв при плануванні.

– *Навчання асоціативних правил*. Набір методик МН для виявлення взаємозв'язків, тобто асоціативних правил, між змінними величинами в великих масивах даних. Використовується в Data Mining.

– *Навчання без вчителя*. Набір, заснованих на технологіях МН, методик, які дозволяють виявити приховані функціональні взаємозв'язки в аналізованих масивах даних. Має спільні риси з кластерним аналізом.

– *Навчання з учителем*. Набір, заснованих на технологіях МН, методик, які дозволяють виявити функціональні взаємозв'язки в аналізованих масивах даних.

– *Обробка природної мови (ОПМ (NLP)).* Набір запозичених з інформатики та лінгвістики методик розпізнавання природної мови людини.

– *Оптимізація*. Набір чисельних методів для редизайну складних систем і процесів для поліпшення одного або декількох показників. Допомагає в прийнятті стратегічних рішень, наприклад, проведенні інвестиційного аналізу тощо.

– *Передбачуване моделювання*. Набір методик, які дозволяють створити математичну модель, наперед заданого ймовірного сценарію розвитку подій.

– *Просторовий аналіз*. Набір частково запозичених з статистики методик аналізу просторових даних – топології місцевості, географічних координат, геометрії об'єктів. Джерелом великих даних у цьому випадку часто виступають геоінформаційні системи (ГІС).

– *Регресія*. Набір статистичних методів для виявлення закономірностей між зміною залежної змінної і однією або декількома незалежними. Часто застосовується для прогнозування. Використовується в Data Mining.

– *Статистика*. Наука про збір, організацію та інтерпретацію даних, включаючи розробку опитувальників і проведення експериментів. Статистичні методи часто застосовуються для оціночних суджень про взаємозв’язки між тими чи іншими подіями.

– *Теорія розпізнавання образів*. Набір методик для автоматичного розпізнавання зразків та закономірностей у даних, має застосування в статистичному аналізі даних, обробці сигналів, аналізі зображень, стисненні даних тощо.

Список не претендує на повноту, проте в ньому відображені найбільш затребувані в різних галузях підходи. При цьому слід розуміти, що дослідники продовжують працювати над створенням нових методик і вдосконаленням існуючих. Крім того, деякі з перерахованих методик зовсім не обов'язково можуть бути застосовані виключно до великих даних і можуть з успіхом використовуватися для менших за обсягом масивів (наприклад, A/B-тестування, регресійний аналіз тощо).

1.2 Технологічні питання задач машинного навчання на основі хмарних обчислень

1.2.1 Технології хмарних обчислень. Загальний огляд хмарних платформ

Впродовж останніх років виникли і динамічно розвиваються нові інформаційні технології, що підтримують, між іншим, наукові дослідження і виконання масштабних обчислень у сфері математичного моделювання і комп'ютерних симуляцій. Такі технології як: віртуалізація (англ. virtualization), хмарні обчислення (англ. cloud computing), розподілені і паралельні обчислення (англ. distribution and parallel computing), кластерні обчислення (англ. cluster computing), грід-обчислення (англ. grid computing) і галузеві гріди (англ. domain grids) дозволяють виконувати обчислювальні завдання, які раніше були неможливими для реалізації. Наведені вище методи розвинулися на основі Інтернет і пов'язаних з ним технологій, таких як WWW, а особливо Web 2.0.

Хмарні обчислення – це особливий спосіб надання обчислювальних ресурсів, а не нова технологія, вони викликали революцію в методах надання

інформації та послуг. Спочатку в інформаційних технологіях (IT) домінували мейнфрейми, великі універсальні високопродуктивні відмовостійкі сервери зі значними ресурсами введення-виведення, великим об'ємом оперативної і зовнішньої пам'яті, призначеній для використання в критично важливих системах (англ. Mission-critical) з інтенсивною пакетною і оперативною транзакціонною обробкою. Згодом ця конфігурація поступилася дорогій клієнт-серверній моделі. Сучасна IT-галузь стає все більш мобільною, хмарною. Щоб правильно розуміти хмарні обчислення, потрібно враховувати, що вони успадковують попередні системи. У сучасному світі хмарних обчислень є місце для інноваційного кооперування хмарної технології та перевіреної ефективності попередніх систем, таких як потужні мейнфрейми [27].

Хмарні обчислення – модель забезпечення доступу через мережу до спільних обчислювальних ресурсів, що підлягають налаштуванню (наприклад, до комунікаційних мереж, серверів, засобів збереження даних, прикладних програм та сервісів), які можуть бути оперативно надані та звільнені з мінімальними управлінськими затратами та зверненнями до провайдера [145].

Хмарні обчислення – це комплексне рішення, яке надає IT-ресурси у вигляді сервісу, засноване на інтернет-технологічних рішеннях, у яких ресурси загального користування надаються аналогічно розподілу електроенергії по електричних проводах. Комп'ютери в хмарі налаштовані на спільну роботу, а різні додатки використовують сукупну обчислювальну потужність так, ніби виконуються на одній системі. Гнучкість хмарних обчислень залежить від можливості розподілу ресурсів на вимогу. Такий розподіл дозволяє використовувати сукупні ресурси системи без виділення конкретних апаратних ресурсів певної задачі. До хмарних обчислень Web-сайти і серверні додатки працювали на окремо взятих системах. З приходом хмарних обчислень ресурси використовуються як об'єднаний віртуальний

комп'ютер. Така об'єднана конфігурація надає середовище, у якому програми виконуються незалежно без прив'язки до будь-якої конкретної конфігурації.

Для переходу до парадигми хмарних обчислень є вагомі причини – як з точки зору бізнесу, так і з точки зору IT. Тут діють такі основні аргументи як: зменшення витрат, оптимальне використання персоналу, надійна масштабованість.

Модель хмарних обчислень складається з зовнішньої (front end) і внутрішньої (back end) частин. Ці два елементи з'єднані мережею, в більшості випадків через Інтернет. За допомогою зовнішньої частини користувач взаємодіє з системою; внутрішня частина – це власне саме хмара. Зовнішня частина складається з клієнтського комп'ютера або мережі комп'ютерів підприємства і додатків, що використовуються для доступу до хмари. Внутрішня – надає додатки, комп'ютери, сервери і сховища даних, що створює хмару сервісів [27].

Завдяки використанню систем хмарних обчислень розробники та співробітники IT-відділів можуть концентруватися на виконанні найважливіших завдань і не витрачати зусилля на роботу з матеріально-технічного забезпечення, обслуговування та планування потужності обчислювальних ресурсів. За мірою зростання популярності даних систем виникло кілька різних моделей та стратегій розгортання, що дозволяють задовольнити потреби різних категорій користувачів. Кожен тип сервісів і кожного способу розгортання забезпечує свій рівень контролю, гнучкості та керованості. Розуміння різниці між моделями Iaas, Paas і Saas та особливостями стратегій розгортання допомагає прийняти рішення про те, який набір послуг найбільш повно задовольнить певні потреби.

Модель «інфраструктура як послуга», скорочено IaaS, включає в себе базові елементи для побудови обладнаної IT-системи. У рамках цієї моделі користувачі отримують доступ до мережевих ресурсів, до віртуальних комп'ютерів або виділеного апаратного забезпечення, а також до зберігання даних. Модель «інфраструктура як служба» забезпечує найвищий рівень

гнучкості експлуатації та керування IT-ресурсами. Вона практично аналогічна сучасній моделі IT-ресурсів.

Модель «платформа як послуга» (PaaS) не вимагає організації управління базовою інфраструктурою (зазвичай включає в себе обладнання та операційні системи) і дозволяє докладати всі зусилля до розробки та керування додатками. Це підвищує продуктивність роботи, оскільки придбання матеріально-технічних ресурсів, планування потужності, обслуговування ПО, встановленням оновлень безпеки та виконанням інших трудоємких завдань, необхідних для роботи додатків не є актуальним.

У рамках моделі SaaS «програмне забезпечення як послуга» користувач отримує завершений продукт, який працює під управлінням постачальника послуги. Зазвичай у цьому випадку мова йде про додатки для кінцевих користувачів. Робота з моделлю SaaS дозволяє повністю сконцентруватися на використанні певного програмного забезпечення, оскільки підтримка сервісу чи управління базовою інфраструктурою не є актуальною. Приклад програми SaaS – вебсервіс електронної пошти, що дозволяє відправляти і отримувати електронні листи без необхідності керувати доповненнями до програмного продукту або обслуговувати сервери і операційні системи, на яких працює сервіс [177].

Найбільшими постачальниками хмарних [80] обчислень є: Amazon Web Services (AWS), Google Cloud Platform (GCP) і Microsoft Azure.

У січні 2017 року RightScale проведено шосту щорічну оцінку «State of the Cloud Survey», в якій взяли участь більше 1000 спеціалістів інформаційних технологій для аналізу поточних тенденцій хмарних обчислень [1]. Було отримано такі результати:

– У 2016 році 32% учасників відповіли, що проблема, пов'язана з областю хмарних обчислень, була пов'язана з браком ресурсів або досвіду. У 2017 році ця цифра знизилася до 25%.

– У 2016 році 29% учасників відповіли, що вони стурбовані безпекою хмарних обчислень. У 2017 році ця цифра знизилася до 25%.

– У 2016 році 15% учасників відповіли, що продуктивність була складною задачею у сфері хмарних обчислень. У 2017 році ця цифра знизилася до 11%.

Ці дані демонструють тенденцію, що бар’єр входу в галузь хмарних обчислень швидко зменшується. Продуктивність і безпека також зробили стрибки і обмеження протягом останніх кількох років тому, що постачальникам хмарних обчислень довелося винайти нові способи більш безпечного розміщення даних, одночасно забезпечуючи їх швидкість.

Ще один цікавий аспект опитування полягав у перегляді загальноприйнятих статистичних даних щодо прийняття хмарності від 2017 року та 2016 року. Дані AWS залишилися приблизно однаковими, у той час як у Azure та Google Cloud відбулося значне зростання. AWS залишається лідером, це пов'язано з тим, що вони першими інвестують та формують галузь промисловості хмарних обчислень. Ось деякі додаткові статистичні дані та прогнози щодо галузі хмарних обчислень:

При розгляді Microsoft Azure варто звернути особливу увагу на Microsoft Azure Machine Learning [159] – це сукупність послуг та інструментів, призначених для розробки та розгортання навчальних моделей МН. Корпорація Майкрософт забезпечує ці інструменти та сервіси через загальну хмару Azure.

Вчені та розробники даних можуть використовувати інструменти Microsoft Azure Machine Learning для створення та розгортання моделей локально, у хмарі Azure і зовні, використовуючи обчислювальні технології Azure IoT. Проте Azure також пропонує кілька високопродуктивних варіантів розгортання, в тому числі:

– VMs із графічними процесорами (GPU): візуалізатори Azure, розроблені для керування проектами МН, все частіше використовують графічні процесори, а не більш традиційні центральні процесори (CPU), оскільки вони можуть обробляти складну математичну та паралельну

обробку, необхідну для ефективного відтворення зображень, яка ідеально підходить для штучного інтелекту та обчислень у МН.

– Microsoft Machine Learning Server: ця опція розгортання надає сервер корпоративного класу, призначений спеціально для розподілених, дуже паралельних робочих навантажень, розроблених мовами R або Python. Він призначений для виконання завдань, таких як високопродуктивна аналітика, МН та аналіз даних, і працює на Linux, Windows, Hadoop і Apache Spark.

– Віртуальна машина Azure Data Science: це віртуальна машина для Azure, призначена для наукових проектів під управлінням Windows Server, Ubuntu Linux та OpenLogic CentOS. Вона включає в себе засоби інформатики та інструменти розробки, підприємства можуть використовувати його для створення аналітики даних та додатків для МН. Розробники можуть викликати візуалізацію Azure Data Science з Azure Experimentation або Model tools.

– Microsoft Azure Machine Learning інтегрується з безліччю платформ і механізмів МН, багато з яких є відкритим вихідним кодом. Окрім Microsoft Cognitive Toolkit, основні засоби підтримки включають Spark ML, TensorFlow та scikit-learn framework [159].

Компанія Amazon об'єднує свою інфраструктуру як сервісну пропозицію у чотири категорії: обчислення, зберігання та доставка контенту, бази даних та мереж. Всі ці ресурси використовуються за умови забезпечення служб безпеки та ідентифікації Amazon, що включають в себе Amazon Active Directory, AWS Identity Management, менеджер сертифікатів AWS для керування сертифікатами SSL/TLS і апаратну кеш-пам'ять та управління через AWS CloudHSM. Можна відстежувати використання ресурсу інфраструктури за допомогою таких інструментів управління, як Amazon CloudWatch, AWS CloudTrail для відстеження активності користувачів та використання API, а також AWS Config для відстеження інвентаризації ресурсів та змін [129].

Далі є необхідність розглянути можливості МН в Amazon ML, а саме ключові поняття:

– Джерела даних містять метадані, пов'язані з введенням даних до Amazon ML.

– Моделі МН створюють прогнози за допомогою шаблонів, витягнутих із вхідних даних.

– Оцінки вимірюють якість моделей МН.

– Пакетні прогнози асинхронно створюють прогнози для кількох спостережень вхідних даних.

– Прогнози в режимі реального часу синхронно створюють прогнози для окремих спостережень даних.

Джерело даних – це об'єкт, який містить метадані про вхідні дані. Amazon ML зчитує вхідні дані, обчислює описову статистику своїх атрибутів і зберігає статистику разом із схемою та іншою інформацією як частина об'єкта джерела даних. Далі Amazon ML використовує джерело даних для навчання та оцінки моделі ML і генерації пакетних прогнозів.

Модель МН – це математична модель, яка генерує прогноз, знаходить шаблони у даних. Amazon ML підтримує три типи моделей МН: бінарну класифікацію, багатокласову класифікацію та регресію.

Якщо ж порівняти GCP та AWS, то необхідно звернути увагу, що служби у сервісах хмарних обчислень в основному представляють одну і ту ж функціональність та мають певні свої особливості. Наприклад, в AWS центральною обчислювальною службою є сервіс Elastic Compute Cloud (EC2), а у GCP – Compute Engine.

GCP складається з безлічі різних послуг та рішень, які дозволяють використовувати ту ж програмну та апаратну інфраструктуру, яку Google використовує для власних продуктів, таких як YouTube та Gmail. Вони запустили свою першу службу Google App Engine у 2008 році. Кількість їхніх продуктів тепер складає більше 50, до основних належать: Google Compute Engine, Google App Engine, Google Container Engine, Google Cloud Bigtable,

Google BigQuery, Google Cloud функції, Google Cloud Datastore, Google Storage, CDN Google Cloud, Google Cloud DNS.

AWS є дочірньою компанією Amazon.com, яка розпочала свою діяльність у 2006 році для надання послуг хмарних обчислень для бізнесу та приватних осіб. Як і у GCP, вони мають безліч різних послуг та рішень, що складає більше 200. До основних належать: Amazon Elastic Compute Cloud (Amazon EC2), AWS Elastic Beanstalk, Amazon EC2 Container Service, Amazon DynamoDB, Amazon Redshift, Amazon Lambda, Amazon S3, Amazon CloudFront, Amazon Route 53.

Якщо ж проводити порівняння, то Google Cloud і AWS дуже схожі, тому простіше звести порівняння до одних категорій. Таким чином, порівняння може бути здійсненим по таких категоріях, як обчислення (англ. compute), зберігання/диск (англ. storage/disk), мережа (англ. network), оплата та ціноутворення (англ. billing and pricing), підтримка та відмовостійкість (англ. support and uptime), безпека (англ. Security).

Також існують менші постачальники хмарних послуг, наприклад IBM SmartCloud, Rackspace Cloud, Oracle Exalogic Elastic Cloud, Parallels.

Хмарне рішення, пропоноване компанією IBM, а саме IBM SmartCloud, реалізує всі три моделі (IaaS, SaaS, PaaS) в рамках не тільки публічної, але і приватної гібридної хмари. У його склад входить хмарний сервіс IBM Lotus Live, який надає бізнес-додатки за моделлю SaaS. Містить повний набір інтерактивних сервісів, які надають масштабовані для організації захищені системи електронної пошти, проведення вебконференцій та колективної роботи. Елементи управління системою захисту, розгорнуті для Lotus Live, забезпечують приватність і керований доступ до важливої інформації при виконанні бізнес-операцій. Всі клієнтські взаємодії кодуються стійкими алгоритмами шифрування і здійснюються по протоколу SSL для HTTP і через RC2 в протоколі системи миттєвого обміну повідомленнями Lotus Sametime. Резервні копії системи шифруються [154].

Платформа Rackspace Cloud пропонує набір продуктів для автоматизації хостингу і хмарних обчислень, реалізується модель PaaS. Об'єднує в собі Cloud Files, Cloud Servers, Cloud Sites. Завдяки серверній віртуалізації користувачі отримують можливість розгортати сотні хмарних серверів одночасно і створювати архітектуру, що забезпечує високу доступність. Є конкурентом Amazon Web Services .

Розглянемо детальніше сервіси хмарних платформ та необхідні дії для розгортання додатків. Необхідно звернути увагу, що служби у сервісах хмарних обчислень в основному представляють одну і ту ж функціональність, лише назви відрізняються і мають звичайно певні свої особливості. Наприклад, в AWS центральною обчислювальною службою є сервіс Elastic Compute Cloud (EC2), а у GCP – Compute Engine.

Основними службами у GCP є: Compute Engine, App Engine Standard Environment, App Engine Flexible Environment, App Engine, Cloud SQL, Cloud Spanner, Cloud Datastore, Cloud Bigtable та інші. Якщо мова йде про створення та розгортання вебдодатків то особливу увагу слід звернути на Google App Engine – це гнучка платформа як служба (PaaS), сервіс хостингу сайтів і вебаплікацій на серверах Google з безкоштовним ім'ям в домені appspot.com або з власним ім'ям.

Типовий спосіб розгортання додатку на сервісі Google App Engine виглядає таким чином: в першу чергу у інтегрованому середовищі розробки (ÍDE), наприклад Eclipse, потрібно встановити плагін під назвою «Google Cloud Platform». З допомогою встановленого плагіна можна створити проєкти типу «Google App Engine Standart Java Project», структура створеного проєкту містить необхідні файли для розгортання на хмарній платформі. Наступним кроком потрібно зареєструватися у сервісі Google App Engine за посиланням appengine.google.com, після реєстрації потрібно створити новий проєкт, важливо розуміти що ім’я проєкту буде частиною домену сайту та вказати регіон розташування сервера Google (наприклад us-central). Для розгортання проєкту потрібно залогінитися в плагін на IDE, вказати доступи

до служби Google App Engine і у випадку Eclipse скористатися функцією «Deploy to App Engine Standart». Після завантаження проєкту, результат його роботи буде вже безпосередньо на домені appspot.com, тобто на платформі Google Cloud Platform. Крім того, у адміністративній панелі App Engine можна переглядати різноманітні статистики сайту, які надає Google.

Основними службами у Microsoft Azure є: Virtual Machines, Virtual Machine Scale Sets, Cloud Services, Service Fabric, Storage (Block Blob), SQL Database, Database for MySQL, Database for PostgreSQL, Cosmos DB, Table Storage та інші. Служби Azure дають можливість вирішення багатьох завдань – від звичайних (наприклад, створення віртуальної машини (VM) і завантаження програмного забезпечення або розгортання нової бази даних SQL) до складних (наприклад, впровадження робочих процесів безперервної інтеграції (CI) і безперервного постачання (CD) або роботи з контейнерами). За допомогою Azure є можливість автоматичного налаштовування бази даних та швидкої діагностики проблем з веб-додатками. Крім того, для використання служб необхідні мінімальні зусилля. Якщо мова йде про створення та розгортання вебдодатків, то особливу увагу слід звернути на Azure App Service – це набір служб для хостингу і оркестрації.

Типовий спосіб розгортання додатків на платформі Azure виглядає таким чином: спочатку необхідно зареєструватися в Azure, після цього на порталі Azure потрібно створити вебдодаток, при створенні якого заповнити необхідні поля. Щоб розмістити базу даних, потрібно створити базу даних SQL Azure, яка працює так само, як локальна база даних SQL Server, під час створення необхідно заповнити необхідні поля і вкінці скопіювати стрічку підключення до бази даних. Наступним кроком є запуск проєкту локально: для цього потрібно виконати git команди для завантаження і виконання проєкту локально, тут же локально є можливість створення нових проєктів Azure. Відповідно, на цьому етапі є додаток з базою даних, який працює локально. Для подальшого розгортання на платформі Azure необхідно в базі даних змінити конфігурацію локально, а також на порталі Azure. Щоб

продовжити розгортання, потрібно підключити локальний репозиторій git до Azure, пройти перевірку ідентичності. Користувач розгортання визначається на рівні облікового запису і відрізняється від облікового запису підписки на Azure. Користувач розгортання налаштовується один раз. Для кінцевого розгортання на порталі Azure потрібно скористатися Azure Cloud Shell та командами git на локальному репозиторії для завантаження додатку.

Основними службами у AWS є: Elastic Compute Cloud (EC2), Elastic Beanstalk, Simple Storage Services (S3), RDS, DynamoDB, SimpleDB та інші. Якщо мова йде про створення та розгортання вебдодатків, то особливу увагу слід звернути на Elastic Beanstalk – це найшвидший і простий спосіб розгортання додатків в AWS. Щоб завантажити додаток, слід використовувати консоль управління AWS, репозиторій Git або інтегроване середовище розробки (IDE), наприклад Eclipse або Visual Studio. Elastic Beanstalk виконає розгортання, виділить ресурси, забезпечить балансування навантаження, автоматичне масштабування і моніторинг працездатності програми.

Типовий спосіб розгортання, наприклад додатку node.js, на платформі Azure виглядає таким чином: спочатку потрібно створити акаунт у Elastic Beanstalk, з акаунта створити новий додаток, вказати назву, опис, створити новий веб-сервер при створені якого вказати тип додатку, який буде розгорнутий (node.js, java, php та інші), на наступному кроці потрібно завантажити безпосередньо сам додаток, наприклад архівом .zip, вказати та перевірити остаточне ім’я сайту в мережі з доменом .elasticbeanstalk.com. Далі потрібно вказати деякі конфігураційні дані потужність сервера та деякі додаткові опції, після цього здійснюється процес створення інстансу та розгортання додатку. Після проведених дій з’являється можливість переглянути за посиланням додаток, розгорнутий на платформі AWS

З огляду на вище викладене, обрано для подальшої роботи Google Cloud Platform, оскільки дана платформа хмарних обчислень володіє усім необхідним для реалізації алгоритму індукції дерева рішень C5.0, що будує

дерево рішень, яке може передбачити клас для нових пацієнтів на основі їхніх атрибутів. Таким чином, у кожному пункті блок-схеми задається питання про значущість певного атрибута, і, залежно від цих атрибутів, пацієнт потрапляє до певного класу.

1.2.2 Головні складові Google Cloud Platform

Google Cloud Platform (GCP) – це набір загальнодоступних послуг хмарних обчислень, пропонованих Google. Платформа включає в себе ряд сервісних служб для обчислення, зберігання та розробки додатків, що працюють на апаратній системі Google. До служб GCP можуть звертатися розробники програмного забезпечення, адміністратори хмарних мереж та інші корпоративні IT-фахівці через загальнодоступний Інтернет або через спеціальне мережеве з’єднання.

До складу GCP входять наступні продукти та служби [43]: обчислення (compute), зберігання (storage), мережа (networking), великі дані (Big data), МН (machine learning), інтернет речей (IoT), інструменти керування хмарою (management tools), безпеки (security), інструменти розробника (developer tools), передача даних (data transfer), API платформа та екосистеми (API Platform and Ecosystems) .

Основними продуктами (Таблиця 1.1) хмарних обчислень в GCP є [158]:

– Google Compute Engine, що пропонує інфраструктуру як службу (IaaS), яка надає користувачам віртуальні примірники для робочого хостингу.

– Google App Engine, що пропонує платформу як послугу (PaaS), дає розробникам програмного забезпечення доступ до масштабованого хостингу Google. Розробники також можуть використовувати комплект розробника програмного забезпечення (SDK) для розробки програмних продуктів, що працюють в App Engine.

– Google Cloud Storage, яка є платформою хмарного зберігання, призначена для зберігання великих, неструктурованих наборів даних. Google

також пропонує варіанти зберігання бази даних, включаючи Cloud Datastore для нереляційного зберігання NoSQL, Cloud SQL для повного реляційного зберігання MySQL і базу даних Cloud Bigtable Google.

– Google Container Engine, яка є системою керування для контейнерів Docker, що працюють у загальнодоступній хмарі Google.

Таблиця 1.1 – Перелік продуктів GCP

Compute	**Storage and databases**	**Networking**	**Big data and IoT**	**Machine learning**
▪ Compute Engine ▪ App Engine ▪ Container Engine ▪ Cloud Functions	▪ Cloud Storage ▪ Cloud SQL ▪ Cloud Bigtable ▪ Cloud Spanner ▪ Cloud Datastore ▪ Persistent Disk ▪ Data Transfer	▪ Virtual Private Cloud (VPC) ▪ Cloud Load Balancing ▪ Cloud CDN ▪ Cloud Interconnect ▪ Cloud DNS	▪ BigQuery ▪ Cloud Dataflow ▪ Cloud Dataproc ▪ Cloud Datalab ▪ Cloud Dataprep ▪ Cloud Pub/Sub ▪ Genomics ▪ Google Data Studio ▪ Cloud IoT	▪ Cloud Machine Learning Engine ▪ Cloud Jobs API ▪ Cloud Speech API ▪ Cloud Translation API ▪ Cloud Vision API ▪ Cloud Video Intelligence

Більшість сервісів GCP користуються моделлю pay-as-go, в якій користувачі платять лише за витрати хмарних ресурсів, які вони споживають.

GCP пропонує цілий ряд обчислювальних сервісів, які надають користувачам повний контроль (наприклад, Compute Engine) для високо абстрактних функцій (наприклад, Firebase і Cloud функції) (Рис. 1.1), що дозволяє Google стежити все більше і більше за управлінням та операціями на цьому шляху.

Рисунок 1.1 – Служби обчислень GCP

Google Cloud Platform пропонує послуги з розробки та інтеграції додатків. Наприклад, Google Cloud Pub / Sub є службою керування повідомленнями в реальному часі, яка дозволяє обмінюватися повідомленнями між додатками. Крім того, кінцеві точки Google Cloud дозволяють розробникам створювати служби на основі API RESTful, а потім робити ці служби доступними для клієнтів Apple iOS, Android та JavaScript. Інші пропозиції включають в себе Anycast DNS-сервери, прямі мережеві з'єднання, балансування навантаження, послуги з контролю та реєстрації.

Google продовжує додавати свої послуги на більш високому рівні, наприклад, ті, що стосуються великих даних і МН, до своєї хмарної платформи. Сервіси Google для великих даних включають інструменти для обробки та аналізу даних, такі як запити на Google BigQuery для SQL. Крім того, Google Cloud Flow – це служба обробки даних, призначена для аналітики; витягнення, перетворення та завантаження (ETL); і обчислювальні проекти в режимі реального часу. Платформа також включає Google Cloud Dataproc, який пропонує послуги Apache Spark та Hadoop для великої обробки даних.

Для штучного інтелекту (Google Analytics) Google пропонує Google Cloud Machine Learning Engine, керовану службу, яка дозволяє користувачам створювати та тренувати моделі машинного навчання. Різні API також доступні для перекладу та аналізу мови, тексту, зображень та відео.

Google також надає послуги для IoT, такими як Google Cloud IoT Core, яка являє собою серію керованих служб, що дозволяє користувачам використовувати та керувати даними з пристроїв IoT.

Набір послуг Google Cloud Platform завжди розвивається, і компанія Google періодично вводить, змінює або припиняє надання послуг на основі попиту або конкурентного тиску користувачів.

1.2.3 Аналіз сервісу Google App Engine та його аналогів

Google App Engine (GAE) - сервіс хостингу сайтів і вебдодатків на серверах Google з безкоштовним ім'ям <ім'я_сайт> .appspot.com, або з власним іменем, задіяним за допомогою Google служб.

Google App Engine – система розробки хмарних додатків від Google, створена на основі інструментів, які використовуються Google в інфраструктурі своїх прикладних програм. App Engine дозволяє розробникам створювати в Python і Java масштабовані додатки, не задумуючись про низькорівневу архітектуру. У той же час App Engine має важливі архітектурні обмеження: створювати додатки можна тільки на мовах Python і Java (також доступні для реалізації декількох інших мов, які виконуються на поверхні Java), замість традиційного СУБД з SQL-синтезатором надано СУБД з синтезом GQL, де не підтримується оператор JOIN, що дозволяє об’єднувати таблиці в базі даних. Доступ до файлової системи можливий в режимі «тільки для читання» [69].

Схема інфраструктури Google App Engine зображена на рис. 1.2.

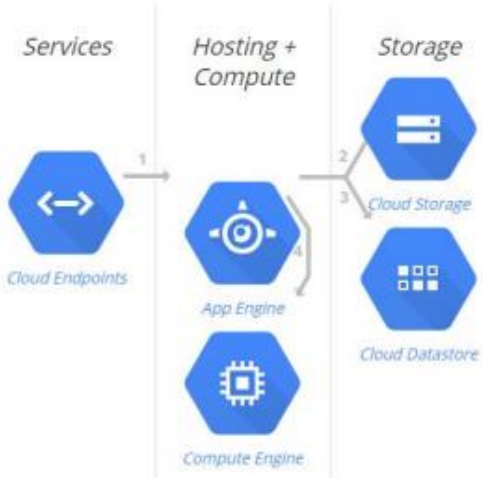


Рисунок 1.2 – Схема інфраструктури Google App Engine

Додатки, що розгортаються на базі App Engine, повинні бути написані на Python, Java, Go або PHP. Середовище виконання Python включає в себе повну реалізацію можливостей самого Python, більшість функцій стандартної бібліотеки мови, обмежену версію Django. Пропонується набір API для служб сховища, datastore API (BigTable), акаунтів Google, завантаження даних по URL, електронної пошти. Надано можливість використовувати планувальник завдань cron як для додатків, реалізованих на Python, так і на Java. Дозволено планування не більше 20-ти завдань. Використання служби облікових записів Google дозволяє швидко почати роботу з додатком, немає необхідності проводити окрему реєстрацію облікових даних на кожному сайті. Це також дозволяє розробнику не піклуватися про реалізацію ще однієї системи реєстрації користувачів спеціально для свого застосування. Платформа Google конкурує з аналогічними сервісами від Amazon, які надають можливості розміщувати файли і вебдодатки, використовуючи свою інфраструктуру [70].

На відміну від багатьох звичайних розміщень додатків на віртуальних машинах, таких як Amazon EC2, платформа App Engine тісно інтегрована з додатками і накладає на розробників деякі обмеження. Конкуруючі середовища дозволяють оперувати безліччю програмного забезпечення, створеного під *NIX системами, в той час як App Engine вимагає від розробника обов'язкового використання мов програмування Python або Java і збереження інформації у власному сховищі (Datastore) – підмножині бази даних BigTable.

Хоча більшість з розроблених на мові Python додатків можливо запустити без змін під App Engine (виключаючи ті, які вимагають доступу до низькорівневих функцій системи або доступу до мережі), використання сховища потребують значних сил від розробників на адаптацію додатків. На відміну від популярних СУБД MySQL і PostgreSQL, сховище не є реляційною базою даних, зокрема не підтримує схему і не вимагає нормалізації даних. Архітектура платформи підтримує масштабування без

зміни коду. Реалізація підтримки реляційної логіки призвела б до значного уповільнення роботи. Нереляційний характер сховища вимагає від розробників використання нетрадиційної архітектури зберігання і обробки даних. Наприклад, замість нормалізованого зберігання даних в двох або декількох таблицях з подальшим об'єднанням в запиті (join), логічно пов'язані дані рекомендується зберігати в одній таблиці [69].

GAE підходить для практично будь-яких вебдодатків, сайтів та мобільних бекендів. Коли мова йде про мобільний бекенд, то мається на увазі сервер, що надає якийсь API для додатків або прикладних програм. GAE також підходить для завдань збирання та обробки даних, так само буде відмінним рішенням у задачах взаємодії типу сервер-сервер [20].

GAE відрізняється від стандартних рішень, для його характеристики можна розглянути три слогани: Zero to sixty, Supercharged APIs, You're in control.

Кожен з цих слоганів характеризує основні відмінності від стандартних рішень. До прикладу, перший «Zero to sixty» означає, що можна не турбуватися про горизонтальне масштабування. Відповідно немає потреби в ручному контролі різних серверів. І, дійсно, GAE не надає засобів для ручного налаштування серверів. Єдине, що доступне користувачеві - це тільки його додаток.

Другий слоган «Supercharged APIs» описує основний спосіб взаємодії програми та інфраструктури. Фактично додаток буде жити в якійсь пісочниці і користуватися різними сервісами через їх API. Це дуже зручно, так як досягається поділ обов'язків, що, в свою чергу, мінімізує можливість помилки з боку програміста.

Третій слоган «You're in control» означає, що користувач буде все контролювати: споживання ресурсів, статистику та інше через спеціальний веб-сервіс. На практиці дається два важливі налаштування, це час відгуку вашого додатка і скільки примірників додатку будуть перебувати в очікуванні запитів. За замовчуванням вони стоять в налаштуваннях

автоматично. У майбутньому ці налаштування допоможуть або домогтися збільшення швидкості, або знизити витрати на оплату, але не разом. Потрібно відзначити, це стосується frontend примірників додатку, інші сервіси теж можна налаштовувати в певних межах [20].

Є кілька важливих особливостей, які потрібно враховувати – вони відрізняють середовище GAE від інших вебфреймворків (GAE фреймворком і не є – це ціла інфраструктура):

– До інших комп'ютерів в мережі можна достукатися тільки і виключно по HTTP (s). Вірно і зворотне – додатки працюють на GAE доступні тільки по HTTP (s) і тільки по стандартних портах.

– Немає write-доступу до файлової системи. Максимум, що може робити додаток - це читати файли, які були завантажені разом з самим додатком. Для організації зберігання даних слід використовувати або GAE Datastore або memcache.

– Є три сценарії роботи програми: відповідь на HTTP (s) запит, в окремому випадку - запит користувача; запуск завдань за розкладом (cron); черга завдань.

Важливо відзначити, що два останніх типи є підтипами першого - будь-яке завдання запускається як HTTP запит за певною адресою, різниця лише в тому, хто його запускає – користувач, менеджер черги завдань або cron. Тривалість роботи одного завдання не повинна перевищувати 30 секунд [22].

Для успішної роботи з платформою необхідно дотримуватися ідіом платформи, що відображено у документації та прикладах в ній. Варто дотримуватися принципу мінімізації побічних ефектів і відмови від стану. Таким прикладом є протокол HTTP, який є stateless. На практиці це буде виглядати так: більшість запитів будуть безпечними, а деструктивні запити будуть виділятися в спеціальну групу, тобто вони не будуть залежати від перших.

Також важливо звернути увагу на деякі компоненти платформи, такі як users і datastore. Компонент users відповідає за аутентифікацію, але не за

авторизацію. Щодо компоненту datastore дійсними є наступні твердження: дані зберігаються в datastore як є, і щоб просто додати одне поле в сутність, це потрібно робити через код; вручну потрібно робити трансформації даних, якщо це необхідно, за додаткову плату; досить важко відображати відносини між об’єктами. В цілому, якщо потрібно зберігати об’єкти в плоскому вигляді без залежностей, то це робиться дуже просто. Типовий патерн - список статей, список коментарів.

Платформа AWS Elastic Beanstalk від сервісу Amazon Web Services та Azure Cloud Services від сервісу Microsoft Azure є основними аналогами платформи GAE. Дані сервіси важко порівнювати, тому що кожен має свої особливості, однак все ж таки варто зробити невеликий огляд.

AWS Elastic Beanstalk – це простий у використанні сервіс для розгортання та масштабування вебдодатків та сервісів, розроблених з використанням Java, .NET, PHP, Node.js, Python, Ruby, Go і Docker, на серверах Apache, Nginx, Passenger та IIS. Від користувача вимагається завантажити код, а Elastic Beanstalk автоматично виконає розгортання: виділить ресурси, займеться балансуванням навантаженням, автоматичним масштабуванням і моніторингом справності роботи програми. При цьому користувач зберігає повний контроль над ресурсами AWS для додатка і в будь-який час можете отримати до них доступ [36].

Компанія Amazon об'єднує свою інфраструктуру як сервісну пропозицію в чотири категорії обчислень, зберігання та доставки вмісту, бази даних та мереж. Всі ці ресурси використовуються за умови забезпечення служб безпеки та ідентифікації Amazon, що включають в себе Amazon розміщує Active Directory, AWS Identity Management, менеджер сертифікатів AWS для керування сертифікатами SSL/TLS і навіть апаратну кешпам'ять та управління через AWS CloudHSM. Є можливість відстежувати використання ресурсу інфраструктури за допомогою таких інструментів управління, як Amazon CloudWatch, AWS Cloudtrail для відстеження активності

користувачів та використання API, а також AWS Config для відстеження інвентаризації ресурсів та змін [38].

Azure зможе охопити всі ці широкі категорії (Compute) управління даними (включаючи базу даних), Performance і Networking. Він забезпечить ці послуги за допомогою комбінації Azure Active Directory, служби Active Directory, Multi-Factor Auth та складної моделі керування доступом на основі ролі, яка використовується для розширення сучасної архітектури безпеки в хмарі. Azure також має ряд послуг та інтеграції для поглибленого моніторингу та оповіщення про показники та журнали ефективності інфраструктури [38].

1.3 Основні типи невизначеностей у медичних застосуваннях

Невизначеність – це загальне явище МН, яке можна виявити на кожному етапі навчання, наприклад, попередня обробка даних, вибір алгоритму та моделі. Представлення, вимірювання та обробка невизначеності мають значний вплив на ефективність навчальної моделі.

Невизначеність класифікують на дві категорії [56], [126]: алеаторна та епістемічна. Таку класифікацію особливо часто застосовують у медичних застосування.

Алеаторна невизначеність також відома як статистична невизначеність, вона відображає невідомі, які відрізняються кожного разу, коли проводиться один і той же експеримент. Наприклад, постріл однієї стріли з механічного лука, який точно дублює кожен запуск (однакове прискорення, висота, напрямок і кінцева швидкість), не буде впливати на попадання в одну і ту ж точку на цілі через випадкові та складні вібрації стріли, знання про які неможливо визначити в достатній мірі для усунення наслідків розсіювання точок удару. Те, що немає змоги виміряти в достатній мірі за допомогою наявних на даний момент вимірювальних приладів, не виключає існування такої інформації. *Aleatoric* походить від латинського *alea* або *кістки*, посилаючись на азартну гру.

Епістемічна невизначеність також відома як систематична невизначеність, і вона пов’язана з речами, які можна було б принципово знати, але на практиці не можна. Це може бути тому, що вимірювання не є точним, тому що модель нехтує певними ефектами, або тому, що певні дані навмисно були приховані. Прикладом джерела цієї невизначеності може бути опір у експерименті, призначеному для вимірювання прискорення сили тяжіння поблизу земної поверхні. Звичайно використовуване гравітаційне прискорення ігнорує вплив опору повітря, але опір повітря для об'єкта можна виміряти та включити в експеримент, щоб зменшити результуючу невизначеність при розрахунку гравітаційного прискорення.

Алеаторна невизначеність приписується результатам, які з практичних цілей не можуть бути передбачені, і тому їх розглядають як стохастичні, наприклад, результат підкидання монети, тоді як епістемічна невизначеність пояснюється відсутністю інформації або знань, наприклад, чи правильно хтось відповів на питання іспиту.

1.4 Алгоритмічні питання задач машинного навчання в умовах невизначеності

1.4.1 Загальний огляд типових алгоритмів прийняття рішень

Дерево рішень. Дерево рішень [2] – це спосіб подачі вирішальних правил в ієрархічній структурі, що складається з елементів двох типів - вузлів (node) і листів (leaf). У вузлах знаходяться вирішальні правила і проводиться перевірка відповідності прикладів цього правила з якого-небудь атрибуту навчальної множини. У найпростішому випадку, в результаті перевірки, безліч прикладів, які потрапили в вузол, розбиваються на дві підмножини, в одну з яких потрапляють приклади, що задовільняють правило, а в іншу – що не задовільняють.

Потім до кожної підмножини знову застосовується правило і процедура рекурсивно повторюється, поки не буде досягнута деяка умова зупинки

алгоритму. У результаті, в останньому вузлі перевірка і розбиття не проводиться, він оголошується листом. Лист визначає рішення для кожного прикладу, який в нього потрапив. Для дерева класифікації – це клас, що асоціюється з вузлом, а для дерева регресії - відповідний листу модальний інтервал цільової змінної.

Таким чином, на відміну від вузла, в листі міститься не правило, а підмножина об'єктів, які відповідають всім правилам гілки, яка закінчується даними листом.

Очевидно, щоб потрапити в лист, кожне тестове значення повинне відповідати всім правилам, які лежать на шляху до цього листа. Оскільки шлях в дереві до кожного листа єдиний, то й кожне тестове значення може потрапити тільки в один лист, що забезпечує єдність розв'язку. Основна сфера застосування дерев рішень – підтримка процесів прийняття управлінських рішень, що використовується в статистиці, аналізі даних і МН. Завданнями, які розв'язуються за допомогою даного апарату, є:

– Класифікація – віднесення об'єктів до одного з заздалегідь відомих класів. Цільова змінна повинна мати дискретні значення.

– Регресія (чисельне пророкування) – прогноз числового значення незалежної змінної для заданого вхідного вектора.

– Опис об’єктів – набір правил в дереві рішень, що дозволяє компактно описувати об’єкти. Тому замість складних структур, що описують об’єкти, можна зберігати дерева рішень.

У ході побудови дерева рішень потрібно розв’язати кілька основних задач, з кожною з яких пов’язаний відповідний крок процесу навчання:

– вибір атрибута, за яким буде проводитися розбиття в даному вузлі (атрибута розбиття);

– вибір критерію зупинки навчання;

– вибір методу відсікання гілок (спрощення);

– оцінка точності побудованого дерева.

Процес побудови дерев рішень полягає в послідовному, рекурсивному розбитті навчальної множини на підмножини з застосуванням вирішальних правил у вузлах. Процес розбиття триває до того часу, поки всі вузли в кінці всіх гілок не будуть оголошені листами. Оголошення вузла листом може статися природним чином (коли він буде містити єдиний об'єкт, або об'єкти тільки одного класу), або після досягнення деякої умови зупинки, що задається користувачем (наприклад, мінімально допустиму кількість прикладів у вузлі або максимальну глибину дерева).

Алгоритми побудови дерев рішень відносять до категорії так званих «жадібних» алгоритмів. «Жадібними» називаються алгоритми, які допускають, що локально-оптимальні рішення на кожному кроці (розбиття в вузлах) призводять до оптимального підсумкового рішення. У випадку дерев рішень це означає, що якщо один раз був обраний атрибут, і по ньому було здійснено розбиття на підмножини, то алгоритм не може повернутися назад і вибрати інший атрибут, який дав би краще підсумкове розбиття. Тому на етапі побудови не можна визначити чи забезпечить обраний атрибут, в кінцевому підсумку, оптимальне розбиття.

Алгоритми класифікації C5.0 [89], [87], [132], CLS [79] та ID3 [152] на основі дерева рішень отримали широке застосування.

Випадковий ліс. Випадковий ліс [3] – це безліч дерев рішень. У задачі регресії їх відповіді усереднюються, в задачі класифікації приймається рішення голосуванням більшістю. Всі дерева будуються незалежно за наступною схемою:

– Вибирається підвибірка навчальної вибірки розміру samplesize – по ній будується дерево (для кожного дерева - своя підвибірка).

– Для побудови кожного розбиття в дереві переглядається max_features випадкових ознак (для кожного нового розбиття - свої випадкові ознаки).

– Вибирається найкраща ознака і розбиття відбувається по ній (по заздалегідь заданому критерію). Дерево будується, як правило, до вичерпання вибірки (поки в листі не залишаться представники тільки одного

класу), але в сучасних реалізаціях є параметри, які обмежують висоту дерева, число об'єктів в листі і число об'єктів в підвибірці, при якому проводиться розбиття.

Алгоритм k-середніх. Алгоритм k-середніх – це простий ітеративний метод для розподілу набору даних на певну кількість кластерів *k*. Цей алгоритм був виявлений кількома дослідниками різних дисциплін, зокрема Ллойд [97], Форгей, Фрідман і Рубін, Маккуїн. Детальна історія алгоритму *k*-середніх разом з описаними декількома інтерпретаціями наведена в [81]. Грей і Нейхофф [71] представляють історичний розвиток алгоритму k-середніх, розміщених у більш широкому контексті алгоритмів сходження на верх.

Алгоритм працює на наборі *d*-вимірних векторів, $D = \{x_i \mid \mathrm{i} = 1, ..., \mathrm{N}\}$, де $x_i \in \Re^d$ позначає i-ту точку даних. Алгоритм ініціалізується шляхом вибору точок k в $\Re^d$ як початкового представника кластерів k або «centroids». Способи вибору цих початкових даних включають випадкове вибіркове взяття з набору даних, встановлюючи їх як розв’язок кластеризації невеликої підмножини даних або збурювання глобального середнього значення даних k разів.

Метод опорних векторів. У сучасних програмах машинного навчання, метод опорних векторів (ОВМ) [181] вважається обов’язковим для спроб, він пропонує одне з найбільш надійних і точних методів у всіх відомих алгоритмах. Цей метод має гарну теоретичну основу, вимагає всього лише десятка прикладів для навчання і нечутливий до кількості вимірів. Крім того, ефективні методи навчання ОВМ також розвиваються у швидкому темпі.

У двокласній задачі навчання мета ОВМ полягає в тому, щоб знайти найкращу класифікаційну функцію для відмінності між членами двох класів у навчальних даних. Метрика концепції «найкращої» класифікації функції може бути реалізована геометрично. Для лінійно відокремлюваного набору даних лінійна класифікаційна функція відповідає роздільній гіперплощині

$f(x)$, яка проходить через середину двох класів, розділяючи їх. Як тільки ця функція буде визначена, новий екземпляр даних x_n може бути класифікований шляхом простого тестування знаку функції $f(x_n)$; x_n належить до позитивного класу, якщо $f(x_n) > 0$.

Наївний байєсівський класифікатор. З огляду на набір об'єктів, кожен з яких належить до відомого класу, і кожен із яких має відомий вектор змінних, мета полягає в тому, щоб побудувати правило, яке дозволить призначити майбутні об'єкти класу, надаючи лише вектори змінних, що описують майбутні об'єкти. Задачі такого роду, що називаються задачами класифікації з вчителем, є широкомасштабними і розроблено багато методів побудови таких правил. Одним з важливих є наївний метод Байєса, який також називають простим Байєсом та незалежністю Байєса. Цей метод важливий з кількох причин: його дуже легко побудувати, не потребуючи складних ітераційних схем оцінювання параметрів; легко застосувати до величезних наборів даних; легко інтерпретувати; він може бути не найкращим класифікатором в конкретному випадку, але, як правило, він буде надійним. Загальне дослідження наївного методу Байєса подано в [57], [73].

Ансамбль методів. Ансамблеві методи швидко розвиваються протягом десятиліть, оскільки продуктивність завжди перевершує показники інших методів класифікації [194], [167]. Це стало однією з головних областей МН та розпізнавання образів. Основним процесом побудови системи класифікаторів ансамблевих методів є використання базових класифікаторів з різними підмножинами одного набору навчальних даних, а потім поєднання вичерпності базових класифікаторів цих варіантів [179]. У дослідженні ансамблевих методів, точність базового класифікатора та різноманітність між різними базовими класифікаторами є двома основними факторами. Це призводить до кращої класифікації з вищою індивідуальною точністю та більшою різноманітністю. Серед ансамблевих методів найпопулярнішими є:

Bagging [39], AdaBoost [64], Random Subspace [75], Random Forest [40] та Rotation Forest [84].

1.4.2 Наукове машинне навчання

Наукове машинне навчання – це нова сфера досліджень, зосереджена на можливостях та викликах МН в контексті складних застосувань у науці, техніці та медицині. Найбільші проблеми використання у цих областях мають атрибути, які роблять їх дуже різними за своєю суттю до застосувань у комп’ютерних науках, де МН на основі даних досягло успіху [46].

Ці проблеми характеризуються:

– складною багатомасштабною мультифізичною динамікою: невеликі зміни параметрів системи можуть призвести до різко великих змін у реакціях системи;

– мають невизначені параметри великих розмірів, які неможливо спостерігати безпосередньо;

– наукових та технічних даних часто не вистачає: експерименти є дорогими, трудомісткими, а іноді і небезпечними;

– найчастіше найважливіші рішення визначають рідкісні події (наприклад, невдачі). Для деяких застосувань такі важливі рішення повинні прийматися в режимі реального часу, не залишаючи часу для перенавчання моделей нещодавно відмічених критичних даних.

Наукове МН поєднує диференційоване програмування, наукове моделювання та машинне навчання. Основними обчислювальними засобами наукового МН є нейронні мережі та бібліотеки для числових диференціальних рівнянь [153], [82].

1.4.3 Прийняття рішень в умовах невизначеності на основі мінімаксного підходу

Мінімаксний підхід (разом із максиміном та максимаксом) традиційно застосовується для задач регресії [138], [128]. У випадку МН один із

узагальнених мінімаксних підходів відомий як Minimax Probability Machine (MPM) [91]. Можна стверджувати, що MPM є класичним результатом з вивчення надійності інтелектуальних моделей [136], [138], який можна розглядати як типовий метод класифікації надійності навчання. Завданням оптимізації MPM є мінімізація верхньої межі ймовірності неправильної класифікації вивчення параметрів моделі. Верхню межу ймовірності неправильної класифікації можна використовувати як явний показник для оцінки надійності класифікаційних моделей. Версія MPM з параметричною редукцією була запропонована в [91], [90] для нелінійних задач класифікації. Декілька вдосконалених алгоритмів MPM були представлені з різних точок зору [54], [77], [76], [171]. У [77], [76] вказувалося, що в деяких випадках слід розрізняти ймовірність неправильної класифікації двох класів, оскільки один клас може бути важливішим за інший. У [171] MPM було розширено для регресії. У роботі [54] MPM було введено для підготовки нечіткого класифікатора TSK до більш прозорої та зрозумілої моделі класифікації. На додаток до MPM, вивчення надійності інтелектуальних моделей розглядалося з інших точок зору. Наприклад, поняття «конфлікт» та «незнання» були введені для позначення надійності класифікаційних моделей у [100], [101].

Щоб зробити метод прийняття мінімального ймовірного методу доступним для навчання додаткових інтелектуальних моделей та реалізації вивчення надійності цих моделей, пропонується узагальнена прихована мінімальна машина ймовірностей (Generalized Hidden-Mapping Minimax Probability Machine (GHM-MPM)).У роботі [55] верхня межа ймовірності неправильної класифікації MPM була використана як явний показник для характеристики надійності класифікаційної моделі.

Рис. 1.3 використовується для ілюстрації функції прийняття рішення. Вхідний вектор x можна відобразити у прихований простір через $\beta(\cdot)$ і підставити в MPM. Якщо вихідний результат у дорівнює -1, тоді x присвоюється негативному класу, в іншому випадку він віднесений до позитивного класу. Тут зовнішній $\beta(\cdot)$ є запасною частиною, тоді як

внутрішній MPM є фіксованим. Різниця лише у $\beta(\cdot)$, а обробка в MPM однакова.

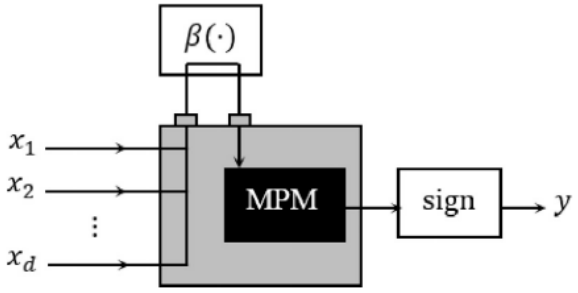


Рисунок 1.3 – Функція прийняття рішення Generalized Hidden-Mapping Minimax Probability Machine (GHM-MPM)

Недоліком MPM є те, що він не включає термін регуляризації для побудови розділової гіперплощини. Регуляризація використовується для уникнення неправильно поставлених задач, а також для того, щоб добре підігнати навчальні зразки, одночасно знижуючи ризик перенавчання [180]. Наприклад, регуляризація Тихонова, або l_2 -норма, широко використовується в навчальних машинах; стратегія, популяризована методом опорних векторів (ОВМ) [180]. Ця стратегія є однією з найпопулярніших методів класифікації та широко використовується в таких сферах, як комп'ютерний зір [96], [172], медична діагностика [37], [183] та бізнес-аналітика [49], [106]. Сакета Натх (Saketha Nath) і Бхаттачарія (Bhattacharyya) [139] запропонували регульовану альтернативу MPM, у якій l_2 -норма була мінімізована. Однак цей підхід використовує фіксовані значення для коефіцієнтів неправильної класифікації, а не оптимізує ці методи із завданням оптимізації.

Іншим недоліком MPM є те, що він передбачає унікальне відкликання класів для обох класів. Передбачається, що обидва класи однаково важливі. Машина мінімальних помилок мінімальної ймовірності (Minimum Error Minimax Probability Machine, MEMPM) [78] – це відповідне розширення MPM, яке мінімізує опуклу комбінацію коефіцієнтів неправильної класифікації, збалансовуючи два класи на основі їх попередніх ймовірностей.

Пропонується розглянути регуляризовані рецептури MPM та MEMPM, в яких l_2 - норма вектора мінімізується разом із коефіцієнтами помилок. На відміну від моделі, запропонованої Saketha Nath та Bhattacharyya [139], ці показники є частиною задачі оптимізації, що веде до нелінійних формулювань конусного програмування другого порядку (NSOCP).

Незважаючи на те, що MPM демонструє чудові результати узагальнення, він зазвичай вирішується розв’язуванням задачі програмування конусів другого порядку (SOCP). Коли кількість зразків збільшується, це збільшує складність його обчислення, що значно обмежує його придатність. Тому, спираючись на MPM та TELM, пропонується парна мінімаксна ймовірність екстремального нахилу машини (TMPELM) для розпізнавання шаблонів. Подібно до TELM, TMPELM також шукає дві непаралельні розділяючі гіперплощини в просторі об'єктів TELM. Для кожної гіперплощини TMPELM намагається максимізувати ймовірність правильного класу класифікації зразків, тобто мінімізувати найгірший випадок (максимальну) ймовірність неправильної класифікації класу зразків, тоді як відстань до іншого класу якомога більша.

У роботі [192], [191] була запропонована машина подвійної мінімаксної ймовірності (TWMPM), яка поєднує в собі переваги як MPM, так і подвійної векторної машини підтримки. Ma (Jun Ma) та інші дослідники вдосконалили цей підхід до MPM, який може безпосередньо оцінювати точність ймовірності, пов'язану із задачами класифікації [105] та регресії [104], і таким чином робить модель MH більш прозорою.

У роботі [68] розглянуто функціональний градієнтний підхід до імовірнісного мінімаксу активного навчання, де розглядається імовірнісний мінімакс активного навчання. Зокрема, нехай $(x, y) \in R^d \times \{-1,1\}$ та Y, X - відповідні їм випадкові величини. Крім того, нехай $p(Y = y \mid X = x)$ є умовною ймовірністю мітки y наведеного прикладу x. Простим класифікатором, заснованим на цій умовній ймовірності, є:

$$f(x)=\begin{cases}1 & p(Y=1\mid X=x)\geq\frac{1}{2}\\ -1 & p(Y=1\mid X=x)<\frac{1}{2}\end{cases} \tag{1.1}$$

Прикладами з найбільшою невизначеністю є такі, що $p(Y=1\mid X=x)$ найближче до $\frac{1}{2}$. Будь-які оцінки $p(Y=1\mid X=x)$, що дорівнюють $\frac{1}{2}$, свідчать про те, що мітка точки є випадковою. Враховуючи налаштування активного навчання, де маркованих даних мало і є багато немаркованих даних, неоптимально вибирати точки, оцінка яких $p(Y=1\mid X=x)$ наближається до $\frac{1}{2}$. Бажано вибрати метод, заснований на обчисленні граничних значень для класифікації, а не просто поріг, що дорівнює $\frac{1}{2}$ [127].

У прикладі $x_i, i\in[1,n]$ найкращою міткою є $y_i^b=\arg\max_{y_i} p(\mathrm{Y}=y_i\mid X=x_i)$ відповідно до поточної оцінки умовної ймовірності; найгірша позначка – $y_i^{\omega}=\arg\max_{y_i} p(\mathrm{Y}=y_i\mid X=x_i)$ щодо поточної оцінки. Наприклад, коли умовні ймовірності цих двох міток далекі одна від одної, правильною міткою, безумовно, є $\arg\max_{y_i} p(\mathrm{Y}=y_i\mid X=x_i)$. В іншому випадку правильна мітка не визначена. Одним із способів знайти такий приклад є пошук такого з максимальною ймовірністю гіршого позначення, тобто

$$q=\arg\max_{q}\min_{y_q} p(\mathrm{Y}=y_q\mid X=x_q) \tag{1.2}$$

Однак ця модель не бере до уваги інші немарковані дані. Якщо припустити, що точність умовної оцінки ймовірності буде підвищена шляхом врахування немаркованих даних, інтуїтивно зрозумілою цільовою функцією ймовірнісного мінімаксного активного навчання (Probabilistic Minimax Active Learning, PMAL) є:

$$q=\arg\max_{q}\min_{y_q}\max_{y_{u-q}} p(y\mid x) \tag{1.3}$$

Де u - набір немаркованих даних, $x=[x_{1,}x_2,...,x_n]$ та $y=[y_{1,}y_2,...,y_n]$.

Під час підготовки класифікатора Теварі та Бартлетт (Tewari and Bartlett) [174] доводять, що розрідженість не виникає, якщо умовні ймовірності можна точно оцінити. Для набору втрачених функцій вони забезпечують точний компроміс між можливістю оцінки умовної ймовірності та розрідженістю рішення. У статті [173] продемонстровано, що задача класифікації визначає унікальну умовну ймовірність, яку опукла мінімізація ризику знаходить для неї. Крім того, дослідження [127] емпірично показує, що $p[y=1|x]$ не може бути оцінено Logit-Boost та AdaBoost, оскільки ці два методи перевантажують навчання $p[Y=1|X=x]$, навіть незважаючи на те, що обидва ефективні як класифікатори. У [195] зазначено, що всі емпіричні алгоритми мінімізації ризиків, такі як ОВМ та логістична регресія, мінімізують функцію відстані оцінки умовної ймовірності даного прикладу міток до її справжнього невідомого $p[Y=y|X=x]$. Оскільки багато з цих алгоритмів, таких як ОВМ, є універсальними [170], PMAL також вважається універсальним, тобто він вибирає приклад, який мінімізує ризик класифікатора будь-якого розподілу даних.

AdaBoost був вперше представлений для підвищення точності набору простих правил великих вирівнювачів або слабких класифікаторів з метою отримання сильного класифікатора із задовільними властивостями узагальнення [161]. У багатьох задачах навчання, порівняно легко знайти просте емпіричне правило з низькою точністю, дуже важко отримати класифікатор з високою точністю. Наприклад, якщо в документі є слово «футбол» як слабкий класифікатор, документ може стосуватися спорту. Тобто, в кожній ітерації AdaBoost знаходить слабкий класифікатор для неправильно класифікованих точок даних у навчальному наборі і додає цей слабкий класифікатор до зваженої суми раніше обраних слабких класифікаторів [161].

AdaBoost можна трактувати по-різному. Кожне з цих уявлень про посилення забезпечує багате теоретичне розуміння навчальних задач. Однією з них є перспектива функціонального градієнта Фрідмана [65], який пропонує

більш загальне розуміння підсилення і може бути використаний для виведення його нових варіантів підсилення [144]. Дослідження Bühlmann та Yu [42] виявляє, що, хоча дисперсія або складність L2boosting для лінійних моделей зростає на кожній ітерації, величина збільшення дисперсії експоненційно зменшується у числах ітерацій. Це є причиною того, чому підсилювач стійкий до перенавчання. Хоча підсилення можна розглядати як підхід, заснований на секціях, таких як ансамбль або методи пакунків, він принципово відрізняється [74].

Мета дослідників у роботі [107] полягала у підвищенні точності будь-якого даного класифікатора, використовуючи немарковані дані. Для покращення навчання з вчителем вони пропонують алгоритм посилення, який називається SemiBoost [107]. SemiBoost використовує міру схожості між маркованими даними та немаркованими, щоб призначити псевдомітку для будь-яких немаркованих даних. Присвоєння псевдоміток немаркованим даним було запропоновано в методах самонавчання, ASSEMBLE та Semisuperposed MarginBoost [107]. За таких підходів запас, як правило, покращується, але ніякої нової інформації класифікатор надавати не буде. Вибрані приклади для призначення псевдоміток насправді далекі від межі розв’язку. Перепідготовка з нещодавно присвоєними мітками підвищує рівень їхньої класифікації ще вище [107]. На відміну від них, автори у [107] порівнюють приклади з маркованими та немаркованими даними та використовують подібність між прикладами, щоб призначити їм псевдомітки.

Цікавим підходом до маркованих даних є вибір немаркованих даних, у мітках яких класифікатор є найбільш точним, і присвоєння цих міток, які називаються псевдомітками, до цих вибраних немаркованих даних [107]. Всебічне дослідження методів, заснованих на цьому підході, їх властивості та експериментальні результати відображено у [175]. У цій перспективі PMAL зображується як метод, який розглядає псевдомітки для немаркованих даних і вибирає найгіршу мітку для вибраного прикладу за цільовою функцією.

1.5 Постановка наукового завдання дисертаційного дослідження

На основі проведеного огляду попередніх результатів встановлено актуальність дослідження щодо врахування невизначеностей у задачах класичного та глибокого МН. Попри наявність робіт, що стосуються мінімаксного МН, на сьогодні відсутня загальна схема побудови моделі МН з урахуванням алеаторних і епістемічних невизначеностей, що є особливо важливим у клінічних застосуваннях. До того ж є проблема у розробках технологій МН на основі хмарних платформ з доступом до електронних медичних записів. Усе це веде до необхідності вирішення ряду задач розробки програмного та математичного забезпечення інформаційних технологій, пов’язаних із прийняттям рішень в системних медичних дослідженнях, а саме реалізація алгоритмів МН на основі хмарних обчислень, розробка методів на основі мінімаксного підходу до вирішення задач класичного та глибокого МН, розвиток методів наукового МН в кіберфізичних системах медико-біологічних процесів.

Висновки до першого розділу

1. Проаналізовано основні області використання і можливості Big Data. Розглянуто методики аналізу великих даних. Визначено, що особливої уваги заслуговує аналіз даних у медицині, так як це одна із найбільш важливіших галузей для людства і відповідно ефективне і правильне опрацювання даних у цій галузі є одним із найпріоритетніших завдань. Варто відзначити, що у зв’язку із зростанням обсягу даних, також паралельно активно розвиваються технології опрацювання великих даних, мова йде про МН, що дало можливість здійснити багато теоретичних проривів і застосовується зараз фактично у всіх сферах діяльності людини, де відбувається опрацювання різного роду інформації.

2. Розглянуто технологічні питання задач МН на основі хмарних обчислень. Здійснено загальний огляд хмарних платформ. Описано причини переходу до парадигми хмарних обчислень, з яких можна виділити зменшення витрат, оптимальне використання персоналу та ресурсів, надійна масштабованість.

3. Розглянуто сервіси хмарних платформ та необхідні дії з точки зору розгортання додатків. Наприклад, основними службами у GCP є: Compute Engine, App Engine Standard Environment, App Engine Flexible Environment, App Engine, Cloud SQL, Cloud Spanner, Cloud Datastore, Cloud Bigtable та інші. Якщо мова йде про створення та розгортання вебдодатків, то особливу увагу слід звернути на Google App Engine – це гнучка платформа як служба (PaaS), сервіс хостингу сайтів і вебаплікацій на серверах Google з безкоштовним ім'ям в домені appspot.com або з власним ім'ям. Для подальшої роботи обрано Google Cloud Platform, оскільки дана платформа хмарних обчислень володіє усім необхідним для реалізації алгоритму індукції дерева рішень C5.0.

4. Здійснено аналіз сервісу Google App Engine – складової Google Cloud Platform, що обрано для подальшої роботи. Google App Engine – це система розробки хмарних додатків від Google, створена на основі інструментів, які

використовуються самим Google в інфраструктурі своїх прикладних програм. App Engine дозволяє розробникам створювати в Python і Java масштабовані додатки, не задумуючись про низькорівневу архітектуру. Сервіс Google App Engine необхідний для реалізації алгоритму індукції дерева рішень у системних медичних дослідженнях.

5. Наведено типові види невизначеностей у медичних застосуваннях: алеаторна та епістемічна, а також дано визначення поняття наукового машинного навчання.

6. Розглянуто алгоритмічні питання задач МН в умовах невизначеності. Зокрема розглянуто такі типові алгоритми прийняття рішень: дерево рішень, випадковий ліс, алгоритм k-середніх, метод опорних векторів, наївний баєсівський класифікатор, ансамблеві методи. Представлені алгоритми прийняття рішень можуть бути реалізовані на хмарних платформах.

7. Проведено огляд результатів в галузі МН на основі мінімаксного підходу. Описано важливість оптимізації, ефективність та дієвість алгоритмів чисельної оптимізації. Оптимізація є важливою частиною МН. Суть більшості алгоритмів МН полягає у побудові моделі оптимізації та вивченні параметрів у цільовій функції з поданих даних. В епоху великих даних ефективність та дієвість алгоритмів чисельної оптимізації різко впливає на популяризацію та застосування моделей МН. Проте є потреба у розробці методів на основі мінімаксного підходу для розв’язання задач класичного та глибокого МН.

8. На основі проведеного аналізу визначено наукове завдання, яке полягає у розробці програмного та математичного забезпечення інформаційних технологій, пов’язаного із прийняттям рішень в системних медичних дослідженнях. Є необхідність у реалізації алгоритмів МН на основі хмарних обчислень, розробці методів на основі мінімаксного підходу для розв’язування задач класичного та глибокого МН, розвитку методів наукового МН в кіберфізичних системах медико-біологічних процесів.

РОЗДІЛ 2. ТЕХНОЛОГІЧНІ ПИТАННЯ РЕАЛІЗАЦІЇ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ НА ОСНОВІ ХМАРНИХ ОБЧИСЛЕНЬ У ВИПАДКУ СИСТЕМНИХ МЕДИЧНИХ ДОСЛІДЖЕНЬ

У розділі розглянуто задачу індукції дерева рішень на основі хмарних платформ та питання архітектури систем прийняття рішень на основі системних медичних досліджень. Здійснено математичне представлення задачі індукції дерева рішень, наведено приклад реалізації індукції дерева рішень, що є прототипом для реалізації на основі хмарної платформи для системних медичних досліджень. Представлено розробку архітектури на основі мікросервісів, здійснено концептуальне представлення інформаційної моделі задачі прийняття рішень для невідкладної медичної допомоги при настанні серцевої недостатності.

Основні результати цього розділу опубліковано в роботах [119], [131], [130].

2.1 Задача індукції дерева рішень на основі хмарних платформ

2.1.1 Математичне представлення задачі індукції дерева рішень

Математично задача індукції дерева рішень формулюється наступним чином. У нас є набір D, що містить N наборів навчальних даних. У цьому випадку кожен *i*-й набір $(A_1^i, A_2^1, ..., A_p^i, C^i)$ складається з вхідних даних $A_1, ..., A_p$ – атрибутів та вихідних даних – атрибута класу C. Атрибути $A_1, ..., A_p$ можуть приймати числові та категоріальні значення. Атрибут класу C приймає одне K з дискретних значень: $C \in \{1, ..., K\}$. Мета полягає в передбаченні дерева рішень значення атрибута класу C на основі значень атрибутів $A_1, ..., A_p$. У цьому випадку ми повинні максимізувати точність передбачення атрибута класу, а саме $P\{C = c\}$ в кінцевих вузлах для довільного $c \in \{1, ..., K\}$. Алгоритми індукції дерев рішень автоматично

розділяють на вузли значення числових атрибутів A_i на два інтервали: $A_i \le x_i$ та $A_i > x_i$, категоріальні атрибути A_j – на дві підмножини: $A_j \in S_j$, $A_j \notin S_j$. Розділ числових атрибутів базується, як правило, на вимірах, заснованих на ентропії, або індексі Джині. Процес розділення повторюється, доки не буде покращено точність прогнозування. Останній крок передбачає видалення вузлів, щоб уникнути перенавчання моделі. У результаті повинні отримати набір правил, які починаються від кореня до кожного термінального вузла, містять нерівності для числових атрибутів та умови включення для категоріальних атрибутів.

За основу береться найпростіша рекурсивна процедура.

Генерація дерева рішень

Вхідні дані: *D* – набір навчальних наборів даних $(A_1^i, A_2^1, ..., A_p^i, C^i)$.

Метод:

1. Створити вузол *N*.
2. Якщо всі набори *D* належать до загального класу *C*, повернути вузол *N* як лист із назвою класу *C*.
3. Якщо список атрибутів (і отже *D*) порожній, повернути вузол *N* як лист із назвою найпоширенішого класу в *D*.
4. Застосувати алгоритм вибору атрибутів зі списку атрибутів і для набору *D* знайти «найкращий» атрибут розділу.
5. Видалити розділення атрибутів зі списку атрибутів.
6. Для кожної умови дільника *j* для атрибута поділу розглянути *D* – набір множин з D, що задовольняє умові ділення *j*.
7. Якщо *j* – порожнє, то приєднати до вузла *N* лист під заголовком найпоширенішого класу D_j, інакше – зв'язати вузол *N* рекурсивним викликом методу генерування дерева рішень із вхідними даними D_j та списком атрибутів.
8. Кінець циклу на кроці 6.

9. Повернути вузол N.

Основою алгоритму вибору атрибута на *j*-му кроці рекурсії є наступний інформаційний індекс:

$$Gain(A_i) = Info(D_j) - Info_{A_i}(D_j) \tag{2.1}$$

Де

$$Info(D_j) = -\sum_{k=1}^{K} p_k^j \log_2(p_k^j) \tag{2.2}$$

Інформація, необхідна для класифікації набору $(A_1, A_2, ..., A_p)$ в D_j

$$Info_{A_i}(D_j) = \sum_{l=1}^{K_i} \frac{\#(D_j^l)}{\#(D_j)} Info(D_l) \tag{2.3}$$

Інформація, необхідна для класифікації $(A_1, A_2, ..., A_p)$ в D_j після поділу D_j на підмножину D_j^l, що відповідає значенням атрибута A_j.

У формулі 2.2 імовірність належності D_j довільній множини C_{k,D_j} оцінюється як $p_k^j = \frac{\#(C_{k,D_j})}{\#(D_j)}$, де C_{k,D_j} – набір множин з D_j, для яких атрибут класу $C = k$. Тут $\#[\bullet)$ – кількість елементів у наборі.

У формулі 2.3 $\frac{\#(D_j^l)}{\#(D_j)}$ проводиться оцінка ймовірності того, що довільна множина D_j належить до множини D_j^l, де D_j^l – це множина множин D_j для яких атрибут $A_i = a_i^l$. Ось атрибут $A_i \in \{a_i^1, a_i^2, ..., a_i^{K_i}$ $A_i \in \{a_i^1, a_i^2, ..., a_i^{K_i}\}$.

Отже, $Gain(A_i)$ оцінює зменшення інформації, необхідної для класифікації довільного набору даних D_j, через відоме значення атрибута A_j. Таким чином, із доступних атрибутів на кожному вузлі дерева рішень для умови поділу слід вибрати атрибут A_{i^*} з найбільшим значенням $Gain(A_{i^*})$. У результаті цього вибору для завершення процесу класифікації набору даних D_j потрібно якнайменше інформації.

2.1.2 Прототип реалізації індукції дерева рішень на основі хмарної платформи для системних медичних досліджень

Для забезпечення методу база даних містить дві таблиці – таблицю *attribute*, призначену для зберігання інформації про атрибути та таблицю *categorized_data* – для наборів навчальних даних. Програмні класи включено до пакету *decision_tree.model*. Сюди входять beans-класи *Attribute*, *Attribute_for_list* та *CategorisedData* для роботи з даними відповідних таблиць. SQL-запити щодо отримання відповідних даних, включаючи розрахунки інформаційних показників реалізовано в класі *AttributeListPeer*.

З метою візуалізації представлення дерева використано клас *com.abego.TreeLayout* вільнорозповсюджуваної бібліотеки.

Ключовим в реалізації методу *Attribute_selection_method* є розрахунок інформаційних показників $Info(D_j)$ та $Info_{A_i}(D_j)$ на j-ому кроці рекурсії для атрибута A_i. Мова структурованих запитів SQL має досить гнучкі засоби для реалізації алгоритмів в галузі машинних знань. Так, використавши вкладені запити, псевдоніми та агрегативні функції, розраховуються показники $Info(D_j)$ та $Info_{A_i}(D_j)$.

Метод продемонстровано на прикладі щодо діагностування серцевої недостатності на основі даних 63-х пацієнтів навчально-практичного центру первинної медико-санітарної допомоги Тернопільського національного медичного університету імені І. Я. Горбачевського. Використано такі атрибути:

Таблиця 2.1 – Перелік атрибутів для аналізу серцевої недостатності

Позначення атрибута	Зміст атрибута
A1	What is age?
A2	What is sex?
A3	What is pulse?
A4	What is SAP?
A5	What is DAP?

Набори включають лише категоріальні (попередньо оброблені) дані. На рис. 2.1 представлене побудоване дерево рішень.

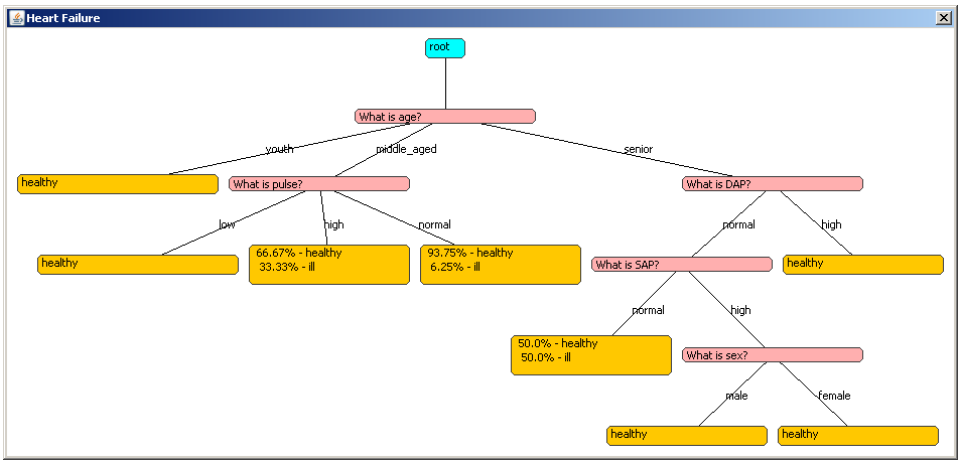


Рисунок 2.1 – Дерево рішень для діагностування серцевої недостатності на основі показника приросту інформації

Традиційний метод індукції дерева рішень грунтується на обчисленні показника приросту інформації $Gain(A_i)$, який має ухил в сторону атрибутів з більшою кількістю значень (виходів). Для позбавлення такого недоліку пропонується використання показника відношення приростів інформації. Він застосовує до приросту інформації свого роду нормалізацію. Означення вводиться за аналогією до $Info(D)$:

$$SplitInfo_{A_i}(D_j) = -\sum_{l=1}^{K_i} \frac{\#(D_j^l)}{\#(D_j)} \log_2 \left(\frac{\#(D_j^l)}{\#(D_j)}\right) \tag{2.4}$$

Таке значення представляє потенційну інформацію, яка генерується при поділі набору навчальних даних D_j на K_i розбиттів, що відповідають K_i виходам тесту щодо атрибуту A_i. Зауважимо, що для кожного виходу розглядається кількість записів, що відповідають умові даного виходу відносно загального числа записів в D_j. Це відрізняє його від показника приросту інформації, у якому вимірюється інформація щодо класифікації, яка отримується на основі того ж розбиття. Згідно означення відношення приростів становить:

$$GainRatio(A_i) = \frac{Gain(A_i)}{SplitInfo(A_i)} \tag{2.5}$$

Атрибут з максимальним значенням відношення приростів обирається як атрибут поділу. Для даного інформаційного показника досліджено той же приклад щодо діагностування серцевої недостатності. Побудоване дерево рішень представлене на рис. 2.2.

Оскільки індукція дерева рішень – це рекурсивний алгоритм, то доцільним є дослідження обчислювальної складності. Час виконання алгоритму індукції дерева рішень оцінюється величиною:

$$O(p \times \#(D) \times \log(\#(D))) \tag{2.6}$$

Дана оцінка була перевірена експериментально на клінічних даних – дані біохімічних досліджень у випадку політравм.

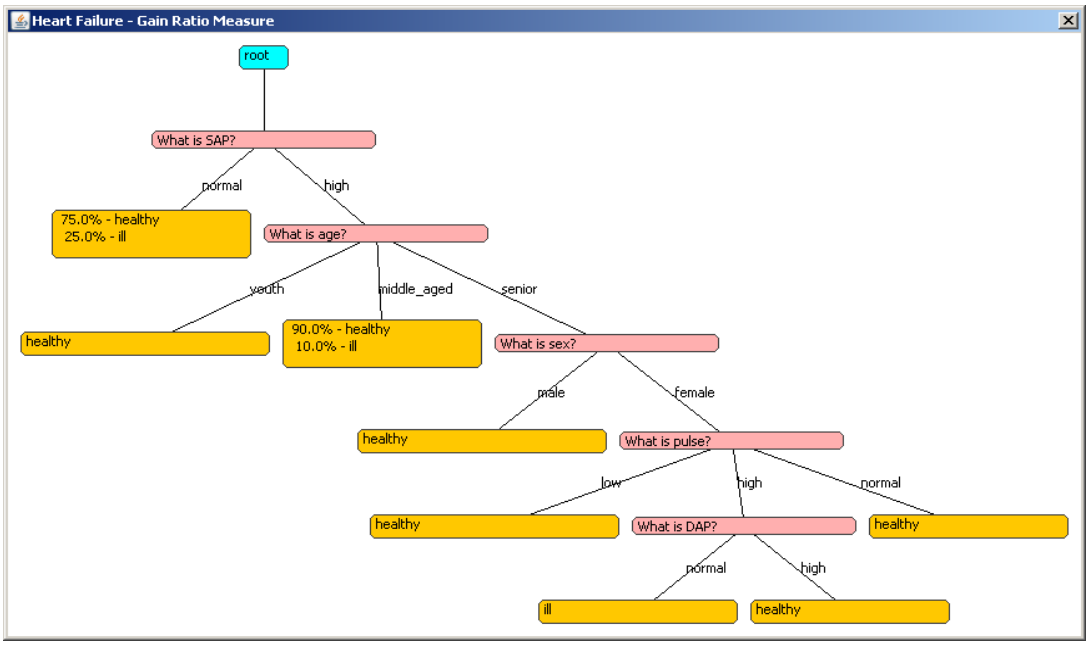


Рисунок 2.2 – Дерево рішень для діагностування серцевої недостатності на основі показника відношення приростів інформації

За допомогою розроблених програмних класів побудоване дерево рішень щодо класифікації політравм на основі даних 6-ти класифікаційних груп, а саме:

– черепно-мозкова та скелетна травми мали місце 2 години тому;

– черепно-мозкова та скелетна травми з кровотечею мали місце 2 години тому;

– черепно-мозкова та скелетна травми мали місце 12 годин тому;

– черепно-мозкова та скелетна травми з кровотечею мали місце 12 годин тому;

– черепно-мозкова та скелетна травми мали місце 24 години тому;

– черепно-мозкова та скелетна травми з кровотечею мали місце 24 години тому.

Використано таблицю атрибутів, що представляють дані біохімічних досліджень:

Таблиця 2.2 – Атрибути, що представляють дані біохімічних досліджень

Позначення атрибута	**Зміст атрибута**
A1	Weight
A2	Total.bil.
A3	AsAT
A4	AlAT
A5	GP
A6	GR
A7	VG
A8	SOD
A9	Catalaza
A10	MDA
A11	DK
A12	TsP
A13	TsIK
A14	API
A15	Ig A
A16	Ig M
A17	Ig G
A18	Il-2
A19	Il-6
A20	Il-10
A21	TNF-a

Побудоване дерево рішень представлене на рис. 2.3.

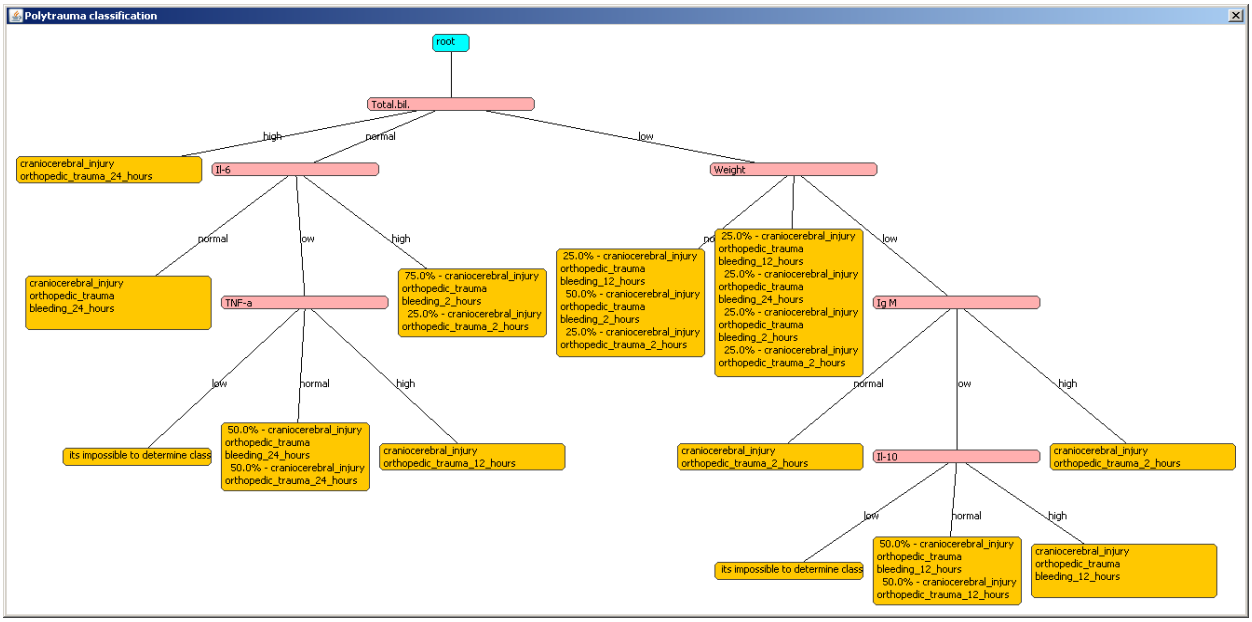


Рисунок 2.3 – Дерево рішень для класифікації політравм

2.2 Питання архітектури систем прийняття рішень на основі системних медичних досліджень

2.2.1 Питання архітектури систем прийняття рішень: розробка архітектури на основі мікросервісів

При розробці інформаційних систем для медичної практики та досліджень застосовуються різноманітні підходи та методи [103]-[114]. У першу чергу вони базуються на так званій монолітній архітектурі. У такому випадку розробляється серверний корпоративний додаток, який повинен підтримувати безліч різних клієнтів, включаючи настільні браузери, мобільні браузери та власні мобільні програми [142].

У свою чергу, мікросервіс – це парадигма, яка служить для організації та використання розподілених послуг, які можуть мати різні власники. Основні ідеї цього архітектурного підходу були викладені Мартіном Фаулером у 2014 році в [94].

Мікросервіси дозволяють будувати великі системи з ряду взаємодіючих компонентів. Мікросервіс робить на рівні процесу те, що

багато фреймворків (наприклад, Spring) завжди робили на рівні компонентів: вільно зв'язані процеси замість нещільно зв'язаних компонентів.

Багато технологій та протоколів, що створюються для реалізації бізнес-процесів [58] та їх підтримки, включені до змісту архітектури мікросервісів, що широко доступна для використання. Ці технології разом створюють потужні інструменти для:

– реалізації бізнес-процесів;

– відкриття доступу для клієнта до різних типів послуг через мережу;

– пошуку послуг (універсальний опис UDDI, відкриття та Інтеграція);

– використання послуг через клієнта;

– визначення бізнес-процесів за допомогою мов визначення потоку задач і створення складних послуг.

Вищезазначені можливості технологій мікросервісів та, зокрема, організація послуг у більших процесах, є фоном застосування мікропослуг до побудови системи медичних інформаційних послуг. Однак на ринку таких рішень зараз не достатньо.

Композиція служб в архітектурі мікросервісів призводить до поєднання окремих служб (Web Service) у структурі, названій процесом, що описує алгоритм реалізації серії служб. Для цього потрібно володіти детальною інформацією про рух процесу (до того, як це буде визначено).

У разі надання екстреної медичної допомоги детальне визначення завдань неможливе. Причиною є непередбачуваність руху інциденту, а також фактори, що впливають на потік можливої схеми лікування:

– кількість та стан здоров’я пацієнтів;

– різноманітність послуг, що належать багатьом оздоровчим установам;

– динаміка стану пацієнта може змінюватися у процесі здійснення лікування;

– доступність (насамперед відстань) до закладів охорони здоров’я для пацієнта.

Можна розділити ці фактори на дві групи:

1) Інформація про стан пацієнта, відома раніше. Це повинно бути відомо з точки зору охорони здоров’я. Наприклад, дані паспорта пацієнта, оцінка стану здоров’я, попередні історії хвороб тощо.

2) Зміна певних факторів. Це може бути миттєва зміна добре відомого або невідомого раніше фактора. Наприклад: перелом, кровотеча, надзвичайна ситуація, тощо.

Все це призводить до того, що детальне налаштування за замовчуванням, як повинна виглядати процедура екстреної медицини, неможливе. Більше того, налаштування зазвичай неможливі, якщо потрібно надавати екстрену допомогу в ситуації, коли ця ситуація вже складеться. Причиною цього є типовий дефіцит достатньої кількості даних під час здійснення дії, оскільки на початку, як правило, ми не маємо можливості надати відповідно точні дані [148].

Потенційні задачі, що стосуються екстреної медицини та ідея побудови системи, яка спирається на парадигму мікросервісів, формує окремі послуги охорони здоров’я в один скоординований процес охорони здоров’я. Запропоноване рішення означає динамічне визначення процесу без детального визначення з точки зору всієї дії. Засновуючи поведінку на типових діаграмах, система виконує певні вступні кроки. Під час їх виконання отримуються додаткові дані на основі системи визначення подальших дій.

2.2.2 Концептуальне представлення інформаційної моделі задачі прийняття рішень для невідкладної медичної допомоги при виникненні серцевої недостатності

Наприклад, на певній вулиці в місті виникла серцева недостатність у людини. У результаті раптового виникнення проблем із серцем людана потребує екстреної допомоги.

З точки зору рятувальних служб задачу важко однозначно вирішити, оскільки на даному етапі інформація про випадок, як правило, неточна: невідомо, які точно у людини проблеми, можливо є гостра необхідність у лікарях певного профілю, до якої саме лікарні необхідно доставити хворого.

Потрібні швидкі та скоординовані рятувальні дії. Людський фактор за таких обставин може призвести до помилок, які можуть вплинути на порятунок життя постраждалого. У цій ситуації найкраще розпорядитися певною системою, здатною керувати рятувальними діями, ліквідовуючи помилки, пов’язані з емоціями, працюючи в стресі та необхідності швидкого реагування.

Можливість підтримки рятувальних служб тут зручна через комп’ютерну систему, яка має доступ до всієї загальновідомої інформації про випадок, яка здатна ефективно планувати дії на основі всіх відомих даних, здатна динамічно коригувати дії до змінних умов, а також для координації дій елементів різних служб.

В ідеальному випадку система, яка вирішує вищезазначені задачі, що виникають у службі порятунку, повинна мати можливість:

– надати незалежність різним елементам, які є його складовими;

– робити можливу пропозицію власних послуг у системі;

– зробити можливий склад служб, що надають компоненти, на більш складні, такі як складні послуги для постраждалої людини.

Основним принципом є відкриття для доступу дій медичних послуг як мережевих послуг вебсервісу при застосуванні парадигми мікросервісів. Своє завдання система реалізує шляхом побудови скелета процесу на початковій

стадії. Тоді, оскільки потік додаткових даних буде поступово зростати, це дозволяє відповідно адаптувати дії до запитів для конкретного випадку.

З метою розвитку системи потрібно визначити базовий повний набір служб, які зможуть служити для створення довільних рятувальних дій. Здається, що визначити повний набір послуг в екстреній медицині чи при серцевій недостатності неможливо, враховуючи той факт, що разом із розвитком медицини з'являться нові послуги, які слід враховувати під час складання. Аргументація вимагає визначення певного стандарту опису послуг, що дозволяє натомість розробляти нові спроби визначення повної бази, необхідної для роботи системи. Приклади послуг, які будуть враховані при розробці рятувальних заходів, є:

– прибуття швидкої допомоги до місця події (ArrivalToPlaceOfIncident) є простою послугою, що означає прибуття машин швидкої допомоги до даного місця події. Це включає прибуття до місця та здійснення оцінки стану потерпілого;

– надання первинної медичної допомоги; підготовка звіту, включаючи дані постраждалої особи у деталях. Госпіталізація постраждалого (HospitalizationOfPatient) – це послуга, основним завданням якої є прийняття пацієнта до лікарні та надання відповідної йому допомоги;

– транспортування пацієнта (TransportationOfPatient) передбачає послугу транспортування постраждалого пацієнта до лікарні.

2.2.3 Реалізація рішення задачі невідкладної медичної допомоги на основі архітектури мікросервісів

Приклад вирішення задачі медичної допомоги при серцевій недостатності з використанням базового набору послуг, згаданих вище (Рис. 2.4).

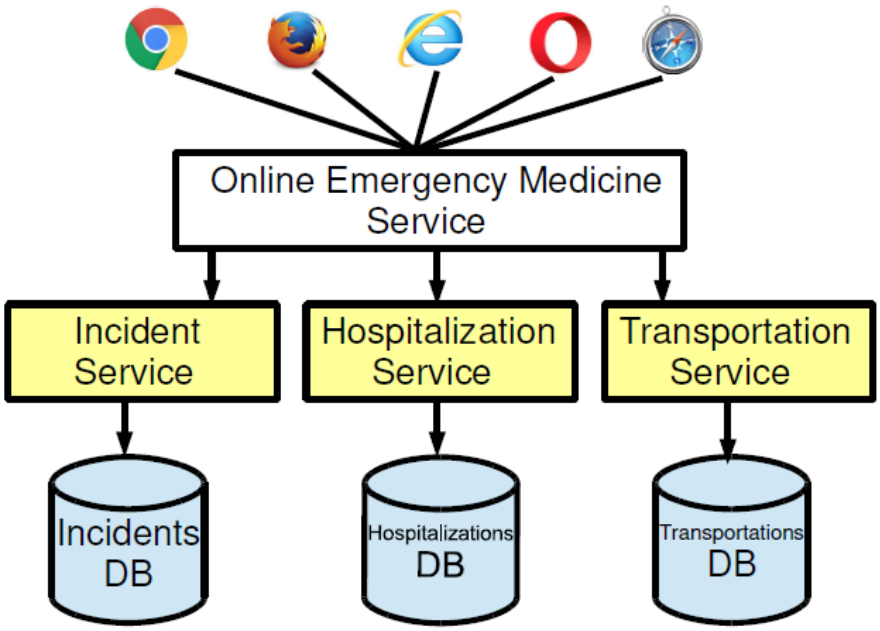


Рисунок 2.4 – Вирішення задачі невідкладної медицини при серцевій недостатності за допомогою базового набору мікропослуг

Розглянемо приклад: у людини стався випадок серцевої недостатності і випадковий свідок телефонує на лінію. Оператор лінії вводить системні дані, що стосуються справи, після чого система негайно починає обробку справи.

Ми уявляємо онлайнслужбу екстреної медицини з окремими мікрослужбами для приїзду до місця пригоди, госпіталізації пацієнта та транспортування пацієнта: неминуче існує ряд змінних частин, які нам доводиться налаштовувати та конфігурувати для побудови такої системи. Як змусити їх працювати разом. Необхідно знати Spring Boot, оскільки Spring Cloud використовує його, потрібні декілька проектів Netflix або інших OSS, і, звичайно конфігурація Spring [47].

Розробники Netflix мали цю проблему під час створення своїх систем і створили сервер реєстрації під назвою Eureka («Я знайшов це» грецькою мовою). Вони зробили свій сервер відкриття з відкритим кодом, і Spring ввійшов у Spring Cloud, що полегшує роботу сервера Eureka. Ось повна програма-сервер виявлення:

Лістинг 2.1 – Програма-сервер виявлення Eureka

```
@SpringBootApplication
@EnableEurekaServer
public class ServiceRegistrationServer {
public static void main(String[] args) {
// Tell Boot to look for registration-server.yml
System.setProperty("spring.config.name", "registrationserver");
SpringApplication.run(ServiceRegistrationServer.class,
args);
}
}
```

Spring Cloud побудований на Spring Boot і використовує. Важливими частинами pom файлу є:

Лістинг 2.2 – Частини pom файлу Spring Cloud

```
<parent>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-parent</artifactId>
<version>_Brixton_.RELEASE</version> <!-- Name of
release train -->
</parent>
<dependencies>
<dependency>
<!-- Setup Spring Boot -->
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter</artifactId>
</dependency>
<dependency>
<!-- Setup Spring MVC & REST, use Embedded Tomcat
-->
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<!-- Spring Cloud starter -->
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter</artifactId>
</dependency>
<dependency>
<!-- Eureka for service registration -->
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-eurekaserver</
artifactId>
</dependency>
</dependencies>
```

За замовчуванням програми Spring Boot шукають файл *application.properties* або *application.yml* для конфігурації. Встановивши властивість *spring.config.name*, можна сказати Spring Boot шукати інший файл – корисно, якщо є кілька програм Spring Boot в одному проекті.

Ця програма шукає *registration-server.properties* або *registration-server.yml*. Ось відповідна конфігурація від *registration-server.yml*:

Лістинг 2.3 – Конфігурація від *registration-server.yml*

```
# Configure this Discovery Server
eureka:
instance:
hostname: localhost
client: # Not a client, don't register with yourself
registerWithEureka: false
fetchRegistry: false
server:
	port: 1111 # HTTP (Tomcat) port
```

За замовчуванням Eureka працює на порту 8761, але тут використовується натомість порт 1111. Крім того, включаючи реєстраційний код у свій процес, можна бути сервером або клієнтом. Конфігурація визначає, що я не є клієнтом, і зупиняє процес сервера, намагаючись зареєструватися.

Для запуску RegistrationServer потрібно відкрити інформаційну панель Eureka http: // localhost: 1111, і розділ із програмами буде порожнім. Потрібно посилатися на сервер виявлення, оскільки це може бути Eureka чи Consul. Створення мікросервісу: служба прибуття до місця події. Мікросервіс – це самостійний процес, який обробляє чітко визначені вимоги.

Налаштовуючи програми за допомогою Spring, наголошуємо на «вільному зчепленні» та «щільній згуртованості». Це не нові поняття (Ларрі Константин приписується першим їх визначенням наприкінці 1960-х [52]), але зараз застосовуємо їх, а не взаємодіючі компоненти (Spring Beans), але до взаємодіючих процесів.

У цьому прикладі є простий мікросервіс *Arrival-to-Place-of-Incident* (Incident), який використовує Spring Data для реалізації JPA IncidentRepository та Spring REST для надання RESTful інтерфейсу до інформації про інцидент (Рис. 2.5). У більшості випадків це простий додаток Spring Boot.

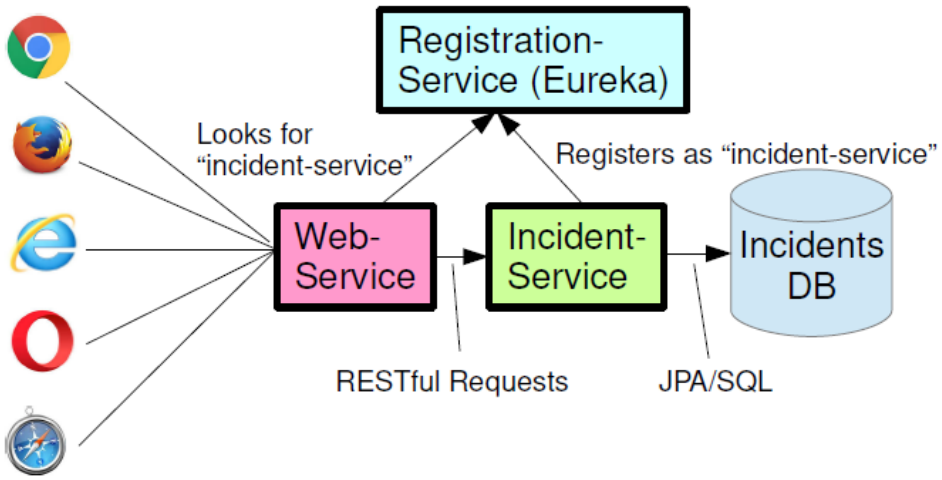


Рисунок 2.5 – Мікросервіс інцидентів, який використовує Spring Data для реалізації JPA IncidentRepository та Spring REST для надання інтерфейсу RESTful до інформації про інцидент

Особливістю є те, що він реєструється на сервері виявлення під час запуску. Ось клас запуску Spring Boot:

Лістинг 2.4 – Клас запуску Spring Boot

```
@EnableAutoConfiguration
@EnableDiscoveryClient
@Import(IncidentsWebApplication.class)
public class IncidentsServer {
@Autowired
IncidentRepository incidentRepository;
public static void main(String[] args) {
// Will configure using incidents-server.yml
System.setProperty("spring.config.name", "incidentsserver");
SpringApplication.run(IncidentsServer.class, args);
}
}
```

Анотації виконують:

- @EnableAutoConfiguration – визначає це як додаток Spring Boot.
- @EnableDiscoveryClient – це дозволяє реєструвати та виявляти послуги. У цьому випадку цей процес реєструється у службі Discovery-Server за допомогою його програмного ім'я.
- @Import (AccountsWebApplication.class) – цей клас конфігурації Java встановлює все інше, що робить цю мікросервіс реєстрацією на сервері виявлення.

– @EnableDiscoveryClient та його конфігурації YML завершують налаштування:

Лістинг 2.5 – @EnableDiscoveryClient та його конфігурації YML

```
# Spring properties
spring:
application:
name: accounts-service
# Discovery Server Access
eureka:
client:
serviceUrl:
defaultZone: http://localhost:1111/eureka/
# HTTP Server
server:
  port: 2222 # HTTP (Tomcat) port
```

Важливо, що файл встановлює назву програми як службу інцидентів. Ця послуга реєструється під цією назвою, і до неї також можна отримати доступ. Визначає власний порт для прослуховування (2222). Усі наші процеси використовують Tomcat, вони не можуть слухати порт 8080. URL-адреса процесу служби Eureka.

Запустивши програму IncidentsService до завершенняи ініціалізації, оновивши інформаційну панель http://localhost:1111, можна побачити послугу інцидентів. Реєстрація триває до 30 секунд (за замовчуванням), тому потрібно перевірити вихідні дані журналу з RegistrationService.

Для більш детальної інформації потрібно перейти: http://localhost:1111 / eureka/apps/, побачивши щось на зразок цього:

Лістинг 2.6 – Інформація за посиланням http://localhost:1111/ eureka/apps/

```
<applications>
<versions__delta>1</versions__delta>
<apps__hashcode>UP_1_</apps__hashcode>
<application>
<name>INCIDENTS-SERVICE</name>
<instance>
<hostName>autgchapmp1m1.corp.emc.com</hostName>
<app>INCIDENTS-SERVICE</app>
<ipAddr>172.16.84.1</ipAddr><status>UP</status>
<overriddenstatus>UNKNOWN</overriddenstatus>
<port enabled="true">3344</port>
<securePort enabled="false">443</securePort>
...
```

```
</instance>
</application>
      </applications>
```

Або перейшовши за http://localhost:1111/eureka/apps/INCIDENTS-SERVICE і переглянувши лише деталі для IncidentsService, якщо він не зареєстрований, буде отримано 404.

Висновки до другого розділу

1. Здійснено математичне представлення задачі індукції дерева рішень та наведено алгоритм для генерації дерева рішень, це дало змогу реалізувати прототип програми до розв’язування задач МН на віддалених платформах на основі даних електронних медичних записів з підтримкою технології Big Data для приросту даних та впровадити його на прикладі діагностування серцевої недостатності.

2. Наведено приклад реалізації індукції дерева рішень, що є прототипом для реалізації на основі хмарної платформи для системних медичних досліджень. Метод продемонстровано на прикладі діагностування серцевої недостатності на основі даних 63-х пацієнтів навчально-практичного центру первинної медико-санітарної допомоги Тернопільського національного медичного університету імені І. Я. Горбачевського.

3. Розглянуто питання архітектури систем прийняття рішень на основі системних медичних досліджень. Інформаційні системи для медичної практики та досліджень часто засновані на так званій монолітній архітектурі. У свою чергу, мікросервіс – це парадигма, яка служить для організації та використання розподілених послуг. На основі архітектури мікросервісів запропоновано рішення для екстреної медицини та ідею побудови системи, яка спирається на парадигму мікросервісів. Запропоноване рішення означає динамічне визначення процесу без детального визначення з точки зору всієї дії. Засновуючи поведінку на типових діаграмах, система виконує певні вступні кроки. Під час їх виконання отримуються додаткові дані на основі системи визначення подальших дій.

4. Здійснено концептуальне представлення інформаційної моделі задачі прийняття рішень для невідкладної медичної допомоги при настанні серцевої недостатності та представлено підхід до реалізації рішення задачі невідкладної медичної допомоги на основі архітектури мікросервісів, підхід до реалізації заснований на використанні фреймворків Spring, Spring Boot та Spring Cloud frameworks. Це дозволило отримати концептуальне представлення інформаційної моделі для прийняття рішень у системних медичних дослідженнях на основі архітектури мікросервісів, що може бути використаним в проблемах профілактики, діагностики та лікування, та впровадженим в шпитальні та лабораторні інформаційні системи на хмарних платформах. Це дає змогу в запропонованому підході підтримувати технологію Big Data.

РОЗДІЛ 3. ЗАДАЧІ КЛАСИЧНОГО МАШИННОГО НАВЧАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ ІНТЕГРОВАНІ З ДАНИМИ ІЗ СИСТЕМ З ВІДКРИТИМ КОДОМ ЕЛЕКТРОННИХ МЕДИЧНИХ ЗАПИСІВ

У розділі здійснено постановку задачі МН, інтегрованих з даними з систем з відкрити кодом електронних медичних записів: наведено математичний опис задачі МН в системних медичних дослідженнях в умовах визначеності; розроблено модель МН для системних медичних досліджень. У результаті експериментальних досліджень отримано моделі МН на основі п’яти методів. Досліджено, що найточнішою моделю відповідно до оцінки якості прогнозування є метод випадкового лісу.

Розглянуто задачу класичного МН в умовах епістемічної та алеаторної невизначеності: здійснено математичний опис задачі МН в умовах невизначеності, розглянуто питання зменшення розмірності в умовах невизначеності, пропонується модифікація методу МГК щодо стратегії передискретизації, представлено загальну блок-схему побудови моделі МН на основі мінімаксного критерію в умовах епістемічної та алеаторної невизначеності, реалізовано на основі розробленої схеми МН приклад для кардіологічних даних. Експериментальне дослідження включає дані 1651 пацієнта з діагнозом інфаркт міокарда.

Основні результати цього розділу опубліковано в роботах [16], [120].

3.1 Задача класичного машинного навчання інтегрованого з даними із систем з відкритим кодом електронних медичних записів в умовах визначеності

3.1.1 Постановка задачі

Сьогодні МН в системних медичних дослідженнях є одним із знарядь для аналізу експериментальних даних у клінічних дослідженнях, так само як і мова, за допомогою якої отримані результати можуть бути представлені, це є

не лише задачею МН в медицині. Математичний апарат МН широко використовується з метою діагностики розв'язку задач класифікації і пошуку нових представлень для прогнозування, формулювання і перевірки нових наукових гіпотез. Використання алгоритмів МН припускає знання основних методів та етапів аналізу даних: їх послідовність, необхідність і достатність.

Рішення для охорони здоров'я, які ґрунтуються на програмному забезпеченні з відкритим кодом активно розвивалися в останні десятиліття разом з комерційними засобами [35], [156], [95]. Найпоширеніші системи з відкритим кодом електронних медичних записів це WorldVistA, OpenEMR та OpenMRS. Перспективи програмного забезпечення електронних медичних записів з відкритим кодом та безкоштовних систем для країн, що розвиваються та країн з фінансовими труднощами розглядалися в роботах Ф. Амінкура, Ф. Фріцса, К. Рейнольдса та інших [35], [156], [66]. Підходи для реалізації систем з відкритим кодом електронних медичних записів, особливо OpenEMR, OpenMRS та OpenDental, в системі охорони здоров'я України також досліджувалися, як і методи інтеграції цих систем з іншим програмним забезпеченням медичного спрямування, розроблялися авторами протягом останніх років [163], [162], [123], [109].

3.1.2 Математичний опис задачі машинного навчання в системних медичних дослідженнях в умовах визначеності

Математично задачі МН в системних медичних дослідженнях ґрунтуються на таких даних: є множина D, яка містить N кортежів. Залежно від задачі певні множини кортежів можуть використовуватися для тренування, тестування і прогнозування. Кожен i-тий кортеж $(a_1^i, a_2^i, ..., a_p^i, c^i)^T$ складається з вхідних даних $(a_1^i, a_2^i, ..., a_p^i)^T$ (мають назву атрибути) і вихідних даних c^i, які є атрибутом класу. Нехай вектор рядок $a_j = (a_j^1, a_j^2, ..., a_j^N)$ представляє значення його j-го атрибуту всіх N кортежів. Атрибути $a_1, ..., a_p$

можуть приймати як числові так і категоріальні дані. Атрибут класу C приймає один з K дискретних значень $c \in \{1,...K\}$.

Метою є прогнозування, використовуючи деякий предиктор, значення атрибуту класу C на основі значень атрибутів $a_1,...,a_p$. Це повинно максимізувати точність прогнозування атрибуту класу, а саме ймовірність $P\{c = c^*\}$ для атрибуту $c^* \in \{1,..., K\}$.

Першою задачею, яка розв'язується в реальних системних медичних дослідженнях, є зменшення розмірності $p \in N$. З цією метою пропонується така модифікація методу аналізу головних компонент (PCA). Ця модифікація включає такі кроки:

Вхідні дані: $A = \{(a_1^i, a_2^i,..., a_p^i, c^i)^T\}_{i=1}^N$

Вихідні дані: головні компоненти, разом з атрибутами.

1. Перетворити всі категоріальні атрибути, кодуючи їх як набір булевих входів, кожен з яких представляється категорією 0 або 1. Можна згенерувати стовпці з категоріальними прапорцями автоматично. У результаті отримуємо числову матрицю $X = \{(a_1^i, x_2^i,..., x_{p_1}^i)^T\}_{i=1}^N \in R^{p_1+1\times N}$.
2. Обчислити середнє значення для рядків $\bar{x}_i = \frac{1}{N}\sum_{j=1}^N x_i^j, i = \overline{1, p_1}$.
3. Обчислити варіації $Var(x_i)$, $i = \overline{1, p_1}$. Нехай варіація – це є повна варіація $Var(X) = \sum_{i=1}^{p_1} Var(x_i)$ (сума вибіркових варіацій).
4. Обчислити матрицю відхилень $X' = \{x_j^i - \overline{x_i}\}_{i=\overline{1,p_1}, j=\overline{1,N}} \in R^{p_1\times N}$.
5. Обчислити матрицю коваріації $C = \frac{1}{p_1} X'(X')^T \in R^{p_1}$.
6. Обчислити власні значення матриці $C: \lambda_1 \le \lambda_2 \le ... \le \lambda_{p_1}$
7. Обчислити власні вектори C. Розглянемо власні вектори w_{p_1} та $w_{p_1-1} \in R^{p_1}$, які відповідають λ_{p_1} і λ_{p_1-1} відповідно. Візьмемо дві перші компоненти, а саме $PC1 = X^T w_{p_1}$ та $PC2 = X^T w_{p_1-1}$. Обчислюємо варіації $Var(PC1)$ та $Var(PC2)$. Звідси маємо відсоток поясненої варіації, який

відповідає першим двом компонентам, а саме $ExplainedVar(PC1) := \frac{Var(PC1)}{Var(X)}$ і $ExplainedVar(PC2) := \frac{Var(PC2)}{Var(X)}$ відповідно.

8. Впорядковуємо значення власних векторів w_{p_1} і w_{p_1-1} в спадаючому порядку їх абсолютних значень. З цією метою застосовуємо перестановки $\pi(w_{p_1})$ і $\pi(w_{p_1-1})^5$. Тоді повертаємо назви перших $ExplainedVar(PC1)*100\%$ атрибутів в перестановці $\pi(w_{p_1})$ і перших $ExplainedVar(PC2)*100\%$ атрибутів в перестановці $\pi(w_{p_1-1})$.

У результаті зменшення розмірності отримуємо деяку числову матрицю $X^{red} = \{(x_1^i, x_2^i, ..., x_{p_2}^i, c^i)^T\}_{i=1}^N \in R^{p_2+1\times N}$. Тоді ці дані можуть бути використані в якості навчальних для ряду задач МН.

3.1.3 Розробка моделі машинного навчання

Підхід ґрунтується на розробці моделі МН, тут маємо на увазі загальну блок-схему, що дозволяє використати розв’язувач МН з можливостями точних, чутливих, стійких результатів.

Відповідно до схеми, представленої на рис. 3.1, починаємо з імпортування і підготовки (feature ingeneering, заповнення пропусків, нормалізації), які збираються в EMR-системах. Способи, з допомогою яких є можливість імпортувати набори даних з EMR систем. Тут зауважуємо, що вибір систем електронних медичних записів з відкритим кодом порівняно з комерційними системами є критичним, оскільки це надає можливості мати відкритий доступ до клінічних даних, які обробляються і обираються на наступних кроках МН.

У реальних застосуваннях, як це показано в результатах дослідження, маємо справу з багатьма атрибутами і лише кілька з них можуть бути суттєвими для задач МН. Отже, природньо постаратися зменшити розмірність атрибутів, щоб залишилися з найбільшими варіаціями.

Тоді повинні означити саму задачу в термінах МН. Це може бути регресія, класифікація, рейтинг і інше. Далі вибираємо відповідний метод (ленер) розв'язку. Найважливішим є вибір параметрів для методів. Це впливає на точність моделі.

Останні кроки (починаються з вибору методу) можуть повторюватися для того, щоб отримати найбільш ефективну модель. Можливості сучасних засобів програмування дозволяють навіть порівняти різні методи, використовуючи відповідні бенчмарки, які розробляються відповідно до певних задач.

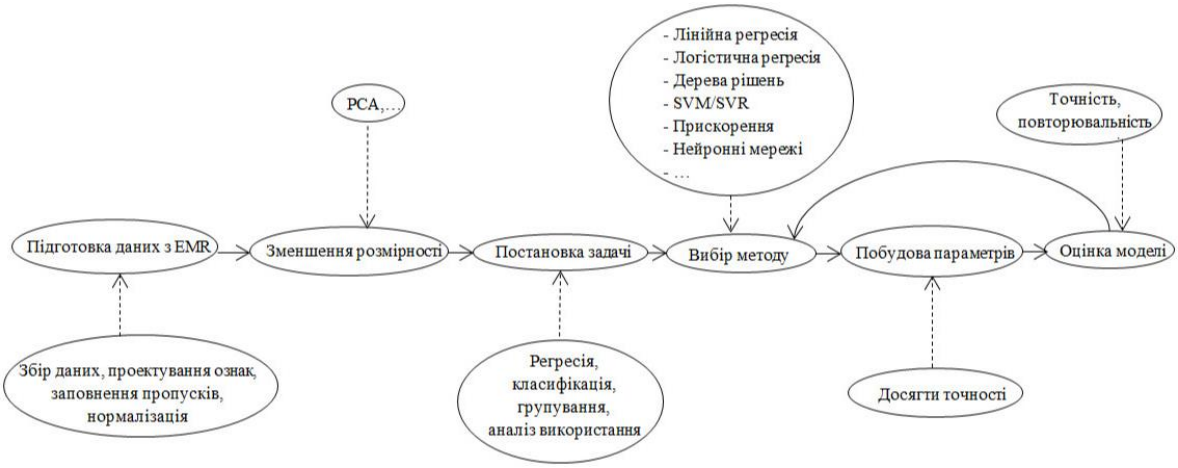


Рисунок 3.1 – Розробка моделі машинного навчання для системних медичних досліджень

Модель МН, яка представлена вище, узгоджується повністю з пакетом **mlr,** який продемонстрований у результатах дослідження.

3.1.4 Задача імпорту даних з систем з відкритим кодом електронних медичних записів

Задача імпорту тренувальних, тестувальних і прогнозувальних наборів даних з систем EMR в середовищі МН не є такою тривіальною через ряд причин: з точки зору програмування, етичних та правових. Є кілька основних підходів для доступу до даних медичних записів пацієнтів у типових системах EMR:

– з допомогою бібліотек API віддаленого доступу для систем EMR (REST, XMLRPC, SOAP);

– використовуючи стандартизовані формати даних HL7;

– використовуючи загальні стандартні формати даних (XML/CSV).

Тут представляємо деякі особливості застосування зазначених вище підходів для двох найбільш поширених систем з відкритим кодом EMR: OpenEMR, OpenMRS.

Розглядаючи ці рідні бібліотеки віддаленого доступу API для комерційних систем EMR, зазначаємо відсутність (або принаймі відкрита інформація не є доступна) технологічних засобів (особливо бібліотек віддаленого доступу API). Навіть безкоштовні системи EMR з відкритим кодом мають мало можливостей віддаленого доступу за допомогою бібліотек API.

Наприклад, OpenEMR пропонує єдину рідну бібліотеку API з віддаленим доступом. Незважаючи на те, що вона була розроблена в 2013 році Master Mobility App, все ще є корисною і повністю функціональною.

OpenMRS реалізує віддалений доступ до внутрішніх API через протокол REST, використовуючи модуль REST. Специфікація OpenMRS API є доступною.

HL7 надає фреймворк (та певні пов’язані з ним стандарти) для обміну, інтеграції, розповсюдження і отримання електронної медичної інформації. Наступні версії HL7 є надалі активними: HL7 Version 3 (V3) Normative Edition (ґрунтується на великому наборі XML форм); HL7 Version 3 Clinical Document Architecture (CDA); HL7 Fast Healthcare Interoperability Resources Specification (FHIR, також на основі XML); HL7 Version 2 Product Suite (версії 2.x (V2) HL7 використовує підключення запису обміну повідомленнями на основі текстових файлів). Звертаємо увагу, що сучасні HL7 CDA документи використовують Continuity of Care Document (CCD) термін дуже часто.

Незважаючи на тривалий період розробки і застосувань, навіть провідні і широко розповсюдженні EMR системи (як комерційні так і безкоштовні з

відкритим кодом) мають обмежену або часткову підтримку можливостей HL7. Наприклад, OpenEMR має лише кілька вбудованих можливостей пов'язаних з HL7.

– Офіційно підтримуваний імпорт медичних записів пацієнтів, але лише у вигляді повідомлень HL7 Version 2.

– Експорт даних пацієнтів CCD формат даних HL7 сумісний з CDA є наявний, але тільки для окремого пацієнта.

– У 2017 році реалізація інтеграція фреймворку FHIR розпочалася, але вона все ще повністю незавершена. Також є обіцяючим процес інтеграції, який ґрунтується на рефакторингу HAPI FHIR бібліотеки, використовуючи мову програмування PHP.

OpenMRS надає більше можливостей, включаючи двонапрямлену обробку даних HL7, використовуючи модулі користувача:

– Socketh7Listener Module приймає і обробляє імпорт повідомлень HL7(лише версії 2.x);

– CCD Module забезпечує експорт резюме медичних записів пацієнтів у HL7 третьої версії суміжному форматі CCD;

– HL7Query Module24 також підтримує експорт даних про пацієнтів у вигляді повідомлень HL7 версії 2.x (лише поточна версія ORUR01) за допомогою спеціального модуля.

Незважаючи на зазначені вище задачі, які розглядають рівні реалізації, існує достатньо аргументів для адаптації використання форматів даних сумісних з HL7 і розробки інформаційних систем, які є орієнтовані на МН. Наявність та активна розробка великої кількості безкоштовних із відкритим кодом парсерів HL7 для різних мов програмування є однією з таких причин.

Експорт даних в загальному форматі реалізовано в системах EMR такими способами:

– Експорт окремих документів (медичних форм) для вибраних пацієнтів. Такі риси підтримуються більшістю систем EMR. Загальними форматами експорту є PDF, XML, XLS, CSV.

– Експорт повних медичних записів пацієнтів. Типові реалізації включають використання даних, що ґрунтуються на XML, відповідно до специфікації Continuty of Care Record (CCR). Сам CCR є впровадженням іншого стандарту сумісності – ASTM International.

Важливо звернути увагу, що експорт командних файлів медичних записів пацієнтів з систем EMR рідко підтримується, особливо у випадку комерційних систем. Розглядаючи системи EMR з відкритим кодом, система OpenEMR має кілька вмонтованих особливостей, що експортують дані у форматах користувача:

– доступний експорт запису даних про пацієнта в сумісний з ASTM формат даних CCR на основі XML;

– окремі документи форми та записи даних лабораторії можна експортувати у форматі PDF (рідко XML або CSV);

– кожен окремо вибраний запис пацієнта щодо демографічних даних може бути експортований у набір даних XML;

– кілька (вибраних) записів даних про демографічні показники пацієнтів можна експортувати у файл CSV за допомогою Batch Communication Tool (Рис. 3.2).

–

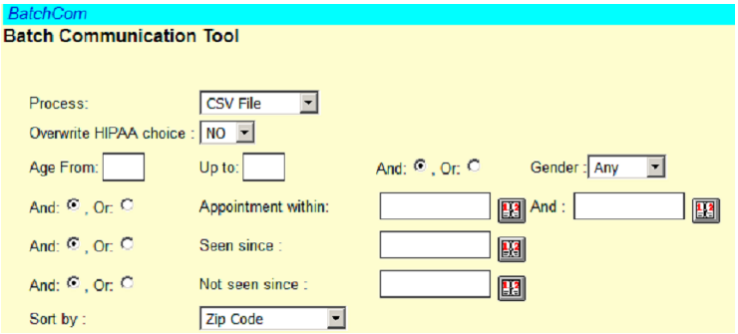


Рисунок 3.2 – Пакетний експорт декількох записів у CSV у OpenEMR

Система OpenMRS надає більше передових і гнучких засобів, але у вигляді клієнтських модулів:

– i2b2export Module експортує дані OpenMRS, включаючи пацієнтів, провайдерів, концепції, джерела концепцій, зустрічі, спостереження та замовлення на наркотики, у користувацьку схему XML;

– Cohort Builder Module29 надає можливості експорту для записів пацієнтів, які включаються до когорти в різні формати (підтримуються XML та CSV).

Отже, встановлення сумісності та обміну даними між різними системами EMR є важливою задачею [32]. Це призводить до розробки не тільки згаданих вище міжнародних стандартів, але і спеціальних платформ і фреймворків. Серед десятків рішень звертаємо нашу увагу на OpenEHR, яка є відкритою доменною платформою для розробки гнучких систем електронного здоров'я. Інший спосіб сумісності запропонував NextGen Healthcare за допомогою інструменту з відкритим кодом NextGen Connect (раніше Mirth Connect). З наведених причин робимо висновок, що ефективним способом навчання, тестування та прогнозування наборів даних із системи EMR є експорт-імпорт записів системних медичних досліджень у сумісні з XML формати. Такий файл частково або навіть повністю сумісний з CCR/CCD (CDA) – залежно від рівня прийняття міжнародних стандартів конкретною системою EMR, з якою працює дослідник. У той же час файли даних XML були перетворені та розроблені за допомогою десятків бібліотек (наприклад, у Python або R).

3.1.5 Задачі зменшення розмірності та класифікації

У системних медичних дослідженнях завжди існує велике число залежностей, які досліджуються. Однак багато з них не є статистично значимими і у відсутності клінічної значимості вони не є настільки необхідними для того, щоб їх брати до уваги й інтерпретувати. Наскільки важливим з точки зору об'єктів дослідження ці взаємозв'язки неможливо припустити наперед. Наприклад, лише 5-25% всіх можливих залежностей є статистично і клінічно важливими.

Більше того, більшість індикаторів, можуть ще мати залежності між собою. Використовуючи їх в якості початкової множини показників, можемо будувати на їх основі більш комплексні, інтегруючі показники, використовуючи, наприклад, метод головних компонент (МГК). Де число таких складних показників компонент буде значно меншим, ніж число початкових показників, у результаті цієї процедури отримаємо нові показники, що компактно презентують значно більше інформації. У результаті з'являється можливість відфільтровувати випадкові компоненти і давати більш надійну інформацію про структуру оригінальних показників, які досліджують групи пацієнтів.

Використовують три головні типи класифікації задач у системних медичних дослідженнях:

1. бінарна класифікація (тут можуть бути додатні і негативні значення);

2. багатокласова класифікація (об'єкт несе характеристику одного класу);

3. об'єкт належить до багатьох класів одночасно.

Розглянемо три приклади для таких задач:

1. Визначити чи особа має певне захворювання X. 1 – особа є хвора, 0 – особа є здорова.

2. Ідентифікація певного захворювання серед захворювань A, B, C, D, E.

3. Визначення кількох захворювань серед A, B, C, D, E.

Алгоритми класифікації (подібно до індукції вузлів дерев рішень) автоматично поділяє значення чисельних атрибутів a_j на два інтервали: $a_j \le y_j$, $a_j > y_j$ і категоріальні атрибути a_k діляться на дві підмножини: $a_k \in S_k$, $a_k \notin S_k$.

Поділ числових атрибутів ґрунтується, як правило, на мірах, на основі ентропії або Gini Index. Процес поділу рекурсивно повторюється поки покращується точність прогнозування. Останній крок включає видалення вузлів для уникнення перенавчання моделі. У результаті, повинні отримати

набір правил, що починаються від коренння дерева до кожного термінального вузла, які включають нерівності для числових атрибутів і умови включення для категоріальних атрибутів.

3.1.6 Результати експериментальних досліджень на основі клінічних даних

Експерементальні дослідження пов’язані з прогнозуванням переломів для жінок середнього віку, що є на сьогодні актуальною, профілактичною задачею.

Розглядаємо дані клінічних, лабораторних досліджень 1469 жінок, які попередньо зберігалися в системі OpenEMR. Найважливішою групою показників є результати денситометрії кісток. Сумарно маємо $p = 182$ показники. Після видалення пропусків у наборі даних залишилося 1242 кортежі.

Пацієнти були погруповані в три групи відповідно до наявності переломів. З цією метою використовуємо атрибут класу «Fractures» з такими категоріальними значеннями: 1 – немає переломів, 2 – периферійні переломи, 3 – вертебральні переломи.

Застосовуючи зменшення розмірності за допомогою методу *МГК*, розглядаємо два головних компоненти. А саме, $PC1$ має 43,4% пояснювальної варіації, $PC2$ – 12,5%. Тобто ці два компоненти в змозі пояснити 55,9% варіації даних. Результати *МГК* відображені на рис. 3.3. Відповідно до підходу запропонованого у постановці задачі, пропорційно до відсотків $ExplainedVar(PC1)$ і $ExplainedVar(PC2)$ обираємо 8 атрибутів, які ґрунтуються на $PC1$ і 2 які ґрунтуються на $PC2$. А саме це “Weight”, “TOTAL_Fat”,“RIGHT_TOTAL_Fat_g”,“LEFT_TOTAL_Fat_g”,“RIGHT_TOTAL_Total_Mass_kg”, “LEFT_TOTAL_Total_Mass_kg”, “TOTAL_Tissue g”, “RIGHT_TOTAL_Tissue_g” для $ExplainedVar(PC1)$ і “Weight”, “TOTAL_Fat” для $ExplainedVar(PC2)$. Всі атрибути, що розглядаються, є числовими і після зменшення розмірності розглядаємо $p_1 = 8$ атрибутів зазначених вище. На

рис. 3.4 можемо проаналізувати площину $PC1$-$PC2$ відповідно до еліпсоїдів, які представляють три групи пацієнтів з прогнозованими переломами. Зменшуючи розмірність, спостерігаємо атрибути з найбільшими варіаціями разом з їх напрямками відповідно до груп пацієнтів.

Розробка моделі МН була реалізована в середовищі вільного програмного забезпечення R, за допомогою пакету **mlr**.

Відповідно до рис. 3.1 визначаємо задачу МН. У нашому випадку це є задачею класифікації відповідно до атрибуту класу «Fractures». У термінах пакету **mlr** це може бути описано таким чином:

Лістинг 3.1 – Опис задачі класифікації у пакеті mlr

```
task = makeClassifTask(id = "fractures_classification", data = df1, target =
                                "Fractures")
```

Наступний крок – це вибір методу розв’язування задачі МН. Створюємо «ленера» з цією метою. Пакет **mlr** дозволяє створити бенчмарк ленерів:

Лістинг 3.2 – Створення бенчмарка ленерів у пакеті mlr

```
lrns <- makeLearners(c("lda","rpart", "C50","rFerns", "randomForestSRC"), type =
                                "classif")
```

Тут бенчмарк включає 5 методів, що можуть бути застосовані для задачі класифікації. А саме лінійний дискрімінантний аналіз (lda), рекурсивне розділення та дерева регресії (rpart), C5.0, випадкових папоротей (random ferns) та випадкового лісу (random forest).

Лістинг 3.3 – Визначення задачі МН, методів і стратегії передискретизації

```
comparison <- benchmark(tasks = task, learners = lrns, resampling = cv5)
```

Тут визначаємо задачу МН, ленерів (методів) і стратегію передискретизації.

У результаті отримуємо моделі МН на основі п’яти методів. Результати ефективності методів, за допомогою міри mmce представлені на рис. 3.6. Бачимо, що найточнішою моделю відповідно до оцінки якості прогнозування є метод випадкового лісу (random forest), найгіршим є випадкових папоротей (random ferns).

Дерево рішень, яке індукується на основі атрибутів після зменшення розмірності показано на рис. 3.5. Тут 5 атрибутів були включені, їх використання показано у таблиці

Таблиця 3.1 – Використання атрибутів для дерева рішень на рис. 3.5

Name of attribute	Usage
RIGHT_TOTAL_Tissue _g	100.00%
RIGHT_TOTAL_Fat_g	14.05%
RIGHT_TOTAL_Total_Mass_kg	14.05%
TOTAL_Fat	11.19%
TOTAL_Tissue_g	2.61%

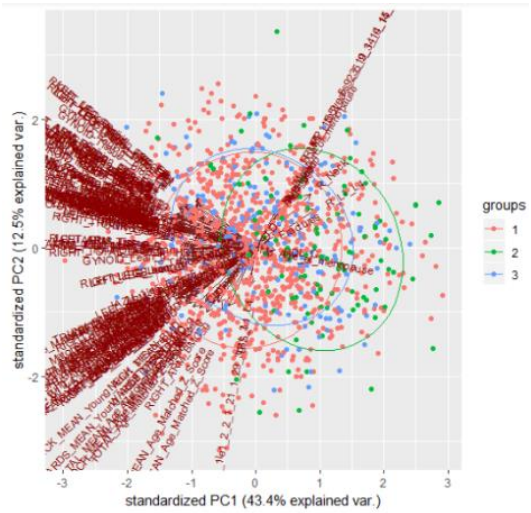


Рисунок 3.3 – Дані в площині $PC1$- $PC2$. Стрілки показують “оберти” атрибутів в площині PC1-PC2

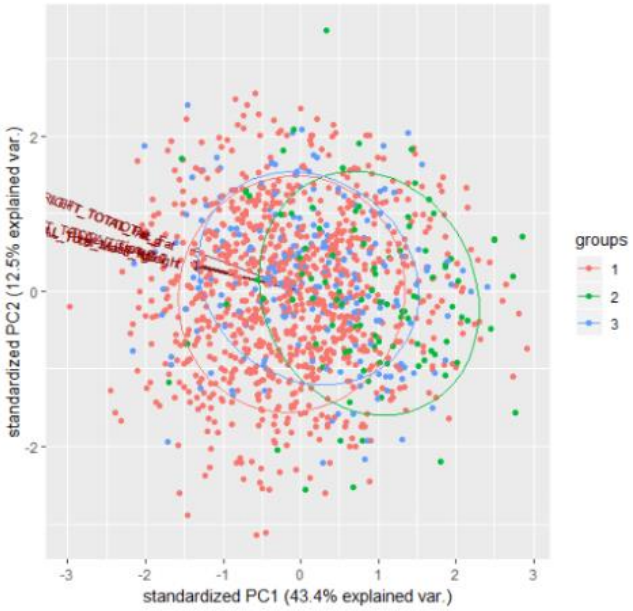


Рисунок 3.4 – Дані в площині $PC1_PC2$. Стрілки показують зменшення розмірності та напрямки змін у групах пацієнтів

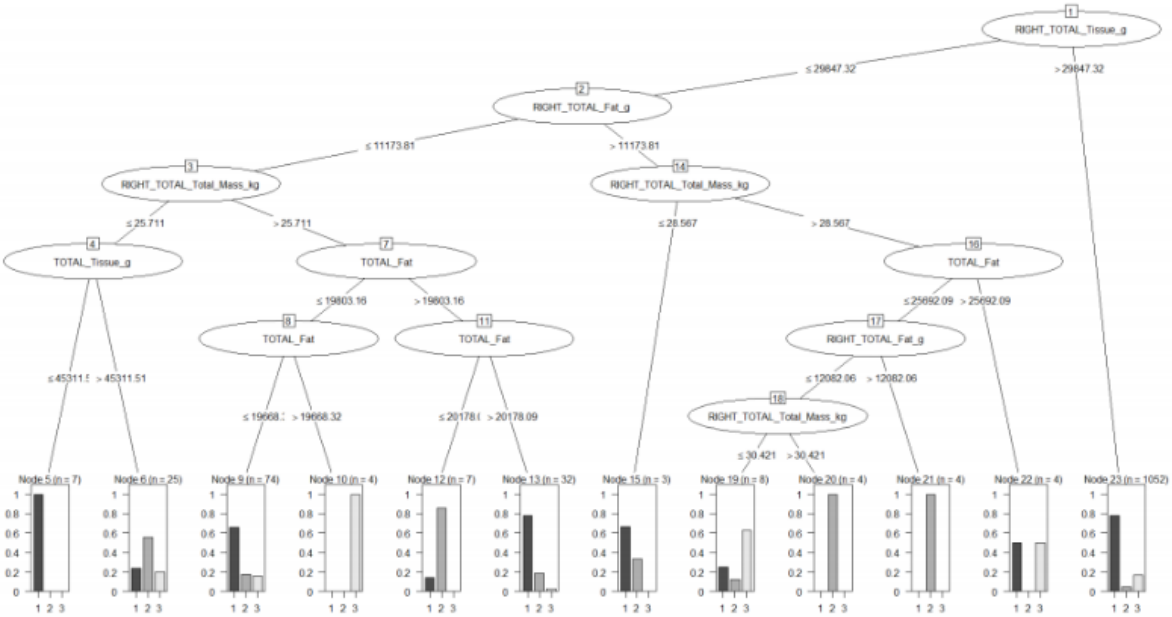


Рисунок 3.5 – Дерево рішень, яке побудоване на основі зменшення розмірності даних

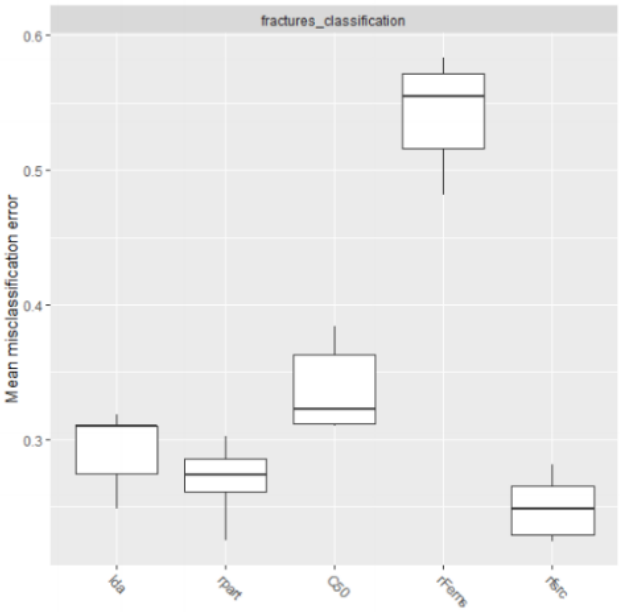


Рисунок 3.6 – Порівняння мір ефективності для різних методів класифікації: Ida, rpart, C50, rFerns, randomForestSRC

3.2 Задача класичного машинного навчання в умовах невизначеності

3.2.1 Математичний опис задачі машинного навчання в умовах епістемічної та алеаторної невизначеності

У роботі [60] задача МН з вчителем, що означає прогнозування Y за допомогою X, функції втрат L та набору розподілів ймовірностей Γ на (X,Y) може бути сформульована як мінімаксна задача щодо L, у випадку, коли максимізація відбувається за всіма можливими розподілами G та мінімізація за правилами прийняття рішень $\psi \in \Psi$.

$$\arg\min_{\psi\in\Psi}\max_{P\in\Gamma} \mathrm{E}[L(\mathrm{Y},\psi(X))] \tag{3.1}$$

Де $E[\bullet]$ – математичне сподівання.

Задача (3.1) була розв’язана шляхом введення узагальнення принципу максимуму ентропії.

Математичний опис задачі МН з вчителем у системних медичних дослідженнях було представлено в [116], [121]. Тут формулюється це у випадку мінімаксу.

Математично задача МН у системних медичних дослідженнях базується на наступних даних. Є набір D, що містить N кортежів

$$D = \left\{ X_i \mid i = \overline{1, N} \right\} \tag{3.2}$$

З метою моделювання алеаторних невизначеностей розглянемо ML з вчителем щодо розподілу навчальних кортежів. У межах класу всіх підмножин D визначимо Γ, розподіли класів навчальних та тестових наборів

$$\Gamma = \left\{ (D_{train,j}, D_{test,j}) \subset D \times D \mid D_{train,j} \cap D_{test,j} = \varnothing, D_{train,j} \cup D_{test,j} = D, j = \overline{1, 2^N}, \right\} \tag{3.3}$$

Де $D_{train,j}$ та $D_{test,j}$ – це всі можливі набори для навчання та тестування відповідно. На практиці стратегії передискретизації – це розподіл класів кортежів, які найкращим чином характеризують алеаторні невизначеності. Вводиться опис стратегії передискретизації $\gamma \subset \Gamma$

$$\gamma = \left\{ (D_{train,k}, D_{test,k}) \subset D \times D \mid D_{train,j} \cap D_{test,j} = \varnothing, \cup_k D_{train,k} = \cup_k D_{test,k} = D \right\} \tag{3.4}$$

Як приклади стратегій передискретизації γ можна розглянути $cv3, cv5, cv10$, що відповідають кратній перехресній валідації для різних *k*.

Кожен i-й кортеж $X_i = (x_1^i, x_2^i, ..., x_p^i, \mathrm{Y}^i)^T$ складається з вхідних даних $(x_1^i, x_2^i, ..., x_p^i)^T$ (їх також називають атрибутами) та вихідних даних Y^i.

Нехай вектор рядка $x_j = (x_j^1, x_j^2, ..., x_j^N)$ представляє значення j-го атрибута всіх N кортежів. Вихідний атрибут $\mathrm{Y} = (\mathrm{Y}^1, \mathrm{Y}^2, ..., \mathrm{Y}^N)$ приймає всі вихідні дані. Атрибути $x_1, ..., x_p$ та Y (залежно від завдання класифікації або регресії) можуть приймати як числові, так і категоріальні значення.

У найпростішому випадку задачею навчання з вчителем є передбачити, використовуючи певний предиктор, значення вихідного атрибута Y^{N+1} на основі значень атрибутів $x_1^{N+1}, ..., x_p^{N+1}$. Предиктор повинен максимізувати точність прогнозування вихідного атрибута, а саме ймовірність $P\{\mathrm{Y}^j \, corresponds \;\; to \;\; x_1^{N+1}, ..., x_p^{N+1}\}$ для довільного $j \in \{1, ..., N\}^1$.

Далі, застосовуючи мінімаксний підхід, вводиться $h \in \Psi$, розглянутий клас моделей МН $h(X,\gamma)$, який можна навчити та налаштувати на дані $X \subset D$ та оцінити з урахуванням деяких стратегій передискретизації γ.

Порівнюючи різні моделі МН, метою є мінімізація очікуваних втрат. Але також потрібно взяти до уваги стратегії передискретизації, за якими також повинні оцінювати функцію втрат. Таке формулювання задачі МН розглядає два види невизначеності. А саме, невизначеність у передискретизації є алеаторичною, оскільки вона пов'язана з даними. У той же час, невизначеність при виборі моделей є епістемічною. Математично мінімальна задача МН описується як пошук моделі h завдяки

$$\arg\min_{h\in\Psi}\max_{\gamma\in\Gamma} \mathrm{E}[L(\mathrm{Y},h(X,\gamma))] \tag{3.5}$$

3.2.2 Зменшення розмірності в умовах невизначеності

Оскільки справжні набори системних медичних досліджень включають десятки життєво важливих показників, морфологічні, біохімічні та клінічні оцінки, природним є бажання зменшити кількість ознак, залишаючи такі, що мають найбільші відмінності.

Метод аналізу головних компонентів (МГК) є одним із широко застосовуваних методів зменшення розмірності. Незважаючи на те, що він застосовується для задач МН без вчителя, він допомагає уточнити результати, коли використовується для МН з вчителем, як-от задача класифікації або регресії. Задача зменшення кількості атрибутів надзвичайно важлива для медичного застосування при інтерпретації результатів. Нижче пропонується спосіб його застосування в умовах алеаторної невизначеності.

Оскільки в задачі мінімаксу МН функція втрат розраховується для всіх стратегій передискретизації, алгоритм зменшення розмірності повинен застосовуватися окремо для кожної стратегії γ. Можемо представити будь-яку стратегію передискретизації γ як $U_l^r =_1 I_{training}^l(\gamma)$ де $I_{training}^l(\gamma), l \in \overline{1,r}$, є l-ою

вибіркою індексів від 1 до N, що відповідає навчальним кортежам для стратегії γ^3

Нехай $D_l(\gamma)$ - вхідні дані, D коли застосовується вибірка індексів $I^l_{training}(\gamma)$. А саме, $D_l(\gamma) = \left\{(x_1^i, x_2^i, ..., x_p^i, \mathrm{Y}^i)^T\right\}_{i\in\{1...N\}\cap I^l_{training}(\gamma)}$, де N_l < N – кількість кортежів у l-тій навчальній вибірці.

Перед тренуванням і налаштуванням моделі МН $h(X,\gamma)$ зменшуємо розмірність $p \in N$ D відносно γ .

Для цього пропонуємо модифікацію методу МГК щодо стратегії передискретизації γ .

Алгоритм МГК щодо стратегії передискретизації γ

Вхідні дані: $D = \left\{(x_1^i, x_2^i, ..., x_p^i, \mathrm{Y}^i)^T\right\}_{i=1}^N, \gamma$

Результат: основні компоненти разом з атрибутами.

1 перетворити дані D в матрицю A з усіма числовими записами;

2 застосувати стратегію передискретизації γ для A: $A_l = \left\{(a_1^{l,i}, a_2^{l,i}, ..., a_p^{l,i}, \mathrm{Y}^{l,i})^T\right\}_{i=1}^{N_l} \in R^{p_1+1\times N_l}, l = \overline{1,r}$;

3 **for each** $D_l(\gamma), l = \overline{1,r}$ **do**

4 $\bar{a}_{l,i} := \frac{1}{\mathrm{N}_1}\sum_{j=1}^{N_l} a_i^{l,j}, i = \overline{1,p_1}$;

5 обчислити $Var(a_{l,i}), i = \overline{1,p_1}$;

6 $Var(A_l) := \sum_{i=1}^{p_1} Var(a_{l,i})$;

7 $A_l^{'} := \{a_j^{l,i} - \bar{a}_{l,i}\}_{i=\overline{1,p_1}, j=\overline{1,N_l}} \in R^{p_1\times N_l}$.;

8 $C_l := \frac{1}{p_1} A_l^{'}(A_l^{'})^T \in R^{p_1}$.,

9 обчислити власні значення $C_l : \lambda_{l,1} \le \lambda_{l,2} \le ... \le \lambda_{l,p_1}$

10 обчислити власні вектори $C_l : w_{l,1}, w_{l,2} ... w_{l,p_1}$, що відповідають $\lambda_{l,1} \le \lambda_{l,2} \le ... \le \lambda_{l,p_1}$ відповідно

11 $PC1_l := A_l^T w_{l,p_1}, PC2_l := A_l^T w_{l,p_1-1}$;

12 обчислити дисперсії $Var(PC1_l)$ та $Var(PC2_l)$;

13 $\mathrm{ExplainedVar(PC1_l)} := \frac{\mathrm{Var(PC1_l)}}{\mathrm{Var(D_l)}}$, $\mathrm{ExplainedVar(PC2_l)} := \frac{\mathrm{Var(PC2_l)}}{\mathrm{Var(D_l)}}$;

14 $\pi(w_{l,p_1})$ та $\pi(w_{l,p_1-1})$

End

16 $\mathrm{ExplainedVar(PC1_l)} := \frac{1}{r}\sum_{l=1}^{r}\mathrm{ExplainedVar(PC1_l)}$,

$\text{ExplainedVar}(\text{PC1}_2) := \frac{1}{r}\sum_{l=1}^{r}\text{ExplainedVar}(\text{PC1}_2)$

17 **return** імена атрибутів з $\pi(w_{l,p_1}), l=\overline{1,r}$ і $\pi(w_{l,p_1-1}), l=\overline{1,r}$

Далі описуємо основні кроки алгоритму. На кроці 1 перетворюємо всі категоріальні атрибути, кодуючи їх як набір булевих входів, кожен з яких представляє одну категорію 0 або 1. Можемо генерувати стовпці з прапорами категорій автоматично.

Далі повторюємо кроки 4-14 для кожного $D_l(\gamma), l=\overline{1,r}$. Отже, вони включають обчислення середніх значень для рядків (Крок 4), дисперсії $Var(a_{l,i}), i=\overline{1,p_1}$ (Крок 5), загальної дисперсії (сума дисперсій вибірки) $Var(A_l)$ (Крок 6), матриця відхилень $A_l^{'} \in R^{p_1 \times N_l}$ (Крок 7), коваріаційна матриця $C_l \in R^{p_1}$ (Крок 8), власні значення матриці C_l у зростаючому порядку (Крок 9), власні вектори. C_l (Крок 10). Тут розглянемо власні вектори w_{l,p_1} та $w_{l,p_1-1} \in R^{p_1}$, що відповідають λ_{l,p_1} і λ_{l,p_1-1}відповідно. На кроці 11 отримуємо дві перші основні складові $PC1_l := A_l^T w_{l,p_1}$ і $PC2_l := A_l^T w_{l,p_1-1}$. Обчислюємо їх дисперсії *Var (PC1$_l$)* та *Var (PC2$_l$)* на кроці 12. Звідси отримуємо відсотки поясненої дисперсії, що відповідає першим двом компонентам *ExplainedVar(PC1$_l$)* і *ExplainedVar(PC2$_l$)* відповідно (Крок 13). Далі впорядковуємо значення власних векторів w_{l,p_1} і w_{l,p_1-1} в порядку зменшення їх абсолютних значень. Для цієї мети застосовуємо перестановки $\pi(w_{l,p_1})$ і $\pi(w_{l,p_1-1})$, використовуємо позначення $\pi(x)$ для перестановки, що упорядковує вектор *x* у порядку зменшення абсолютних значень його елементів

Далі повертаємо імена перших *ExplainedVar (PC1$_l$) 100%* атрибутів у перестановці $\pi(w_{l,p_1})$ та перших *ExplainedVar (PC2$_l$) 100%* атрибутів у перестановці $\pi(w_{l,p_1-1})$ (Крок 14).

Після завершення основного циклу обчислюємо дисперсію основних компонентів для стратегії передискретизації γ (Крок 16). Наприкінці повертаємо назви перших *ExplainedVar (PC1$_l$) 100%* атрибутів, які

найчастіше зустрічаються в перестановках $\pi(w_{l,p_1}), l=\overline{1,r}$ та перших *ExplainedVar (PC2$_l$) 100%* атрибутів, які найчастіше зустрічаються в перестановках $\pi(w_{l,p_1-1}), l=\overline{1,r}$ (Крок 17).

У результаті зменшення розмірності отримуємо деяку числову матрицю $A^{red}=\{(a_1^i;a_2^i,...,a_{p_2}^i,\mathrm{Y}^i)^T\}_{i=1}^N \in R^{p_2+1\times N}$, $p_2 \le p_1$. Тоді ці дані можуть бути використані як навчальні для розв'язання задач МН на основі мінімаксного підходу.

Примітка 1. Етапи 2 та 10-14 – це модифікації традиційного алгоритму МГК. По-перше, на кроці 1 перетворюємо всі категоріальні атрибути, які широко використовуються в системних медичних дослідженях, на булеві дані. По-друге, при розгляді двох основних компонентів, які традиційно використовуються для площинної презентації навчальних наборів, пропонуємо підхід до вибору деякої зменшеної кількості атрибутів для подальшого дослідження (наприклад, розробка моделі МН). Ця кількість пов'язана з кількістю пояснених варіацій. Останнє припущення дозволяє по-справжньому зменшити розмір задач МН у системних медичних дослідженнях в умовах невизначеності.

Примітка 2. Звичайно, повинні взяти до уваги випадок, якщо дисперсія, що пояснюється першими двома компонентами, низька. У таких випадках нам потрібно брати до уваги компоненти *PC3*, *PC4* і так далі для отримання відповідної дисперсії. Етапи алгоритму 10-14 та інші мають бути змінені відповідним чином.

Примітка 3. Варто звернути увагу, що потрібно обчислити МГК залежно від стратегії передискретизації, оскільки МГК застосовується до навчальних кортежів $D_l(\gamma)$ (не для всього набору даних *D*). Отже, на кроці 14 можуть бути обрані різні ознаки залежно від вибірки індексів $I_{training}^l(\gamma)$. У свою чергу, це впливає на вибір атрибутів на останньому кроці алгоритму для всієї стратегії передискретизації.

3.2.3 Загальна блок-схема побудови моделі машинного навчання на основі мінімаксного критерію в умовах епістемічної та алеаторної невизначеності

Загальна блок-схема (Рис. 3.6) дозволяє нам отримати учня (learner) задачі МН на основі мінімаксного підходу з можливістю точних, прийнятних та стабільних результатів. Модель МН, сформульована в умовах визначеності, представлена в [121]. Тут узагальнюємо блок-схему для розв’язання задачі в умовах невизначеності як у моделі, так і в стратегії передискретизації.

Починаємо з імпорту та підготовки даних (генерування ознак, заповнення прогалин, нормалізація), зібраних у EMR системах. Представлено способи імпорту наборів даних із EMR систем. Зазначимо, що вибір EMR систем з відкритим кодом у порівнянні з комерційними є надзвичайно важливим, оскільки він дозволяє отримати відкритий доступ до клінічних даних, які можуть бути опрацьовані та обрані для подальших етапів МН [121].

Рисунок 3.6 – Розробка моделі машинного навчання для системних медичних досліджень

Тоді слід визначити саме завдання з точки зору МН. Це може бути регресія, класифікація, групування тощо.

Також визначаються стратегії передискретизації γ. Наприклад, стратегії передискретизації, що підтримуються пакетом mlr, включають: крос-валідацію (cv), перехресну крос-валідацію (LOO), повторну крос-валідацію (RepCV), колірну субдискретизацію, яка також називається крос-валідацією Монте-Карло (Subsample), Holdout метод (тренування/тестування) (Holdout) [155].

У реальному додатку маємо справу з великою кількістю атрибутів. Лише деякі з них можуть бути важливими для задач МН. Отже, природно намагатися зменшити розмірність, відкидаючи атрибути з найбільшими відхиленнями.

Далі вказуємо набір Ψ відповідних методів (той, хто навчається (learner)) розв'язку. Найважливішим є вибір параметрів для методів, що впливає на точність моделі. У наступному циклі налаштовуємо параметри для кожної моделі з Ψ на основі всіх стратегій передискретизації γ, які використовуються. Вихідна модель задовольнятиме критерію мінімаксного підходу (2).

Можливості пакета mlr дозволяють реалізовувати це, використовуючи відповідні інструменти, розроблені з урахуванням певних завдань [105], [55], [104]. Модель МН, представлена вище, повністю узгоджується з пакетом mlr, який пропонує прототипи випадків задачі МН: завдання, передискретизація, навчання.

З метою аналізу обчислювальної складності запропонованого підходу розглянемо приклад множини Ψ, що включає метод 4-шарової нейронної мережі з кількостями нейронів i, j, k, l на шарах на основі зворотнього поширення помилки та метод C5.0 індукції дерева рішень висотою h.

Припустимо, що навчальна вибірка даних D містить $\#(D)$ наборів на основі i атрибутів.

Тоді обчислювальна складнність зазначеного методу нейронної мережі на основі t ітерацій становить [186]:

$$O_{NN} := O(t\#(D)(ij + jk + kl)) \tag{3.6}$$

Обчислювальна складність індукції дерева рішень [86]:

$$O_{C5.0} := O(h:\#(D)\log\#(D)) \tag{3.7}$$

Обчислювальна складність передискретитизації на основі k–кратної крос-валідації становить [187]:

$$O_{CV} = O(k\#(D)) \tag{3.8}$$

Таким чином обчислювальна складність побудови моделі МН на основі схеми на рис. 3.6 становить

$$O_{CV} = O(O_{NN} + O_{C5.0}) \tag{3.9}$$

Оскільки k є сталою, то з формули (3.9) видно, що обчислювальна складність зростає на один порядок.

3.2.4 Приклад на основі кардіологічних даних

Сучасні системні медичні дослідження (доказова медицина) представляють інтеграцію найкращих наукових доказів із клінічним досвідом та очікуваннями пацієнтів [160]. Вони проводяться з метою покращення охорони здоров’я в майбутньому. Системні медичні дослідження допомагають лікарям та дослідникам отримати знання про здоров’я та хвороби людини. Вони також дозволяють знайти більш ефективні способи профілактики та лікування захворювань. Оцінка стану здоров’я базується на всебічному та систематичному дослідженні пацієнта, яке включає анамнез, об’єктивне обстеження стану організму, аналіз результатів лабораторних досліджень крові та різних виділень, інструментальні та інтервенційні дослідження, включаючи рентген, КТ, МРТ, ендоскопію, біопсію та інші методи.

У наш час серцево-судинні захворювання привертають увагу, оскільки вони є «причиною смерті номер 1 у світі» [45]. При дослідженні кардіологічних захворювань існує досить велика кількість нюансів та показників, на які фахівці звертають увагу під час діагностики. Діагностичні критерії включають як фізичні тести, так і особливості анамнезу, а також лабораторні, інструментальні методи дослідження. Під час опитування лікар може задати питання про членів сім'ї пацієнта (генетична схильність), спосіб життя та звички. Фізична бездіяльність (сидячий спосіб життя), нездорове харчування, вживання алкоголю та тютюнопаління значно збільшують ризик розвитку серцево-судинних захворювань. Під час лабораторних досліджень велика увага приділяється оцінці рівня ліпідів та їх фракцій (ліпідного профілю). Він включає показники загального холестерину, тригліцеридів, високої, низької, дуже високої та дуже низької щільності ліпопротеїдів, а також рівень атерогенності. Дисбаланс ліпідів збільшує ризик розвитку атеросклерозу. Крім усього іншого, надмірна вага пацієнта є одним із небезпечних факторів ризику серцевої патології. Глюкоза в крові та глікований гемоглобін є одними з найважливіших показників вуглеводного обміну в організмі та маркерами цукрового діабету. Цукровий діабет є окремою хворобою, але його наявність значно збільшує ризик розвитку серцево-судинних захворювань. На додаток до оцінки ризику проводяться необхідні розширені гематологічні, біохімічні та інструментальні дослідження. Окрім загального аналізу крові, у пацієнта вимірюють рівень артеріального тиску, застосовують такі інструментальні методи: електрокардіограму (ЕКГ), холтерівський моніторинг, ЕхоКГ, коронарну ангіографію, МР-ангіографію.

Дане експериментальне дослідження включає дані 1651 пацієнта з діагнозом інфаркт міокарда. Цільовим атрибутом прогнозування є тривалість життя. Дані кожного пацієнта включають 97 атрибутів, що містять як числові, так і категоріальні значення. Така інформація охоплює дані про тип інфаркту (вогнищевий або трансмуральний), локалізацію інфаркту (передній

або задній). Також використовується інформація про смертність (в лікарні, короткострокова та довгострокова). Описано наявність супутніх патологій. І тут використовуємо детальний аналіз, оскільки такі патології можна поєднувати. Досліджуються фактори ризику, типові для серцево-судинних захворювань, а саме, клінічна оцінка включає дані про такі фактори ризику як наявність гастриту, жовчнокам'яної хвороби, легеневої хвороби, нефрологічних розладів, ревматичних захворювань щитовидної залози, ангіопатології, шлунково-кишкових захворювань, онкології, хронічної обструктивної хвороби легенів, артеріальної гіпертензії, цукрового діабету, куріння.

Описано детальний клінічний курс: включаються показники життєво важливих функцій, а саме: частота серцевих скорочень, систолічний артеріальний тиск та діастолічний артеріальний тиск, аналізуються ускладнення інфаркту у вигляді порушень ритму, зокрема, детальні ускладнення інфаркту, розвинені в лікарні. У даних перераховані всі показники загального аналізу крові. Особлива увага приділяється лейкоцитам (WBC), представлений біохімічний аналіз крові, включена інформація про ліки, які пацієнт отримував у лікарні.

Після алгоритму зменшення розмірності залишились такі ознаки: стать (sex,), вік (age), повторний інфаркт міокарда (RMI), тривалість життя після ІМ (died_days), індекс маси тіла (BMI), щільність лейкоцитів (White_blood_cells_count) , фракція викиду лівого шлуночка (LVEF).

Розглянули набір Ψ, що включає модель лінійної регресії (regr.lm), ОВМ модель (regr.ksvm) та модель випадкового лісу (regr.ranger).

Стратегії передискретизації включають крос-валідацію cv3, cv5, cv7, cv9, cv10. Функція втрат L обчислювалася як середньоквадратична похибка rmse та час навчання.

У випадку середньоквадратичного відхилення як показника ефективності (Таблиця 3.2), модель ОВМ є розв'язком задачі МН на основі мінімаксного підходу. А саме, спочатку порівнюємо значення помилок для

всіх моделей, що розглядаються. На другому кроці бачимо, що значення середньоквадратичного відхилення для ОВМ-моделі буде мінімальним серед максимальних. На рис. 3.7 можемо побачити аналіз ефективності моделей МН при різних стратегіях передискретизації для середньоквадратичного відхилення як показника ефективності.

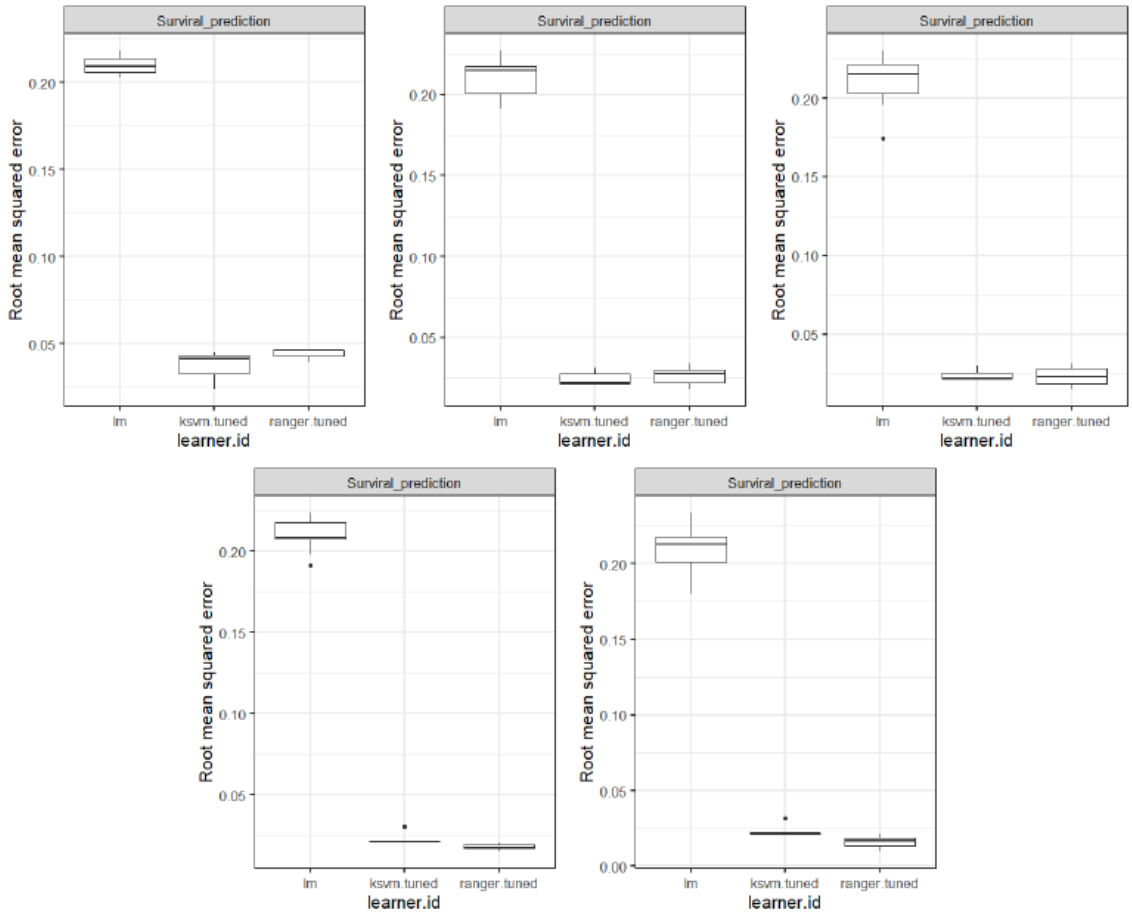


Рисунок 3.7 – Порівняння ефективності моделей МН для різних стратегій передискретизації (cv3, cv5, cv7, cv9, cv10) на основі середньоквадратичного відхилення

Таблиця 3.2 – Аналіз ефективності моделі МН при різних стратегіях передискретизації для середньоквадратичного відхилення як показника ефективності

Стратегії передискретизації	Лінійна регресія	ОВМ модель	Випадковий ліс
cv3	0.2100193167	0.03786423738	0.04423246658
cv5	0.210649921	0.02506639264	0.02696577121
cv7	0.2106224629	0.02404772839	0.02362994412
cv9	0.2098297672	0.02371189948	0.01803804644
cv10	0.2102195218	0.02263394791	0.0164288483
max	0.210649921	0.03786423738	0.04423246658
minimax		**0.03786423738**	

У випадку часу навчання як показника ефективності (Таблиця 3.3), модель лінійної регресії є розв’язком задачі МН на основі мінімаксного підходу. А саме, спочатку порівнюємо значення часу тренування моделі для всіх моделей, що розглядаються. На другому етапі бачимо, що значення лінійної моделі буде мінімальним серед максимальних показників. У свою чергу, на рис. 3.8 бачимо аналіз ефективності моделей МН при різних стратегіях передискретизації у випадку часу навчання як показника ефективності.

Таблиця 3.3 – Час навчання моделі МН

Стратегії передискретизації	Лінійна регресія	ОВМ модель	Випадковий ліс
cv3	0.006666667	6.05333333	15.62000000
cv5	0.0020000	13.98800000	35.35600000
cv7	0.0003222	18.52428571	45.66714286
cv9	0.002222222	25.152222	65.76222222
cv10	0.0050000	28.64400000	76.46600000
max	0.006666667	28.64400000	76.46600000
minimax	**0.006666667**		

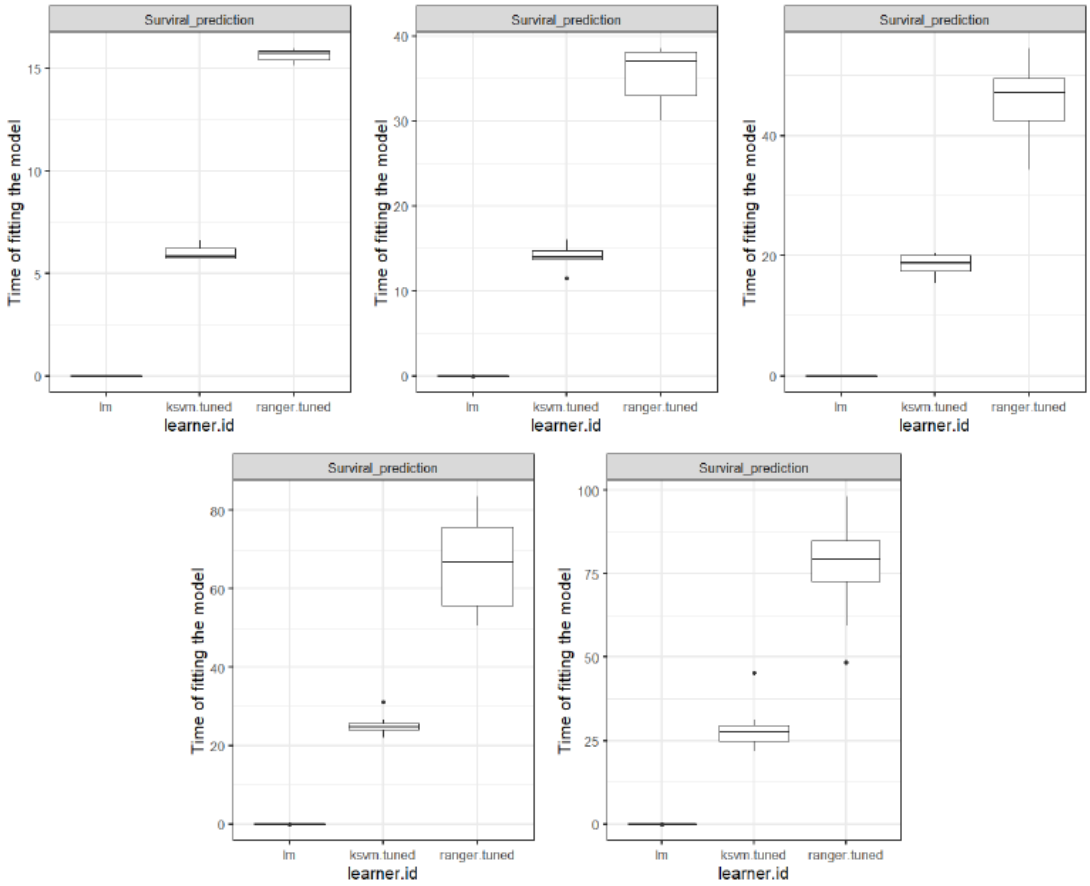


Рисунок 3.8 – Порівняння ефективності моделей ML для різних стратегій передискретизації (cv3, cv5, cv7, cv9, cv10) на основі часу навчання.

На основі наведених прикладів встановлено, що врахування невизначеності у даних (алеаторна невизначеність) суттєво впливає на модель, що будується на основі МН. Наприклад, як видно з таблиці 3.2 існує стратегія передискретизації (а саме cv10), на якій модель випадкового лісу показує найменше значення функції втрат середньоквадратичного відхилення на множині усіх моделей, що розглядаються. У той же час, існують стратегії передискритизації (cv3), на яких дана модель показує більші похибки порівняно з моделлю ОВМ. У цій ситуації вибір моделі випадкового лісу призвів би до неочікуваних втрат, що виникають з алеаторної невизначеності. Тому мінімаксний підхід пропонує встановити стратегію передискретизації з максимальним («найгіршим») значенням функції втрат, на якій досліджувана модель повинна повести себе найкраще (отримати мінімальне значення функції втрат).

Висновки до третього розділу

1. Розглянуто задачу класичного МН інтегровану з даними із систем з відкритим кодом електронних медичних записів в умовах визначеності. А саме: здійснено постановку задачі МН інтегрованих з даними із систем з відкритим кодом електронних медичних записів; наведено математичний опис задачі МН в системних медичних дослідженнях в умовах визначеності; розроблено модель МН для системних медичних досліджень; розглянуто та вирішено задачу імпорту даних із систем з відкритим кодом електронних медичних записів; розглянуто задачі зменшення розмірності та класифікації; наведено результати експериментальних досліджень, пов'язаних з прогнозуванням переломів для жінок середнього віку, що є на сьогодні актуальною, профілактичною задачею. У результаті експериментальних досліджень отримано моделі МН на основі п'яти методів. Результати ефективності методів, за допомогою середньої помилки неправильної класифікації представлені. Досліджено, що найточнішою моделю відповідно до оцінки якості прогнозування є метод випадкового лісу, найгіршим є метод випадкових папоротей. Встановлено, що модель на основі випадкового лісу дозволила покращити прогнозування переломів у $25.12\% \pm 2.10\%$ випадків.

2. Розглянуто задачу класичного МН в умовах невизначеності та запропоновано метод розв'язування на основі мінімаксного критерію. А саме: здійснено математичний опис задачі МН в умовах невизначеності; розглянуто питання зменшення розмірності в умовах невизначеності, запропоновано модифікацію методу МГК щодо стратегії передискретизації γ; представлено загальну блок-схему побудови моделі МН на основі мінімаксного критерію в умовах невизначеності. Це дозволило розробити алгоритм побудови моделі на класах розподілів навчальних та тестових даних на множині ряду методів МН.

3. Реалізовано на основі розробленої схеми МН приклад для кардіологічних даних. Експериментальне дослідження включає дані 1651 пацієнта з діагнозом інфаркт міокарда. Цільовим атрибутом прогнозування є

тривалість життя. Дані кожного пацієнта включають 97 атрибутів, що містять як числові, так і категоріальні значення. Розглянули набір Ψ, що включає модель лінійної регресії, метод опорних векторів та модель випадкового лісу. У випадку середньоквадратичного відхилення, як показника ефективності, модель на основі методу опорних векторів є розв'язком задачі МН на основі мінімаксного підходу. У випадку часу навчання як показника ефективності, модель лінійної регресії є розв'язком задачі МН на основі мінімаксного підходу.

4. Розроблено метод зменшення розмірності даних із втратою найменшої кількості інформації на основі аналізу головних складових (МГК) в умовах алеаторної невизначеності. Це дало змогу враховувати лише атрибути з найбільшими значеннями поясненої варіації на множині стратегій передискретизації, що розглядаються. У прикладі, що розглядається, на основі кардіологічних даних зменшення розмірності атрибутів дозволило скоротити обчислювальну складність в 91 раз.

5. У випадку застосування перехресної перевірки як стратегії передискритизації встановлено, що врахування алеаторної невизначеності веде до збільшення обчислювальної складності на один порядок, у той же час на розглядувальній множині моделей дозволило покращити середньоквадратичну похибку на 17,30 %.

РОЗДІЛ 4. ГЛИБОКЕ МАШИННЕ НАВЧАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ НА ОСНОВІ МІНІМАКСНОГО ПІДХОДУ ТА РОЗРОБКА МОДЕЛІ SCIML

У розділі пропонується реалізація архітектури нейронної мережі на основі щільної U-Net мережі у середовищі Google Collab на загальнодоступному наборі даних DRIVE з використанням бібліотеки МН Tensorflow для сегментації судин сітківки очного дна. У випадку бінарної крос-ентропії як показника ефективності, модель U-Net з плитками 8x8 є розв'язком задачі МН на основі мінімаксного підходу. У випадку з часом навчання як показником ефективності, що продемонстровано у таблиці, модель U-Net з плитками 25x25 є розв'язком задачі МН на основі мінімаксного підходу.

Запропоновано метод розв'язування задачі класифікації кіберфічних біосенсорних систем на основі алгоритму наукового МН, що дозволяє оптимальним чином підібрати параметри системи, а саме: початкові умови, швидкісні параметри та час імунної відповіді, при цьому забезпечивши операційну або самостійкість системи.

Основні результати цього розділу опубліковано в роботах [120], [8], [26].

4.1 Застосування згорткових мереж для завдань класифікації

Кровоносні судини сітківки є основою для клінічної діагностики деяких захворювань. Досягнення автоматичної сегментації судин сітківки на знімках очного дна є важливою та складною роботою. У цьому розділі пропонується архітектура нейронної мережі на основі щільної U- Net мережі.

Згорткові мережі існували вже давно [92], їх успіх був обмежений через розмір доступних навчальних наборів та розмір розглянутих мереж. Прорив Крижевського та інших [88] відбувся завдяки навчанні з вчителем великої мережі з 8 шарами та мільйонами параметрів набору даних ImageNet

з 1 мільйоном навчальних зображень. Відтоді навчили ще більші та глибші мережі [165].

Типове використання згорткових мереж застосовується для завдань класифікації, де вихідним зображенням є одна мітка класу. Однак у багатьох візуальних завданнях, особливо при біомедичній обробці зображень, бажаний результат повинен включати локалізацію, тобто мітка класу повинна призначатися кожному пікселю. Більше того, тисячі навчальних зображень, як правило, недосяжні для біомедичних завдань. Отже, Сайзен та інші [51] навчили мережу в установці з розсувним вікном передбачати мітку класу кожного пікселя, надаючи локальну область (патч) навколо цього пікселя. По-перше, ця мережа може локалізуватися. По-друге, навчальні дані з точки зору патчів набагато більші, ніж кількість навчальних зображень.

Варто розглянути мережеву архітектуру зображену на рис. 4.1, яка проілюстрована у [157]. Вона складається із контрактуючого шляху (ліва сторона) та експансивного шляху (права сторона). Шлях укладання відповідає типовій архітектурі згорткової мережі. Він складається з багаторазового застосування двох звивин 3x3 (незмінених звивин), кожна з яких випрямлена лінійна одиниця (ReLU) і операція об'єднання макс. 2x2 з кроком 2 для зниження вибірки. На кожному кроці зменшення вибірки подвоюється кількість функціональних каналів. Кожен крок у експансивному шляху складається з передискретизації карти об'єктів, за якою слідує згортка 2x2 («згортання згори»), яка зменшує вдвічі кількість каналів об'єктів, конкатенацію з відповідно обрізаною картою об'єктів із контрактуючого шляху та дві 3x3 згортки, кожна з яких супроводжується ReLU. Обрізання необхідне через втрату пікселів межі в кожній звивині. На кінцевому шарі згортка 1x1 використовується для відображення кожного 64-компонентного вектора ознак до бажаної кількості класів. Загалом мережа має 23 згорткові шари.

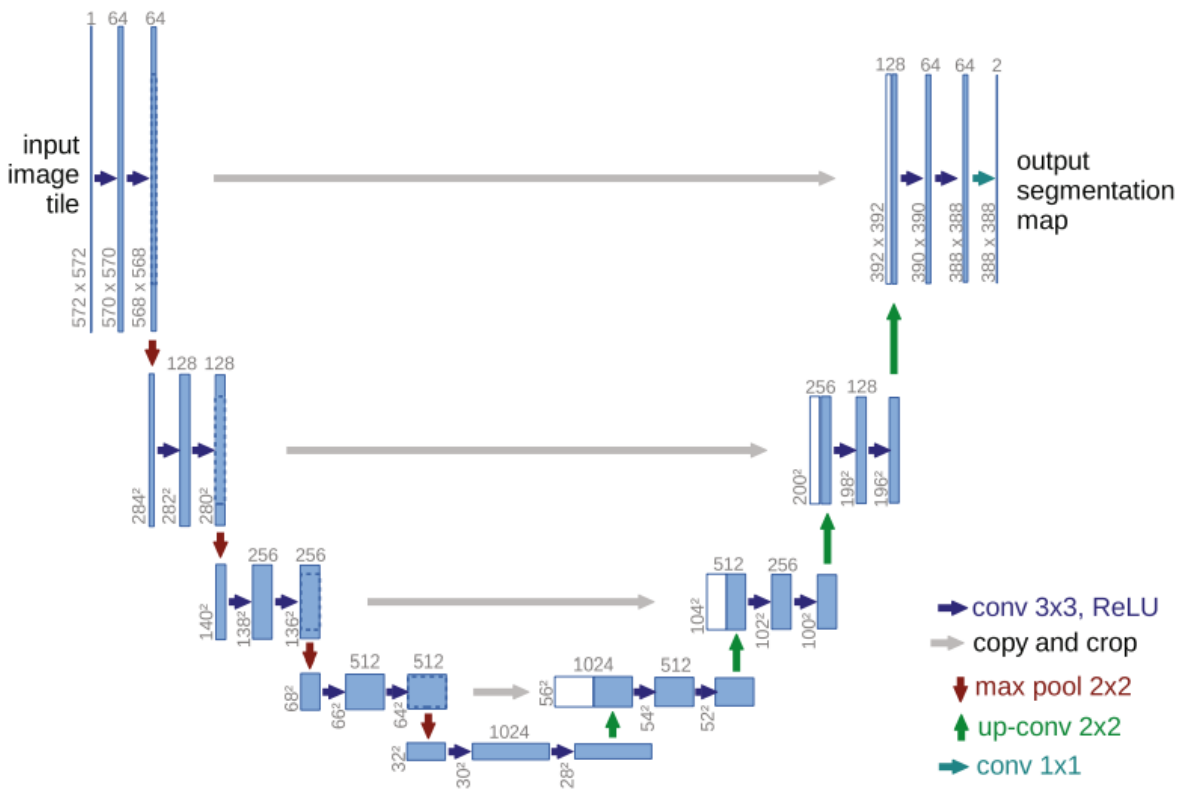


Рисунок 4.1 – Архітектура U-Net (приклад для 32x32 пікселів із найнижчою роздільною здатністю). Кожен синій ящик відповідає багатоканальній карті функцій. Кількість каналів позначається у верхній частині вікна. Розмір x-y розміщений у нижньому лівому краї коробки. Білі поля представляють скопійовані карти об’єктів. Стрілки позначають різні операції.

Кровоносні судини сітківки містять значну інформацію, що стосується здоров'я людини. Спостереження за морфологічною структурою судин сітківки на знімках очного дна використовується не тільки для скринінгу судинних захворювань сітківки, але і для допоміжної діагностики інших захворювань, таких як інсульт [50], гіпертонія [63], індукована діабетом ретинопатія [157] та глаукома [29]. Зображення очного дна мають такі характеристики: низький контраст між судиною та фоном, серйозні втручання в зону ураження та складна судинна структура, що створює багато задач для досягнення сегментації судин сітківки [169]. На сьогодні основним методом сегментації судин сітківки є ручна анотація професійними лікарями.

Однак із постійним розвитком медичної технології візуалізації все більше зображень очного дна потребує сегментації. Ручне анотування судин сітківки вимагає від лікарів багато часу та енергії, і різні лікарі завжди приймають різні стандарти сегментації сітківки. Тому для автоматизованої діагностики (Computer-Aided Diagnosis (CAD)) було запропоновано багато алгоритмів автоматичної сегментації судин сітківки. Досягнення автоматичної сегментації судин сітківки може не тільки зменшити навантаження лікарів, але й уникнути суб'єктивного впливу різних лікарів на результати сегментації.

Відповідно до узагальнених алгоритмів сегментації судин сітківки [33], [134], його можна розділити на дві категорії: алгоритми з вчителем та алгоритми без вчителя. Загалом, алгоритми навчання без вчителя не вимагають даних ручної анотації і в основному використовують деякі заздалегідь встановлені правила для вилучення особливостей судин та досягнення сегментації, такі як відповідність алгоритмів, заснованих на фільтрах [166]; алгоритмів, що деформуються, на основі моделей [85] та заснованих на відстеженні алгоритми [185]. Однак фіксовані правила сегментації часто не можуть відповідати різноманітності морфологічного розподілу судин.

Основна ідея алгоритмів навчання з вчителем полягає у навчанні моделі сегментації із використанням зображень очного дна з анотаціями сегментації, що дозволяють моделі автоматично витягувати особливості судин для досягнення сегментації судин, наприклад, алгоритми на основі моделі Байєса [178], підтримують векторну машину на основі алгоритмів [176] та алгоритмів глибокого навчання [83], [184], [72], [61]. Однак алгоритми навчання з вчителем вимагають величезних даних із ручним маркуванням, які важко отримати.

В останні роки алгоритми, засновані на глибокому навчанні, продовжували розроблятися і добре працювали в області сегментації судин сітківки, які поступово стали основним алгоритмом. Після Лонга та інших

[98] запропоновано повністю згорнуту нейронну мережу (FCN), різні алгоритми сегментації судин сітківки, що базуються на структурі FCN, постійно виходять. Олівейра та інші [141] представили алгоритм сегментації судин сітківки, що поєднує багатомасштабне вейвлет-перетворення та багатомасштабний FCN. Лу та інші [99] використовували згорткову нейронну мережу (CF-FCN) для вилучення кровоносних судин на зображеннях очного дна. CF-FCN мав на меті використовувати вихідну інформацію про дані та компенсував вихід нейронної мережі за допомогою просторового співвідношення між пікселями на зображенні очного дна. Однак результати сегментації структури, що базується на FCN, не мали належних результатів у деталях. Тому йому зазвичай потрібні були деякі методи обробки, такі як умовне випадкове поле (CRF) та випадкове поле Маркова (MRF).

Серед алгоритмів сегментації на основі FCN структура U-net, запропонована Роннебергером [157], змінила метод підключення переходу на карту функцій у FCN з додавання на конкатенат, який широко використовується в галузі сегментації біомедичних зображень. Крім того, для сегментації судин сітківки з’явилася велика кількість вдосконалених алгоритмів, заснованих на U-Net моделі. Наприклад, Сяо та інші [189] пропонують модель на основі U-мережі із зваженим затвором уваги для сегментації малих судин сітківки. Гао та інші [67] поєднують фільтрування за Гаусом та U-Net для досягнення сегментації судин сітківки. Алом та інші [34] пропонують дві вдосконалені моделі на базі U-Net. Перший – це періодична згорткова нейронна мережа (RCNN), а другий – повторювана залишкова згорткова нейронна мережа (RRCNN), обидві успішно застосовуються для сегментації судин сітківки.

Порівнюючи різні моделі МН, метою є мінімізація очікуваних втрат. Але також потрібно взяти до уваги стратегії передискретизації, за якими також повинні оцінювати функцію втрат. Таке формулювання задачі МН розглядає два види невизначеності. А саме, невизначеність у

передискретизації є алеаторичною, оскільки вона пов'язана з даними. У той же час невизначеність при виборі моделей є епістемічною. Математично мінімальна задача МН описується як пошук моделі h завдяки

$$\arg\min_{h\in\Psi}\max_{\gamma\in\Gamma}\mathrm{E}[L(\mathrm{Y},h(X,\gamma))] \tag{4.1}$$

4.2 Реалізація U-Net моделі з використанням TensorFlow то оцінка результатів з використанням мінімаксного підходу.

Нижче представлено реалізацію простої U-Net моделі для сегментації кровоносних судин сітківки. Модель реалізовано на основі бібліотек Keras та Tensorflow. Метод протестовано на загальнодоступному наборі даних DRIVE, що складається з 40 зображень. Для завдань прогнозування використано крос-валідацію як одну зі стратегій передискретизації. Стратегії передискретизації включають cv3, cv5, cv7, cv9.

Для нормалізації зображень використовується контрастне обмежене адаптивне вирівнювання гістограми (CLAHE). При цьому зображення ділиться на невеликі блоки, які називаються «плитками» (за замовчуванням tileSize становить 8x8 у OpenCV). Потім кожен із цих блоків вирівнюється. Отже, на невеликій ділянці гістограма обмежується лише невеликою областю (якщо немає шуму). Якщо шум є, він буде посилений. Щоб уникнути цього, застосовується обмеження контрасту. Якщо перевищено вказану межу контрастності (за замовчуванням 40 у OpenCV), ці пікселі відсікаються та розподіляються рівномірно в інші контейнери перед тим, як застосовувати вирівнювання гістограми. Після вирівнювання для видалення артефактів у межах плитки застосовується білінійна інтерполяція. Використовується гамма-корекція за допомогою таблиці пошуку.

Для нормалізації зображення було поділено на плитки розміром 8x8, 16x16, 25x25, 40x40.

Було використано оптимізатор Адама з *initial_learning_rate=0.0003*, *first_decay_steps=12000*, *t_mul=1000*, *m_mul=0.5*, *alpha=1e-5*.

Для оцінки методу використовуємо точність (Precision), повноту (Recall), матрицю помилок (Confusion matrix), площу під кривою помилок (Receiver Operating Characteristic Curve (AUC-ROC)) (Рис. 4.2 і 4.3).

```
print("average dice score for all predict vessel masks:",np.mean(dice_list))
print("average AUC for all predict vessel masks:",np.mean(roc_list))
print("average PR for all predict vessel masks:",np.mean(pr_list))
print("average recall(sensitivity) for all predict vessel masks:",np.mean(tpr_list))
print("average specificity for all predict vessel masks:",np.mean(tnr_list))
```

```
average dice score for all predict vessel masks: 0.8741437
average AUC for all predict vessel masks: 0.87187457
average PR for all predict vessel masks: 0.6971755
average recall(sensitivity) for all predict vessel masks: 0.76676553
average specificity for all predict vessel masks: 0.9879815
```

Рисунок 4.2 – Обчислення метрик Dice Score, Precision, Recall, AUC при використанні плитки 25x25

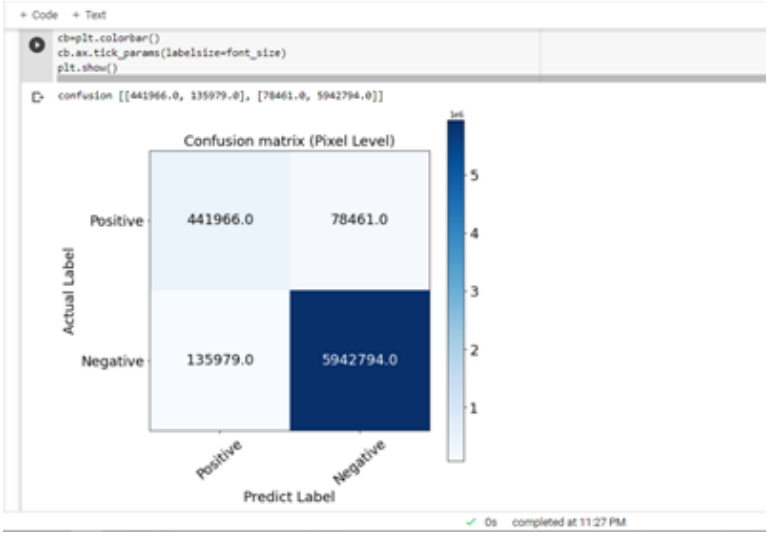


Рисунок 4.3 – Матриця помилок для оцінки моделі при використанні плитки 25x25

Для класифікації зображень було експортовано зображення у вигляді 8, 16, 25 та 40 однакових за розміром плиток для застосування глибокого навчання.

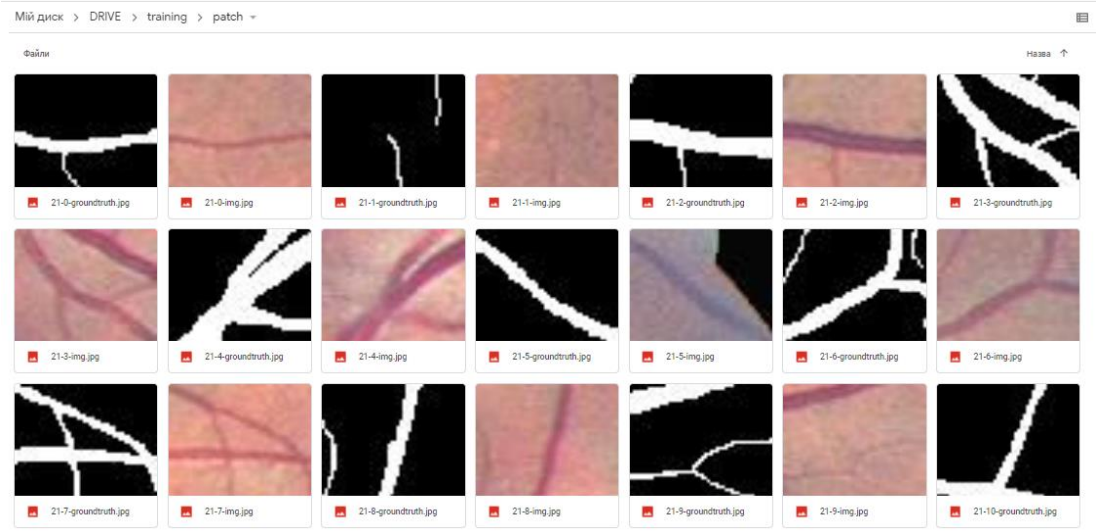


Рисунок 4.4. – Експортовані зображення при використанні плитки 25x25

Сам процес формування плиток для застосування глибокого навчання представлено на рис. 4.5.

Рисунок 4.5 – Процес формування тренувальних плиток у середовищі Google Collab при використанні плитки 25x25

Для визначення помилки між реальними і отриманими результатами та її мінімізації в якості функції втрат використовували бінарну крос-ентропію (binary cross entropy).

У випадку бінарної крос-ентропії як показника ефективності, що продемонстровано у таблиці 4.1, модель U-Net з плитками 8x8 є розв’язком задачі МН на основі мінімаксного підходу. На першому кроці, порівнюємо значення функції втрат для всіх розглянутих моделей. На другому кроці бачимо, що значення бінарної крос-ентропії для моделі U-Net з плитками 8x8 буде мінімальним серед максимальних характеристик.

У випадку з часом навчання як показника ефективності, що продемонстровано у таблиці 4.2, модель U-Net з плитками 25x25 є розв’язком задачі на основі мінімаксного підходу. На першому кроці, спочатку порівнюємо значення часу тренування всіх моделей, що розглядаються. На другому кроці бачимо, що значення часу для моделі U-Net з плитками 25x25 буде мінімальним серед максимальних показників.

Таблиця 4.1 – Аналіз ефективності моделі МН при різних стратегіях передискретизації для бінарної крос-ентропії як показника ефективності

Стратегії передискретизації	***Розмір плиток***			
	8*8	***16*16***	***25*25***	***40*40***
cv3	0.0342	0.0349	0.0353	0.0347
cv5	0.0301	0.0311	0.0374	0.0345
cv7	0.0295	0.0302	0.0323	0.0328
cv9	0.0303	0.0298	0.0318	0.0307
max	0.0342	0.0349	0.0374	0.0347
minimax	**0.0342**			

Таблиця 4.2 – Аналіз ефективності моделі МН при різних стратегіях передискретизації для часу навчання моделі як показника ефективності

Стратегії передискретизації	***Розмір плиток***			
	8*8	***16*16***	***25*25***	***40*40***
cv3	3.207	3.167	4.301	3.252
cv5	6.267	4.312	3.883	4.321
cv7	8.812	5.896	5.231	6.127

Продовження таблиці 4.2

cv9	12.356	9.564	8.462	9.052
max	12.356	9.564	8.462	9.052
minimax			**8.462**	

Процес обчислення функції втрат та індексу Соренса відбувався безпосередньо у програмному середовищі.

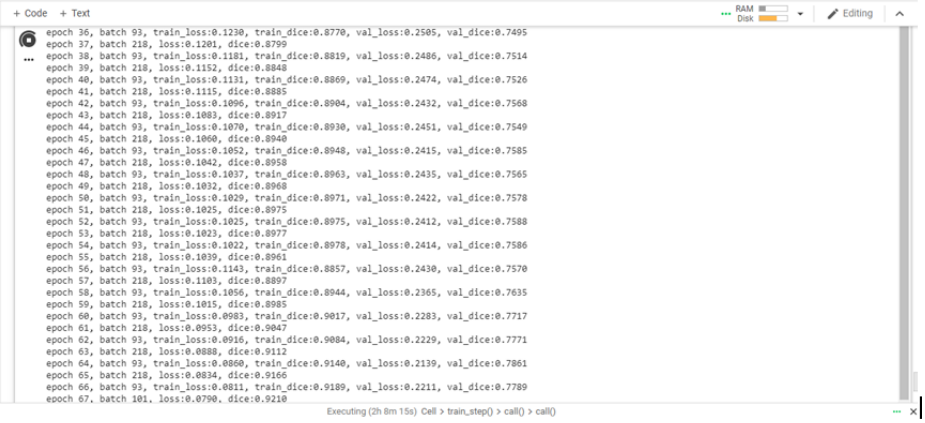


Рисунок 4.6 – Процес обчислення функції втрат та індексу Соренса у середовищі Google Collab при використанні плитки 25x25

Результат роботи нейромережі добре прослідковується,коли порівняти, наприклад, початкове зображення очного дна на одинадцятому зображенні (Рис. 4.7) і результати роботи U-Net моделі представлені на рис. 4.8 з розпізнаними кровоносними судинами для зображення під номером одинадцять.

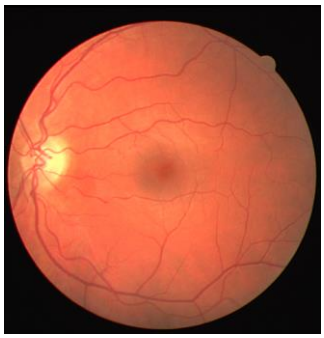


Рисунок 4.7 – Початкове зображення очного дна на одинадцятому зображенні при використанні плитки 25x25

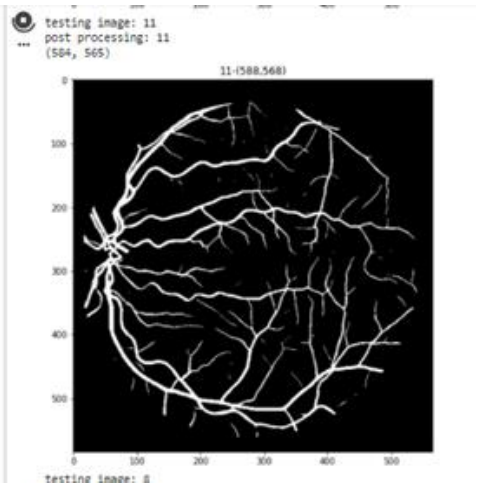


Рисунок 4.8 – Результати роботи U-net моделі для одинадцятого зображення при використанні плитки 25x25

Подальший розвиток досліджень вбачається у застосуванні мінімаксного підходу до порівняльного аналізу ефективності роботи моделей глибокого машинного навчання, зокрема загорткових нейронних мереж, для опрацювання зображень у системних медичних дослідженнях.

4.3 Опис біологічних процесів, що лежать в основі кіберфізичних біосенсорних систем

Відомий спосіб багатомасштабованої платформи для координації клітинної активності з використанням синтетичної біології [135]. Така платформа зменшує вплив шумів, робить вихідні сигнали когерентнішими та надійнішими на макроскопічному рівні; біосенсор містить мікрорідинний масив двох або більше колоній або популяцій чутливих клітин. З метою моделювання просторової синхронізації осцилюючих колоній в мікрорідинному масиві використано модель на основі решітчастих диференціальних рівнянь на прямокутній решітці [149].

Недоліком вказаного способу є неможливість його використання для кіберфізичних біосенсорних систем, які складаються з двовимірного масиву пікселів і які керуються дискретними подіями, що важливо при їх проектуванні.

В основу корисної моделі поставлено завдання вибору оптимальних параметрів (швидкісних параметрів популяцій та величини часу імунної відповіді), які гарантують оптимальні значення алгоритму наукового МН кібер-фізичної біосенсорної системи. Запропонована модель може бути застосована з використанням довільних методів data mining – як з побудовою дерев рішень, так і класифікаційних правил. При цьому не вимагається модифіковувати методи розв'язування систем решітчастих нелінійних диференціальних рівнянь, оскільки траєкторії, згенеровані для випадкових значень параметрів, використовуються в подальшому в якості навчальних наборів класифікаційних алгоритмів. Такий підхід дозволяє ефективно пришвидшувати процес вибору оптимальних значень параметрів в кіберфізичних системах. Опис кіберфізичної біосенсорної системи на прямокутній решітці наведено в роботах [124], [25], з використанням гексагональної решітки – в роботах [122], [13].

Важливо представити конкуренційну модель антитіло-антигени для біопікселів двохмірного масиву відповідно до роботи [117]

$$\begin{aligned} \frac{dV_{i,j}(t)}{dt} &= (\beta - \gamma F_{i,j}(t-\tau)) - \delta_{\upsilon} V_{i,j}(t-\tau) V_{i,j}(t) + \hat{S}\{V_{i,j}\} \\ \frac{dF_{i,j}(t)}{dt} &= (-\mu_f + \eta\gamma V_{i,j}(t-\tau) - \delta_f F_{i,j}(t) F_{i,j}(t)) F_{i,j}(t) \end{aligned} \tag{4.2}$$

Модель (4.2) задана початковими умовами (4.3):

$$\begin{aligned} &V_{i,j}(t) = V_{i,j}^0(t) \geq 0, \quad F_{i,j}(t) = F_{i,j}^0(t) \geq 0, \quad t \in [-\tau, 0), \\ &V_{i,j}(0), \quad F_{i,j}(0) > 0. \end{aligned} \tag{4.3}$$

Загальний підхід до розробки біопікселя представлено у [24].

4.4 Алгоритм наукового машинного навчання кіберфізичної біосенсорної системи

Алгоритм науковго МН кіберфізичної біосенсорної системи полягає в таких кроках:

1. Означення класів траєкторій систем. Використовується термінологія класів, які відповідають $\{G_{0,}\Phi_{t,}\mathrm{T}\}$ стійкості і нестійкості. Для того, щоб позначити клас траєкторії, ми вводимо атрибут класу C, який приймає одне з двох бінарних значень одиниць $C \in \{0,1\}$.

2. Генерування матриці випадкових початкових значень запізнення часу і параметрів швидкості. Припускається, що початкові значення запізнення часу і параметри швидкості є рівномірно розподілені на інтервали. Кожний стовпчик відповідає множині значень одного параметра або початкового значення запізнення часу або констант швидкості. Кожен рядок – це множина початкових значень і констант швидкості для одного запуску моделі на основі (4.2), (4.3).

$$M = \begin{Bmatrix} \beta^1 & \gamma^1 & \dots & F^1_{N,N} \\ \beta^2 & \gamma^2 & \dots & F^2_{N,N} \\ \dots & \dots & \dots & \dots \\ \beta^M & \gamma^M & \dots & F^M_{N,N} \end{Bmatrix} \in R^{N \times N+7} \tag{4.4}$$

3. Запуск моделі і класифікація вхідних даних. Кожна множина початкових значень запізнення часу і параметрів швидкості, яка згенерована на другому кроці, використовується як вхід для моделей біосенсору. Вихідні траєкторії класифікуються на основі критеріїв, запропонованих на першому кроці. На основі результатів класифікації відповідне значення атрибуту класу 0 або 1 присвоюється початковим значенням із запізненням часу та швидкісним параметром.

4. Побудова матриці зв’язків між початковими значеннями, запізненням в часі та між параметрами швидкості. Метод припускає, що для форми траєкторії системи взаємозв’язок між початковими значеннями запізненням в часі та константами швидкості є важливішим ніж їх абсолютні

значення. Отже, матриця, яка відповідає інформації в категоризованому вигляді про взаємозв'язки між початковими значеннями запізненням в часі і швидкісними параметрами, які генеруються на другому кроці має наступний вигляд

$$D = \begin{Bmatrix} \beta \otimes \gamma & \beta \otimes \delta_{\upsilon} & \dots & C \\ p(\beta^1, \gamma^1) & p(\beta^1, \delta_{\upsilon}^1) & \dots & C_1 \\ p(\beta^2, \gamma^2) & p(\beta^2, \delta_{\upsilon}^2) & \dots & C_2 \\ \dots & \dots & \dots & \dots \\ p(\beta^M, \gamma^M) & p(\beta^M, \delta_{\upsilon}^M) & \dots & C_M \end{Bmatrix} \tag{4.5}$$

де

$$p(u, \upsilon) = \begin{cases} 0, & якщо \quad u < \upsilon \\ 1, & якщо \quad u = \upsilon \\ 2, & якщо \quad u > \upsilon \end{cases} \tag{4.6}$$

$C_1 \in \{0,1\}$ – це значення атрибуту класу, яке пов'язане із виглядом відповідних траєкторій. Отже, на цьому кроці числові значення початкових значень – запізнення часу і параметри швидкості, перетворюються в категоризовані значення атрибутів для навчальних даних. Оскільки ймовірність випадкового значення дорівнює нулю, то матриця є свого роду бінаризацією взаємовідносин між початковими значеннями запізненням часу і швидкісними параметрами. Тобто матриця включає лише значення 0 і 2.

Застосування індукції дерева рішень алгоритму C5.0 до взаємозв'язку між початковими значеннями і параметрами швидкості. Матриця бінарних взаємозв'язків побудована в попередньому пункті містить множину навчальних даних для алгоритму data mining, наприклад, індукції дерева рішень. Дерево рішень включає верифікації взаємозв'язків між початковими значеннями, запізненням часу і параметрами швидкості у своїх вузлах. Класи траєкторії моделі $C \in \{0,1\}$ є листками дерева.

Приклад реалізації наукового машинного навчання для класифікації кіберфічних біосенсорних систем представлено у додатку В.

Висновки до четвертого розділу

1. Для розв’язку задачі глибокого машинного навчання на основі мінімаксного підходу пропонується реалізація архітектури нейронної мережі на основі щільної U-Net мережі у середовищі Google Collab на загальнодоступному наборі даних DRIVE з використанням бібліотеки машинного навчання Tensorflow для сегментації судин сітківки очного дна. Для визначення помилки між реальними і отриманими результатами та її мінімізації в якості функції втрат використано бінарну крос-ентропію. Для оцінки методу використано точність (Precision), повноту (Recall), матрицю помилок (Confusion matrix), площу під кривою помилок (Receiver Operating Characteristic Curve (AUC-ROC)). Для нормалізації зображення було поділено на плитки розміром 8x8, 16x16, 25x25, 40x40.

2. У випадку бінарної крос-ентропії як показника ефективності, модель U-Net з плитками 8x8 є розв’язком задачі машинного навчання на основі мінімаксного підходу. На першому кроці порівнюється значення функції втрат для всіх розглянутих моделей. На другому кроці визначено, що значення бінарної крос-ентропії для моделі U-Net з плитками 8x8 буде мінімальним серед максимальних характеристик.

3. У випадку з часом навчання як показником ефективності модель U-Net з плитками 25x25 є розв’язком задачі машинного навчання на основі мінімаксного підходу що дозволив покращити показник ефективності на 31,5%. На першому кроці, спочатку порівнюється значення часу тренування всіх моделей, що розглядаються. На другому кроці визначено, що значення часу для моделі U-Net з плитками 25x25 буде мінімальним серед максимальних показників.

4. Розглядається кіберфізична модель для біосенсорних систем, що грунтується на математичному описі неперервної динаміки популяцій у поєднанні з динамічною логікою, яка використовується для дискретних подій. Дослідження стійкості грунтується на алгоритмі наукового машинного навчання, що пропонує розв’язування задачі класифікації кіберфічних

біосенсорних систем. Алгоритм наукового МН включає в себе такі етапи: означення класів траєкторій систем, генерування матриці випадкових початкових значень запізнення часу і параметрів швидкості, запуск моделі і класифікація вхідних даних, побудова матриці зв'язків між початковими значеннями, запізненням в часі та між параметрами швидкості, застосування індукції дерева рішень алгоритму C5.0 до взаємозв'язку між початковими значеннями і параметрами швидкості. Алгоритм наукового МН дозволяє оптимальним чином підібрати параметри кіберфізичної біосенсорної системи (початкові умови, швидкісні параметри та час імунної відповіді), при цьому забезпечивши операційну або самостійкість системи. На відміну від існуючих методів якісного аналізу динамічних систем на основі теорії стійкості, даний підхід дозволяє прогнозувати можливі стани кіберфізичної біосенсорної системи, кількість яких здатна зростати у випадку технології Big Data, із середньоквадратичною похибкою 5,3%.

ВИСНОВКИ

У дисертаційній роботі розв’язано актуальне наукове завдання, яке полягає у розвитку методів класичного, глибокого та наукового машинного навчання в умовах невизначеності з використанням мінімаксного підходу. У результаті виконання роботи:

1. Здійснено постановку задач класичного і глибокого МН в умовах епістемічної і алеаторної невизначеностей, запропоновано метод розв’язування на основі мінімаксного критерію. Це дозволило розробити алгоритм побудови моделі на класах розподілів навчальних та тестових даних на множині ряду методів МН. У випадку застосування перехресної перевірки як стратегії передискритизації встановлено, що врахування алеаторної невизначеності веде до збільшення обчислювальної складності на один порядок, у той же час на множині моделей, що розглядаються, дозволило покращити середньоквадратичну похибку на 17,30 %. Для задачі глибокого МН у випадку часу навчання як показника ефективності, модель U-Net з плитками 25x25 є розв’язком задачі мінімаксного МН, що дозволив покращити показник ефективності на 31,5%.

2. Реалізовано на основі розробленої схеми МН приклад для кардіологічних даних. Експериментальне дослідження включає дані 1651 пацієнта з діагнозом інфаркт міокарда. Показано, що у випадку середньоквадратичного відхилення, як показника ефективності, модель на основі методу опорних векторів є розв’язком задачі МН на основі мінімаксного підходу. У випадку часу навчання як показника ефективності, модель лінійної регресії є розв’язком задачі МН на основі мінімаксного підходу.

3. Розроблено метод зменшення розмірності даних із втратою найменшої кількості інформації на основі аналізу головних складових (МГК) в умовах алеаторної невизначеності. Це дало змогу враховувати лише атрибути з найбільшими значеннями поясненої варіації на множині стратегій

передискретизації, що розглядаються. У прикладі, що розглядається, на основі кардіологічних даних зменшення розмірності атрибутів дозволило скоротити обчислювальну складність в 91 раз.

4. Отримали подальший розвиток задачі наукового МН, запропоновано метод розв'язування задачі класифікації кіберфічних біосенсорних систем. Це дозволило розробити алгоритм якісного аналізу динамічних моделей медико-біологічних процесів, що зводиться до розв'язування задачі МН. На відміну від існуючих методів якісного аналізу динамічних систем на основі теорії стійкості даний підхід дозволяє прогнозувати можливі стани кіберфізичної біосенсорної системи, кількість яких здатна зростати у випадку технології Big Data, із середньоквадратичною похибкою 5,3%. Алгоритм наукового МН дозволяє оптимальним чином підібрати параметри кіберфізичної біосенсорної системи (початкові умови, швидкісні параметри та час імунної відповіді), при цьому забезпечивши операційну або самостійкість системи

5. Розглянуто задачу класичного МН інтегровану з даними із систем з відкритим кодом електронних медичних записів в умовах визначеності. А саме: здійснено постановку задачі МН інтегрованих з даними з систем з відкрити кодом електронних медичних записів; наведено математичний опис задачі МН в системних медичних дослідженнях в умовах визначеності; розроблено модель МН для системних медичних досліджень. Дані результати використано для експериментальних досліджень, що пов'язані з прогнозуванням переломів для жінок середнього віку. Модель на основі випадкового лісу дозволила покращити прогнозування переломів у $25.12\% \pm 2.10\%$ випадків.

6. Отримали подальший розвиток методи МН на основі хмарних обчислень. Здійснено математичне представлення задачі індукції дерева рішень для реалізації на хмарних платформах. Розглянуто сервіси хмарних платформ та необхідні дії з точки зору розгортання додатків. Це дало змогу реалізувати прототип програми до розв'язування задач МН на віддалених

платформах на основі даних електронних медичних записів з підтримкою технології Big Data для приросту даних та впровадити його на прикладі діагностування серцевої недостатності на основі даних 63-х пацієнтів навчально-практичного центру первинної медико-санітарної допомоги Тернопільського національного медичного університету імені І. Я. Горбачевського.

7. Отримали подальший розвиток методи розробки архітектури інформаційних систем прийняття рішень на основі мікросервісів. Здійснено концептуальне представлення інформаційної моделі задачі прийняття рішень для невідкладної медичної допомоги при настанні серцевої недостатності та представлено підхід до реалізації рішення задачі невідкладної медичної допомоги на основі архітектури мікросервісів, підхід до реалізації заснований на використанні фреймворків Spring, Spring Boot та Spring Cloud frameworks. Це дозволило отримати концептуальне представлення інформаційної моделі для прийняття рішень у системних медичних дослідженнях на основі архітектури мікросервісів, що може бути використаним в проблемах профілактики, діагностики та лікування, та впровадженим в шпитальні та лабораторні інформаційні системи на хмарних платформах. Це дає змогу в запропонованому підході підтримувати технологію Big Data.

Список використаних джерел

[1] Большие данные (Big Data). URL: shorturl.at/pMOST.

[2] Деревья решений: общие принципы. URL: https://loginom.ru/blog/decision-tree-p1.

[3] Дьяконов Олександр. Случайный лес (Random Forest). URL: https://dyakonov.org/2016/11/14/%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9-%D0%BB%D0%B5%D1%81-random-forest/.

[4] Марценюк В. П., Андрущак І. Є. Про концептуальну модель системи підтримки рішень в системних медичних дослідженнях. XIX International Conference Problems of Decision Making under Uncertainties (PDMU- 2012). Abstracts. Mukachevo. Ukraine. 2012. P. 162–163.

[5] Марценюк В. П., Андрущак І. Є., Стаханська О. О. Розробка експертних систем на основі технології Data mining. Здобутки клінічної та експериментальної медицини. Збірник матеріалів підсумкової науково-практичної конференції. 21 червня 2014 року. 2014. С. 141–142.

[6] Марценюк В. П., Сельський П. Р. Ефективність використання телемедичних технологій для покращення якості лікувально-діагностичної роботи на первинному рівні. *Актуальні питання фармацевтичної і медичної науки та практики.* Т. 12. Вип. 3. 2013. С. 53–54.

[7] Марценюк В. П., Сельський П.Р. Ефективність використання інформаційних та телемедичних технологій на первинному рівні надання медичної допомоги. Матеріали науково-практичної конференції з міжнародною участю Інформатизація реабілітаційного процесу. 2013. С. 66– 67.

[8] Марценюк В. П., Андрущак І. Є., Мілян Н. В. Мінімаксний підхід в машинному навчанні: сучасний стан та перспективи розвитку. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво.* Випуск № 42. 2021. С. 164-171. ISSN 2524-0552 (Індексується в Index Copernicus, Google Scholar).

[9] Марценюк В., Кравець Н., Семенець А., Вакуленко Д., Сверстюк А., Климук Н., Сарабун Р., Кучвара О. Про підходи до впровадження emr-систем в галузі охорони здоров'я України. Здобутки клінічної та експериментальної медицини : матеріали підсумкової науково-практичної конференції, присвяченої пам'яті ректора чл.-кор. НАМН України, проф. Л.Я. Ковальчука. Тернопіль. 17 червня. 2015. С. 259–260.

[10] Марценюк В., Мілян Н. Аналіз платформи Google Cloud Platform. Матеріали V Міжнародної науково-технічної конф-ції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 1 – 2 лютого 2018 р.). Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя. 2018. С. 73 – 74.

[11] Марценюк В., Мілян Н. Використання алгоритмів індукції дерева рішень для аналізу великих обсягів даних. Актуальні задачі сучасних технологій : зб. тез доповідей міжнар. наук.-техн. конф. Молодих учених та студентів, (Тернопіль, 28–29 листоп. 2018.). М-во освіти і науки України, Терн. націон. техн. ун-т ім. І. Пулюя [та ін]. Тернопіль : ТНТУ, 2018. С. 114 – 116.

[12] Марценюк В., Мілян Н. Роль і місце хмарних обчислень серед сучасних інформаційних технологій. Фундаментальні та прикладні проблеми сучасних технологій: матеріали Міжнародної науково-технічної конференції до 100-річчя з дня заснування НАН України та на вшанування пам'яті Івана Пулюя (100-річчя з дня смерті), (Тернопіль, 23–24 травня 2018 року). Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя. 2018. С. 219 – 221. ISBN: 978-966-305-093-5.

[13] Марценюк В.П., Сверстюк А.С. Комп'ютерна модель кібер-фізичної імуносенсорної системи з використанням решітчастих диференціальних рівнянь із запізненням на гексагональній решітці. *Науковий журнал Радіоелектроніка, інформатика, управління*. 2019. № 2 (49). С. 131–139.

[14] Марценюк В. П., Сельський П. Р., Семенець А. В. Розробка і впровадження інформаційної системи запису (самозапису) пацієнтів на консультацію до фахівців університетської лікарні. *Український журнал телемедицини та медичної телематики*. Т. 11. №. 2. 2013. С. 173–178.

[15] Мілян Н. Аналіз методів машинного навчання з вчителем. Збірник тез Міжнародної студентської науково-технічної конференції «Природничі та гуманітарні науки. Актуальні питання», 26-27 квітня 2018 року. Т. : ТНТУ, 2018. Том 1. С. 51–52.

[16] Мілян Н. В. Марценюк В. П. Approach to Solving Machine Learning Problems Integrated with Data of the Systems with Open Code of Electronic Medical Records. Вісник ТНТУ. Т. : ТНТУ. 2019. Том 95. № 3. 2019. С. 105–116.

[17] Наконечний О. Г., Марценюк В. П., Баранюк І. О., Сверстюк А. С. Про програмно-технічний комплекс підтримки наукових медичних досліджень. Медичні технології і вища освіта: Матеріали І Всеукраїнської науково-практичної конференції. Луцьк. 28 травня. 2004. С. 92–97.

[18] Наконечний О. Г., Марценюк В. П., Андрущак І. Є. Інформаційні технології прийняття рішень, оптимізації та керування в системних медичних дослідженнях. Луцьк. ЛНТУ. 2014. С. 329. ISBN 978-617-672-090-4.

[19] Наконечний О., Марценюк В., Андрущак І. Методи прийняття рішень, оптимізації та керування в системі підтримки медичних досліджень. XXV International Conference Problems of Decision Making under Uncertainties (PDMU-2015) Abstracts. May 11-15. 2014. Skhidnytsya. Ukraine. 2015. P. 12–13.

[20] Обзор Google App Engine. URL: http://ufacode.ru/blog/web/1123.html.

[21] Работа с Big Data: основные области и возможности. URL: http://www.marketing.spb.ru/lib-around/stat/Big_Data.htm.

[22] Разработка для Google AppEngine: с самого начала. URL: https://habrahabr.ru/post/81895/.

[23] Революция Big Data: Как извлечь необходимую информацию из «Больших Данных»?. URL: http://statsoft.ru/products/Enterprise/big-data.php#analytics.

[24] Сверстюк А. С. Моделі та методи компартментного математичного моделювання кіберфізичних систем медико-біологічних процесів: дис. … док. тех. наук: 01.05.02 – математичне моделювання та обчислювальні методи). Тернопіль, 2020. 611 с.

[25] Сверстюк А.С. Моделювання кібер-фізичної імуносенсорної системи на прямокутній решітці з використанням решітчастих диференціальних рівнянь із запізненням. *Сенсорна електроніка та мікросистемні технології*. 2019. № 2. С. 53–65.

[26] Спосіб кіберфізичного моделювання сенсорного масиву біопікселів. Патент України на корисну модель №139048 МПК G01N33/483 / Н.В. Мілян, В.П. Марценюк, А.С. Сверстюк, А.М. Клос-Вітковська. Заявка № u201901590. Заявл. 18.02.2019. Опубл. 26.12.2019. Бюл.№ 24.

[27] Уокер Г. Основы облачных вычислений URL: https://www.ibm.com/developerworks/ru/library/cl-cloudintro/.

[28] Abbasnejad E. Uncertainty in Machine Learning. URL: https://cs.adelaide.edu.au/~javen/talk/ML05_Uncertainty_in_DL.pdf.

[29] Akil H., Huang A., Francis B., Sadda S., Chopra V. Retinal vessel density from optical coherence tomography angiography to differentiate early glaucoma, pre-perimetric glaucoma and normal eyes. *PLoS ONE*. Vol. 12. No. 2. 2017. Art. no. e0170476.

[30] Alameda-Pineda X., Hospedales T., Ricci E., Sebe N., Wang X. Emerging topics in learning from noisy and missing data. Proceedings of the 2016 ACM on Multimedia Conference. 2016. URL: https://doi.org/10.1145/2964284.2986910.

[31] Ali F., El-Sappagh S., Islam S.M. R., Ali A, Attique M, Imran M., Kwak K. An intelligent healthcare monitoring framework using wearable sensors and social networking data. *Future Generation Computer Systems*. Vol. 114. 2021.

P. 23-43. URL: https://www.sciencedirect.com/science/article/pii/S0167739X1931605X.

[32] Almeida J. Exporting Data from an openEHR Repository to Standard Formats. Procedia Technology. Vol. 16. 2014. P. 1391-1396. URL: https://www.sciencedirect.com/science/article/pii/S2212017314003843.

[33] Almotiri J., Elleithy K., Elleithy A. Retinal vessels segmentation techniques and algorithms: A survey. *Applied Sciences*. Vol. 8. No. 2. 2018. P. 155.

[34] Alom M., Hasan M., Yakopcic C. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. Proc. Comput. Vis. Pattern Recognit. 2018. P. 1–12.

[35] Aminpour F. Utilization of open source electronic health record around the world: A systematic review. *Journal of research in medical sciences*. Vol. 19. 2014. P. 57.

[36] AWS Elastic Beanstalk. URL: https://aws.amazon.com/ru/elasticbeanstalk/.

[37] Barkana B., Saricicek I., Yildirim B. Performance analysis of descriptive statistical features in retinal vessel segmentation via fuzzy logic, ann, svm, and classifier fusion. *Knowledge-Based Systems*. Vol. 118. 2017. P.165–176.

[38] Bertram A. Microsoft Azure vs. Amazon Web Services: Cloud Comparison. URL: http://www.tomsitpro.com/articles/azure-vs-aws-cloud-comparison,2-870.html.

[39] Breiman L. Bagging Predictors. *Machine Learning*. Vol. 24. No. 2. 1996. P. 123-140.

[40] Breiman L. Random forests. *Machine Learning*. Vol. 45. No. 1. 2001. P. 5-32.

[41] Breiman L., Friedman J., Olshen R., Stone C. Classification and regression trees. Wadsworth. Belmont. 1984.

[42] Bühlmann P., Yu B. Boosting with the L2loss: Regression and classification. *Journal of the American Statistical Association*. Vol. 98. No. 462. 2003. P. 324–339. URL: http://dx.doi.org/10.1198/ 016214503000125.

[43] Build What's Next Better software. Faster. URL: https://cloud.google.com/.

[44] Cacciatore V, Carullo A., Parvis M., Vallan A. Uncertainty effects of data compression in measurement applications. 2006 IEEE Instrumentation and Measurement Technology Conference Proceedings. Apr. 2006. URL: https://doi.org/10.1109/imtc.2006.328639.

[45] Cardiovascular diseases. URL: https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1.

[46] Center for Scientific Machine Learning. URL: https://www.oden.utexas.edu/research/centers-groups/scientific-machine-learning/.

[47] Chapman P. Microservices with Spring. URL: https://spring.io/blog/2015/07/14/microservices-with-spring.

[48] Chemweno P., Pintelon L., Muchiri P., and Horenbeek A. Risk assessment methodologies in maintenance decision making: A review of dependability modelling approaches. *Reliability Engineering System Safety*. Vol. 173. 2018. P. 64-77. URL: http:// www.sciencedirect.com/science/article/pii/S0951832016301934.

[49] Chen Z., Fan Z. Distributed customer behavior prediction using multiplex data: A collaborative MK-SVM approach. *Knowledge-Based Systems*. Vol. 35. 2012. P. 111–119.

[50] Cheung C., Ikram M., Chen C., Wong T. Imaging retina to study dementia and stroke. *Progress in Retinal and Eye Research*. Vol. 57. 2017. P. 89–107.

[51] Ciresan D., Gambardella L., Giusti A., Schmidhuber J. Deep neural networks segment neuronal membranes in electron microscopy images. *NIPS*. 2012. P. 2852–2860.

[52] Cohesion (computer science) . URL: https://en.wikipedia.org/wiki/Cohesion_%28computer_science%29.

[53] Cruz H., Rosa C., Oliva A. Immunosensors for diagnostic applications. Parasitology Research. Vol. 88. S4–S7, 2002.

[54] Deng Z., Cao L., Jiang Y., Wang S. Minimax probability TSK fuzzy system classifier: a more transparent and highly interpretable classification model. IEEE Trans. Fuzzy Syst. Vol. 23. No. 4. 2015. P.813–826.

[55] Deng Z., Chen J., Zhang T., Cao L., Wang S. Generalized Hidden-Mapping Minimax Probability Machine for the training and reliability learning of several classical intelligent models. *Information Sciences*. Vol. 436–437. P. 302-319. 2018. URL: https://doi.org/10.1016/j.ins.2018.01.034.

[56] Der Kiureghian A., Ditlevsen O. Aleatory or epistemic? Does it matter?. *Structural Safety*. Vol. 31 (2). 2009. P. 105–112.

[57] Domingos P., Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*. 1997. P. 103–130.

[58] Dournaee B., Birkel S. Improving Performance of Healthcare Systems with Service Oriented Architecture. URL: https://www.infoq.com/articles/soa-healthcare.

[59] Farman Ali, Shaker El-Sappagh, S.M. Riazul Islam, Daehan Kwak, Amjad Ali, Muhammad Imran, and Kyung-Sup Kwak. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion*. vol. 63. 2020. P. 208-222. URL: https://doi.org/10.1016/j.inffus.2020.06.008.

[60] Farnia F., Tse D. A minimax approach to supervised learning. 30th International Conference on Neural Information Processing Systems. 2016. P. 4240–4248. ISBN: 9781510838819.

[61] Feng S., Zhuo Z., Pan D. CcNet: A cross-connected convolutional network for segmenting retinal vessels using multi-scale features. *Neurocomputing*. Vol. 392. 2020. P. 268–2767.

[62] Forys U. Marchuk's model of immune system dynamics with application to tumour growth. *Journal of Theoretical Medicine*. Vol. 4. No. 1. P. 85–93. 2002. URL: http:// www.tandfonline.com/doi/abs/10.1080/10273660290052151.

[63] Fraser-Bell S., Symes R., Vaze A. Hypertensive eye disease: A review. *Clinical & Experimental Ophthalmology*. Vol. 45. No. 1. 2017. P. 45–53.

[64] Freund Y., Schapire R. A decision-theoretic generalization of on-line learning and an application to boosting. In: Proceedings of the 2nd European Conference on Computational Leaming Theory. Barcelona, Spain. 1995. P. 23-37.

[65] Friedman J. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*. Vol. 29. No.5. 2001. P. 1189–1232. URL: http://dx.doi.org/10.1214/aos/1013203451.

[66] Fritz F. Success criteria for electronic medical record implementations in low-resource settings: a systematic review. *Journal of the American Medical Informatics Association*. Vol. 22. 2015. P. 479–488.

[67] Gao X., Cai Y., Qiu C. Retinal blood vessel segmentation based on the Gaussian matched filter and U-net. CISP-BMEI. 2017. URL: https://ieeexplore.ieee.org/abstract/document/8302199.

[68] Ghafarian H., Yazdi H. Functional gradient approach to probabilistic minimax active learning. *Engineering Applications of Artificial Intelligence*. Vol. 85. 2019. P. 21–32.

[69] Google App Engine (GAE). URL: https://www.tadviser.ru/index.php/Продукт:Google_App_Engine_(GAE).

[70] Google App Engine. URL: https://ru.wikipedia.org/wiki/Google_App_Engine.

[71] Gray R., Neuhoff D. Quantization. *IEEE Transactions on Information Theory*. Vol. 44. No. 6. 1998. P. 2325–2384.

[72] Guo S., Wang K., Kang H., Zhang Y., Gao Y., Li T. BTSDSN: Deeply supervised neural network with short connections for retinal vessel

segmentation. *International Journal of Medical Informatics*. Vol. 126. 2019. P. 105–113.

[73] Hand D., Yu K. Idiot's Bayes — not so stupid after all?. *International Statistical Review*. Vol. 69. 2001. P. 385–398.

[74] Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning Data, second ed. Springer. 2008.

[75] Ho T. The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol.8. 1998. P. 832-844.

[76] Huang K., Yang H., King I., Lyu M. Imbalanced learning with a biased minimax probability machine. IEEE Trans. Syst. Man Cybern. Part B. Vol. 36. No.4 2006. P.913–923.

[77] Huang K., Yang H., King I., Lyu M., Chan L. The minimum error minimax probability machine. *Journal of Machine Learning Research*. Vol. 5. No. 4. 2004. P. 1253–1286.

[78] Huang K., Yang H., King I., Lyu M., Chan L. The minimum error minimax probability machine. *Journal of Machine Learning Research*. Vol. 5. 2004. P. 1253–1286.

[79] Hunt E., Marin J., Stone P. Experiments in induction. Academic Press. New York. 1966.

[80] Jackson B. Google Cloud vs AWS in 2018 (Comparing the Giants). URL: https://kinsta.com/blog/google-cloud-vs-aws/.

[81] Jain A., Dubes R. Algorithms for clustering data. Prentice-Hall. Englewood Cliffs. 1988.

[82] Jian Tao. Introduction to Scientific Machine Learning. URL: https://hprc.tamu.edu/files/training/2021/Spring/Scientific_Machine_Learning_2021_spring.pdf.

[83] Jin Q., Meng Z., Pham T., Chen Q., Wei L., Su R. DUNet: A deformable network for retinal vessel segmentation. *Knowledge-Based Systems*. 2019. Vol. 178. P. 149–162.

[84] Juan J. Ludmila I. Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 28. No. 10. P. 1619-1630.

[85] Karn P., Biswal B., Samantaray S. Robust retinal blood vessel segmentation using hybrid active contour model. IET Image Process. Vol. 13. No. 3. 2019. P. 440–450.

[86] Khadiev K., Mannapov I., Safina L. The Quantum Version Of Classification Decision Tree Constructing Algorithm C5.0. *CEUR Workshop Proceedings*. 2019. Vol.2500. URL: http://ceur-ws.org/Vol-2842/paper_10.pdf.

[87] Khan Academy Labs. Modern information theory: Information Entropy. URL: https://www.khanacademy.org/computing/computer-science/informationtheory/moderninfotheory/v/information-entropy.

[88] Krizhevsky A., Sutskever I., Hinton G. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012. P. 1106–1114.

[89] Kuhn M., K. Johnson. Applied predictive modeling. Springer. 2013.

[90] Lanckriet G, Ghaoui L., Bhattacharyya C., Jordan M. Minimax probability machine. Neural Information Processing Systems (NIPS). 2001. P. 801–807.

[91] Lanckriet G., Ghaoui L., Bhattacharyya C., Jordan M. A robust minimax approach to classification. Journal of Machine Learning Research. Vol. 3. No.3. P. 555-582, 2003.

[92] LeCun Y., Boser B., Denker J., Henderson D., Howard R., Hubbard W., Jackel L. Backpropagation applied to handwritten zip code recognition. *Neural Computation*. Vol.1. No. 4. 1989. P. 541–551.

[93] Lee H., Lee M., Chung H., Kim H. Quantification of retinal vessel tortuosity in diabetic retinopathy using optical coherence tomography angiography. *Retina*. Vol. 38. No. 5. 2018. P. 976–985.

[94] Lewis J., Fowler M. Microservices. URL: https://martinfowler.com/articles/microservices.html.

[95] List of open-source health software Electronic health or medical record. URL: https://en.wikipedia.org/wiki/List_of_open-_source_health_softwareElectronic_health_or_medical_record.

[96] Liu W., Zhang L., Tao D., Cheng J. Support vector machine active learning by hessian regularization. *Journal of Visual Communication and Image Representation*. Vol. 49. 2017. P. 47–56.

[97] Lloyd S. Least squares quantization in PCM. *IEEE Trans Inform Theory (Special Issue on Quantization)*. Vol. IT-28. P. 129–137. March 1982.

[98] Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation. Proc. CVPR. 2015. P. 3431–3440. URL: https://www.cvfoundation.org/openaccess/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html.

[99] Lu J., Xu Y., Chen M., Luo Y. A Coarse-to-Fine fully convolutional neural network for fundus vessel segmentation. *Symmetry*. Vol. 10. No. 11. 2018. P. 607.

[100] Lughofer E. Single-pass active learning with conflict and ignorance. Evolving Systems. Vol. 3. No. 4. 2012. P. 251–271.

[101] Lughofer E., Buchtala O. Reliable all-pairs evolving fuzzy classifiers. IEEE Trans. Fuzzy Syst. Vol. 21. No. 4. 2013. P. 625–641.

[102] Luppa P., Sokoll L., Chan D. Immunosensors- principles and applications to clinical chemistry. *Clinica Chimica Acta*. Vol. 314. No. 1. 2001. P. 1–26. 2001. ISSN: 0009-8981. URL: http://www.sciencedirect.com/science/article/pii/S0009898101006295.

[103] Lyapandra A., Martsenyuk V., Igvozdetska. S., Szklarczyk R. Qualitative analysis of compartmental dynamic system using decision-tree induction. IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Vol. 2. 2015. P. 688–692.

[104] Ma J., Shen J. A novel twin minimax probability machine for classification and regression. *Knowledge-Based Systems*. Vol. 196. P. 105703. 2020. URL: https://doi.org/10.1016/j.knosys.2020.105703.

[105] Ma J., Yang L., Wen Y., Sun Q. Twin minimax probability extreme learning machine for pattern recognition. *Knowledge-Based Systems*. Vol. 187. 2020. URL: https://doi.org/10.1016/j.knosys.2019.06.014.

[106] Maldonado S., Bravo C., Pérez J., López J. Integrated framework for profitbased feature selection and svm classification in credit scoring. Decision Support Systems. Vol. 104. 2017. P. 113–121.

[107] Mallapragada P., Jin R., Jain A., Liu Y. SemiBoost: Boosting for semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 31. No.11. 2009. P. 2000–2014.

[108] Marchuk G., Petrov R., Romanyukha A., Bocharov G. Mathematical model of antiviral immune response. i. data analysis, generalized picture construction and parameters evaluation for hepatitis b. *Journal of Theoretical Biology*. Vol. 151. No. 1. 1991. P. 1–40. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0025819779&doi= 10.1016%2fS0022-5193%2805%2980142-0&partnerID=40& md5=f850637085913dc18f8e52c5b3f28600.

[109] Martsenyuk V. On code refactoring for decision making component combined with the open-source medical information system. Advances in Soft and Hard Computing AISC 889. 2019. URL: https://doi.org/10.1007/978-3-030-03314-9.

[110] Martsenyuk V. P., Andrushchak I. Y. Програмне середовище підтримки системних фармакокінетичних досліджень: підхід на основі Web-технологій. Штучний інтелект. No. 3. 2009.

[111] Martsenyuk V. P., Andrushchak I. Y. Розробка клінічної експертної системи, що грунтується на правилах, методом послідовного покриття. Наукові праці. Комп’ютерні технології. Т. 237. Вип. 225. 2014. С. 5–10.

[112] Martsenyuk V. P., Andrushchak I. Y., Gandzyuk N. M., Klymuk N. Y., Kuchvara O. M., Mayhruk Z. V. Decision Support System for Medical System Research. XX International Conference PDMU-2012 Problems of Decision Making under Uncertainties. Brno: Publishing office of the University of Defence. 2012. P. 123–128.

[113] Martsenyuk V. P., Martsenyuk O. M., Andrushchak I. Y. Mathematical tools for decision support system of medical system research under uncertainties. XXIV International Conference Problems of Decision Making under Uncertainties (PDMU-2014). Abstracts. September 1-5. Chesky Rudolets. Chech Republic. 2014. P. 11–13.

[114] Martsenyuk V. P., Semenets A. V., Sverstyuk A. S., Kovalchuk O. Y., Kravets N. O. Про інформаційну модель інтелектуальної медичної бази даних. Збірка тез доповідей учасників Міжнародної науково-практичної конференції студентів, аспірантів та молодих вчених Комп’ютери. Програми. Інтернет. Київ. 2003. С. 46.

[115] Martsenyuk V., Andrushchak I., Kuchvara O. UML-modeling of Decision Support System for Medical Research. *Medical Informatics and Engineerin*. No. 2. 2015. P. 27–34.

[116] Martsenyuk V., Babinets L., Dronyak Y., Paslay O., Veselska O., Warwas K., Andrushchak I., Klos-Witkowska A. On development of machine learning models with aim of medical differential diagnostics of the comorbid states. 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IEEE. Sep. 2019. URL: https://doi.org/10.1109/idaacs.2019.8924345.

[117] Martsenyuk V., Klos-Witkowska A., Sverstiuk A. Stability, bifurcation and transition to chaos in a model of immunosensor based on lattice differential equations with delay. *Electronic Journal of Qualitative Theory of Differential Equations:* No. 2018 (27). P. 1–31.

[118] Martsenyuk V., Kłos-Witkowska A., Sverstiuk A. Stability, bifurcation and transition to chaos in a model of immunosensor based on lattice differential equations with delay. *Electronic Journal of Qualitative Theory of Differential Equations*. No. 27. 2018. P. 1–31. URL: https://doi.org/10.14232/ejqtde.2018.1.27.

[119] Martsenyuk V., Mayhruk Z., Karpinski M., Milian N., Andrushchak I., Veselska O. On implementation of decision tree induction in cloud platforms. *2019 Advances in Science and Engineering Technology International Conferences (ASET*), Dubai, United Arab Emirates. 2019. P. 1-6. doi: 10.1109/ICASET.2019.8714450. ISBN: 978-1-5386-8271-5. (Індексується в Scopus).

[120] Martsenyuk V., Milian N., Milian R. The minimax approach application for segmentation of retinal vessels. *Innovative Solutions in Modern Science*. New York. TK Meganom LLC. Vol. 7(43). 2020. P. 207-217. ISSN 2414-634X (Індексується в Google Scholar).

[121] Martsenyuk V., Povoroznyuk V., Semenets A., Martynyuk L. On an approach of the solution of machine learning problems integrated with data from the open-source system of electronic medical records: Application for fractures prediction. *Artificial Intelligence and Soft Computing*. Springer International Publishing. 2019. P. 228-239. URL: https://doi.org/10.1007/978-3-030-20915-5_21.

[122] Martsenyuk V., Sverstiuk A. Cyber-physical model of the immunosensor system at the hexagonal lattice with the use of difference equations of the population dynamics. *Innovative Technologies and Scientific Solutions for Industries*. 2019. № 1. P. 75–83.

[123] Martsenyuk V., Vakulenko D., Vakulenko L., Kos-Witkowska A. Information system of arterial oscillography for primary diagnostics of cardiovascu- lar diseases. Computer Information Systems and Industrial Management. CISIM 2018. Lecture Notes in Computer Science. 2018. P. 46–56.

[124] Martsenyuk V.P., Sverstiuk A.S., Klos-Witkowska A., Bagriy-Zayats O.A. Numerical Simulation of Cyber-physical Biosensor Systems on the Basis of Lattice Difference Eqations. *Advances in Cyber-physical Systems*. 2019. Vol. 4. No. 2. P. 55–70.

[125] Marzeniuk V. Taking into account delay in the problem of immune protection of organism. *Nonlinear Analysis: Real World Applications*. Vol. 2. No. 4. 2001. P. 483–496.

[126] Matthies, Hermann G. Quantifying Uncertainty: Modern Computational Representation of Probability and Applications. *Extreme Man-Made and Natural Hazards in Dynamics of Structures.* NATO Security through Science Series. 2007. P. 105–135. ISBN 978-1-4020-5654-3.

[127] Mease D., Wyner A., Buja A. Boosted classification trees and class probability/quantile estimation. *Journal of Machine Learning Research*. Vol. 8. 2007. P. 409–439.

[128] Michálek J., Nakonechny O. Minimax estimates of a linear parameter function in a regression model under restrictions on the parameters and variance-covariance matrix. *Journal of Mathematical Sciences*. Vol. 102. No. 1. P. 3790–3802. Oct. 2000. URL: https://doi.org/10.1007/bf02680236.

[129] Microsoft Azure vs. Amazon Web Services: Cloud Comparison. URL: https://www.business.com/articles/azure-vs-aws-cloud-comparison/.

[130] Milyan N., Andrushchak I., Sverstiuk A, Martsenyuk V. Microservices Architecture of One Solution for Emergency. *Przetwarzanie, transmisja i bezpieczeństwo informacji*. Bielsko – Biała. Tom 2. 2019. P. 15-25. ISBN: 978-83-66249-23-3, ISBN: 978-83-66249-25-7 (Tom 2) (Індексується в Google Scholar).

[131] Milyan N., Martsenyuk V. On application of cloud platforms for implementation of decision-making algorithms. *Przetwarzanie, transmisja i bezpieczeństwo informacji*. Bielsko – Biała. Tom 2. 2018. P. 143-153. ISBN: 978-83-65182-92-0, ISBN: 978-83-65182-94-4 (Tom 2) (Індексується в Google Scholar).

[132] Mitchell T. Machine Learning. McGraw-Hill series in computer science. New York : McGraw-Hill. 1997. P. 414.

[133] Mitra E. D., Hlavacek W. S. Parameter estimation and uncertainty quantification for systems biology models. *Current Opinion in Systems Biology*. Vol. 18. P. 9-18. Dec. 2019. URL: https://doi.org/10.1016/j.coisb.2019.10.006.

[134] Moccia S., De Momi E., El Hadji S., Mattos L. Blood vessel segmentation algorithms—Review of methods, datasets and evaluation metrics. *Computer Methods and Programs in Biomedicine*. Vol. 158. 2018. P. 71–91.

[135] Multiscale platform for coordinating cellular activity using synthetic biology. US9500636B2. A.Prindle, P. Samayoa, J. Hasty. Appl. No. PCT/US2012/069914, filed on Dec. 14, 2012. URL: https://patents.google.com/patent/US9500636B2/en.

[136] Nakonechny A. G., Marzeniuk V. P. Uncertainties in Medical Processes Control. Lecture Notes in Economics and Mathematical Systems. Vol. 581. P. 185-192.

[137] Nakonechny A., Marzeniuk V. Uncertainties in medical processes control. Lecture Notes in Economics and Mathematical Systems. Vol. 581. 2006. P. 185–192. URL: https://www. scopus.com/inward/record.uri?eid=2-s2.0-53749093113& doi=10.1007%2f3-540-35262-711&partnerID=40&md5= 03be7ef103cbbc1e94cacbb471daa03f.

[138] Nakonechnyi A. G., Kachinskiy A. B. Minimax parameter estimators of a linear regression with multiplicative noises. *Journal of Automation and Information Sciences*. Vol. 29. No. 2-3. P. 98–104. 1997. URL: https://doi.org/10.1615/jautomatinfscien.v29.i2-3.130.

[139] Nath S., Bhattacharyya C. Maximum margin classifiers with specified false positive and false negative error rates. Proceedings of the SIAM International Conference on Data Mining. 2007.

[140] Noymanee J., Theeramunkong T. Flood forecasting with machinelearning technique on hydrological modeling. *Procedia Computer*

Science. Vol. 156. 2019. P. 377-386. URL: http://www.sciencedirect.com/science/article/pii/S1877050919311330.

[141] Oliveira A., Pereira S., Silva C. Retinal vessel segmentation based on fully convolutional neural networks. Expert Systems with Applications. Vol. 112. 2018. P. 229–242.

[142] Pattern: Monolithic Architecture. URL: http://microservices.io/patterns/monolithic.html.

[143] Pawlak Z. Rough sets. *International Journal of Information and Computer Sciences*. 1982. Vol. 11. P. 341–356.

[144] Peter B., Yu B., Bühlmann P. Sparse boosting. *Journal of Machine Learning Research*. Vol. 7, 2006. P. 1001–1024. URL: http://dx.doi.org/10.1109/ICASSP.2009.4959911.

[145] Peter M. The NIST Definition of Cloud Computing URL: https://csrc.nist.gov/publications/detail/sp/800-145/final.

[146] Pirracchio R., Cohen M., Malenica I., Cohen J., Chambaz A., Cannesson M., Lee C., Resche-Rigon M., Hubbard A. Big data and targeted machine learning in action to assist medical decision in the icu. *Anaesthesia Critical Care Pain Medicine*. Vol. 38. No. 4. P. 377-384. 2019. URL: http://www.sciencedirect.com/science/article/pii/S23525568183.

[147] Platzer A. Differential dynamic logic for hybrid systems. *Journal of Automated Reasoning*. Vol. 41. No. 2. 2008. P. 143–189. ISSN: 0168-7433.

[148] Primary Care Information Project (PCIP). URL: https://www.pcpcc.org/initiative/primary-care-information-project-pcip.

[149] Prindle A, Samayoa P, Razinkov I, Danino T, Tsimring LS, Hasty J. A sensing array of radically coupled genetic 'biopixels'. *Nature*. 2011.N Vol. 481(7379). P. 39-44.

[150] Qayyum A., Qadir J., Bilal M., Al-Fuqaha A. Secure and Robust Machine Learning for Healthcare: A Survey, in IEEE Reviews in Biomedical Engineering. DOI: 10.1109/RBME.2020.3013489.

[151] Quinlan J. C4.5: Programs for machine learning. Morgan Kaufmann Publishers, San Mateo.1993.

[152] Quinlan J. Discovering rules by induction from large collections of examples. Expert systems in the microelectronic age. Edinburgh University Press. 1979.

[153] Rackauckas C., Mac Y., Martensend J., Warnera C., Zubove K., Supekara R., Skinnera D., Ramadhana A., Edelmana A. Universal Differential Equations for Scientific Machine Learning. *arXiv preprint arXiv:2001.04385*. 2020. URL: https://assets.researchsquare.com/files/rs-55125/v1_stamped.pdf?c=1598908557.

[154] Rackspace Cloud. URL: https://www.rackspace.com/cloud.

[155] Resampling. URL: https://mlr.mlr-org.com/articles/tutorial/resample.html.

[156] Reynolds C., Wyatt J. Open source, open standards, and health care information systems. *Journal of medical Internet research*. 2011.

[157] Ronneberger O., Fischer P., Brox T. U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention. (MICCAI). 2015. URL: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.

[158] Rouse M. Google Cloud Platform (GCP). URL: http://searchcloudcomputing.techtarget .com/definition/Google-Cloud-Platform.

[159] Rouse M. Microsoft Azure Machine Learning. URL: https://searchcloudcomputing.techtarget.com/definition/Microsoft-Azure-Machine-Learning.

[160] Sackett D., Rosenberg W., Gray J., Haynes R., Richardson W. Evidence based medicine: what it is and what it isn't. *BMJ*. Vol. 312. No. 7023. 1996. P. 71–72.

[161] Schapire R., Freund Y. Boosting: Foundations and Algorithms. MIT Press. 2012. P. 544. URL: http://dx.doi.org/10.1017/CBO9781107415324.004.

[162] Semenets A. About experience of the patient data migration during the open source emr-system implementation. *Medical Informatics and Engineering*. Vol. 1. 2015.

[163] Semenets A. On organizational and methodological approaches of the emr-systems implementation in public health of Ukraine. *Medical Informatics and Engineering*. Vol. 3. 2013.

[164] Semenets A. V., Martsenyuk V. P. On the CDSS platform development for the open-source MIS OpenEMR. *Medical Informatics and Engineering*. Oct. 2015. No. 3. P. 22–40.

[165] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. arXiv:1409.1556.

[166] Singh N., Srivastava R. Retinal blood vessels segmentation by using Gumbel probability distribution function based matched filter. *Comput. Computer Methods and Programs in Biomedicine*. Vol. 129. 2016. P. 45–50.

[167] Sotiris K. Combining bagging, boosting, rotation forest and random subspace methods. *Aritificial Intelligence Review*. Vol. 35. 2011. P. 223-240.

[168] Spasic I., Nenadic G. Clinical Text Data in Machine Learning: Systematic Review. JMIR Med Inform. vol.8(3):e17984, 2020. URL: https://medinform.jmir.org/2020/3/e17984.

[169] Srinidhi C., Aparna P., Rajan J. Recent advancements in retinal vessel segmentation. *Journal of Medical Systems*. Vol. 41. No. 4. 2017.

[170] Steinwart I., Scovel C. Fast rates for support vector machines using Gaussian kernels. *The Annals of Statistics*. Vol. 35. No. 2. 2007. P. 575–607. URL: http://dx.doi.org/10.1214/ 009053606000001226.

[171] Strohmann T., Grudic G. A formulation for minimax probability machine regression. Neural Information Processing Systems (NIPS). 2002. P. 769–776.

[172] Tao D., Jin L., Liu W., Li X. Hessian regularized support vector machines for mobile image annotation on the cloud. IEEE, Trans. Multimed. Vol. 15. No.4. 2013 P.833–844.

[173] Telgarsky M., Dudík M., Schapire R. Convex risk minimization and conditional probability estimation. Workshop and Conference Proceedings. Vol. 40. 2015. P. 1–54.

[174] Tewari A., Bartlett P. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*. Vol. 8. 2007. P. 1007–1025. URL: http://dx.doi.org/10.1007/11503415_10.

[175] Triguero I., García S., Herrera F. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*. Vol. 42 No. 2. 2015. P. 245–284. URL: http://dx.doi.org/10.1007/s10115-013-0706-y.

[176] Tuba E., Mrkela L., Tuba M. Retinal blood vessel segmentation by support vector machine classification. 27th Int. Conf. Radioelektronika (RADIOELEKTRONIKA). 2017. URL: https://ieeexplore.ieee.org/abstract/document/7936649.

[177] Types of Cloud Computing. URL: ttps://aws.amazon.com/types-of-cloud-computing/.

[178] Uslu F., Bharath A. A recursive Bayesian approach to describe retinal vasculature geometry. *Pattern Recognit*. Vol. 87. 2019. P. 157–169.

[179] Valiant L. A theory of learnable. *Artificial Intelligence and Language Processing*. Vol. 27. No. 11. 1984. P. 1134-1142.

[180] Vapnik V. Statistical Learning Theory. John Wiley and Sons. 1998. ISBN: 978-0-471-03003-4. P. 768.

[181] Vapnik V. The nature of statistical learning theory. Springer. New York. 1995. ISBN: 0-387-98780-0.

[182] Wąchocki G. Zastosowanie SOA do celów konstrukcji systemu wspomagającego ratownictwo medyczne. *Automatyka*. Vol.13. No. 2 2009. P. 653-661.

[183] Wang A., An N., Chen G., Liu L., Alterovitz G. Subtype dependent biomarker identification and tumor classification from gene expression profiles, *Knowledge-Based Systems*. Vol. 146. 2018. P. 104–117.

[184] Wang C., Zhao Z., Ren Q., Xu Y., Yu Y. Dense U-net based on patchbased learning for retinal vessel segmentation. *Entropy*. Vol. 21. No. 2. 2019. P. 168.

[185] Wang W., Zhang J., Wu W., Zhou S. An automatic approach for retinal vessel segmentation by multi-scale morphology and seed point tracking. *Journal of Medical Imaging and Health Informatics*. Vol. 8. No. 2. 2018. P. 262–274.

[186] What is the time complexity for training a neural network using back-propagation? URL: https://ai.stackexchange.com/questions/5728/what-is-the-time-complexity-for-training-a-neural-network-using-back-propagation.

[187] What is the time complexity of K-fold cross-validation? URL: https://stats.stackexchange.com/questions/450322/what-is-the-time-complexity-of-k-fold-cross-validation.

[188] Wu S., Mortveit H. A general framework for experimental design, uncertainty quantification and sensitivity analysis of computer simulation models. 2015 Winter Simulation Conference (WSC), IEEE. Dec. 2015. URL: https://doi.org/10.1109/wsc.2015.7408240.

[189] Xiao X., Lian S., Luo Z. Weighted Res-UNet for highquality retina vessel segmentation. 9th Int. Conf. Inf. Technol. Med. Educ. (ITME). 2018. URL: https://ieeexplore.ieee.org/abstract/document/8589312.

[190] Xizhao W., Junhai Z. Learning with Uncertainty. CRC Press. 2016. P. 240.

[191] Yang L., Wen Y., Zhang M., Wang X. Twin minimax probability machine for pattern classification. *Neural Networks*. Vol. 13. P. 201-214. 2020. URL: https://doi.org/10.1016/j.neunet.2020.07.030.

[192] Yang L., Yang B., Jing S., Sun Q. A minimax probability extreme machine framework and its application in pattern recognition. *Engineering Applications of Artificial Intelligence*. Vol. 81. P. 260-269. 2019. URL: https://doi.org/10.1016/j.engappai.2019.02.012.

[193] Yang X. Practical stability in dynamical systems. *Chaos, Solitons & Fractals*. Vol. 11. No. 7. 2000. P. 1087–1092.

[194] Zhang C., Zhang J. A variant of Rotation Forest for constructing ensemble classifiers. *Pattern Analysis Application*. Vol. 13. 2010. P. 59-77.

[195] Zhang, T. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*. Vol. 32. No.1. 2004. P. 56–134. URL: http://dx.doi.org/10. 1214/aos/1079120130.

ДОДАТОК А

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ

Праці, в яких опубліковано основні наукові результати дисертації:

1. Мілян Н.В. Марценюк В. П. Approach to Solving Machine Learning Problems Integrated with Data of the Systems with Open Code of Electronic Medical Records. *Вісник ТНТУ*. Т. : ТНТУ, 2019. Том 95. № 3. С. 105–116. **ISSN 2522-4433** (Індексується в Index Copernicus, Google Scholar).

2. Martsenyuk V., Milian N., Milian R. The minimax approach application for segmentation of retinal vessels. New York. TK Meganom LLC. *Innovative Solutions in Modern Science*. 7(43). 2020. P. 207-217. **ISSN 2414-634X** (Індексується в Google Scholar).

3. Марценюк В.П., Андрущак І.Є., Мілян Н.В. Мінімаксний підхід в машинному навчанні: сучасний стан та перспективи розвитку. *"Комп'ютерно-інтегровані технології: освіта, наука, виробництво"Луцьк, 2021. Випуск № 42*, С. 164-171. **ISSN 2524-0552** (Індексується в Index Copernicus, Google Scholar).

Праці, які засвідчують апробацію матеріалів дисертації:

4. Martsenyuk V., Mayhruk Z., Karpinski M., Milian N., Andrushchak I., Veselska O., "On implementation of decision tree induction in cloud platforms" 2019 *Advances in Science and Engineering Technology International Conferences (ASET)*, Dubai, United Arab Emirates, 2019, P. 1-6, doi: 10.1109/ICASET.2019.8714450. ISBN: 978-1-5386-8271-5. (Індексується в Scopus).

5. Milyan N., Martsenyuk V. On application of cloud platforms for implementation of decision-making algorithms. *Przetwarzanie, transmisja i*

bezpieczeństwo informacji . Bielsko – Biała, 2018. Tom 2. P. 143-153. ISBN: 978-83-65182-92-0, ISBN: 978-83-65182-94-4 (Tom 2) (Індексується в Google Scholar).

6. Milyan N., Andrushchak I., Sverstiuk A, Martsenyuk V. Microservices Architecture of One Solution for Emergency. *Przetwarzanie, transmisja i bezpieczeństwo informacji* . Bielsko – Biała, 2019. Tom 2. P. 15-25. ISBN: 978-83-66249-23-3, ISBN: 978-83-66249-25-7 (Tom 2) (Індексується в Google Scholar).

7. Марценюк В., Мілян Н. Аналіз платформи Google Cloud Platform. *Матеріали V Міжнародної науково-технічної конф-ції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя*, (Тернопіль, 1 – 2 лютого 2018 р.). Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2018. С. 73 – 74.

8. Марценюк В., Мілян Н. Роль і місце хмарних обчислень серед сучасних інформаційних технологій. *Фундаментальні та прикладні проблеми сучасних технологій: матеріали Міжнародної науково-технічної конференції до 100-річчя з дня заснування НАН України та на вшанування пам'яті Івана Пулюя (100-річчя з дня смерті)*, (Тернопіль, 23–24 травня 2018 року). Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2018. С. 219 – 221. ISBN: 978-966-305-093-5.

9. Марценюк В., Мілян Н. Використання алгоритмів індукції дерева рішень для аналізу великих обсягів даних. *Актуальні задачі сучасних технологій : зб. тез доповідей міжнар. наук.-техн. конф. Молодих учених та студентів*, (Тернопіль, 28–29 листоп. 2018.). М-во освіти і науки України, Терн. націон. техн. ун-т ім. І. Пулюя [та ін]. Тернопіль : ТНТУ, 2018. С. 114 – 116.

10. Мілян Н. Аналіз методів машинного навчання з вчителем. *Збірник тез Міжнародної студентської науково-технічної конференції*

„Природничі та гуманітарні науки. Актуальні питання“, 26-27 квітня 2018 року. Т. : ТНТУ, 2018. Том 1. С. 51–52.

Праці, які додатково відображають наукові результати дисертації:

11. Спосіб кіберфізичного моделювання сенсорного масиву біопікселів. Патент України на корисну модель №139048 МПК G01N33/483 / Н.В. Мілян, В.П. Марценюк, А.С. Сверстюк, А.М. Клос-Вітковська. Заявка № u201901590. Заявл. 18.02.2019. Опубл. 26.12.2019. Бюл.№ 24.

ДОДАТОК Б

Фрагменти програмного коду реалізації мінімаксного підходу в задачі глибокого МН

Встановлення параметрів і шляху до даних:

```
from glob import glob
from tqdm import tqdm
import matplotlib.pyplot as plt
from IPython import display
import numpy as np
import cv2
import os
import random
import time
import shutil
from sklearn.utils import shuffle
%matplotlib inline

patch_size=48
patch_num=1000
patch_threshold=25
TRAIN_OR_VAL=0.7
dataset_path='/content/drive/My Drive/dataset/DRIVE/'

train_dir=dataset_path+"training/"
test_dir=dataset_path+"test/"

train_image_dir=train_dir+"images/"
train_mask_dir=train_dir+"mask/"
train_groundtruth_dir=train_dir+"1st_manual/"
train_patch_dir=train_dir+"patch/"

test_image_dir=test_dir+"images/"
test_mask_dir=test_dir+"mask/"
test_groundtruth_dir=test_dir+"groundtruth/"
test_save_dir=test_dir+"pred_result/"

train_image_path_list=glob(train_image_dir+"*.tif")
test_image_path_list=glob(test_image_dir+"*.tif")

val_image_path_list=random.sample(train_image_path_list,int(len(train_image_path_list)
*(1-TRAIN_OR_VAL)))
train_image_path_list=[i for i in train_image_path_list if i not in val_image_path_lis
t]

print("number of training images:",len(train_image_path_list))
print("number of valid images:",len(val_image_path_list))
print("number of testing images:",len(test_image_path_list))
```

Попередня обробка зображення:

```
def restrict_normalized(imgs,mask):
    imgs_normalized = np.empty(imgs.shape)
    imgs_std = np.std(imgs)
    imgs_mean = np.mean(imgs)
    imgs_normalized = (imgs-imgs_mean)/imgs_std
    for i in range(imgs.shape[2]):
        imgs_normalized[:,:,i] = ((imgs_normalized[:,:,i] -
 np.min(imgs_normalized[:,:,i])) / (np.max(imgs_normalized[:,:,i])-
np.min(imgs_normalized[:,:,i])))*255
    return imgs_normalized

def clahe_equalized(imgs):
  clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
  imgs_equalized = np.empty(imgs.shape)
  for i in range(imgs.shape[2]):
    imgs_equalized[:,:,i] = clahe.apply(np.array(imgs[:,:,i], dtype = np.uint8))
  return imgs_equalized

def normalized(imgs):
  imgs_normalized =np.empty(imgs.shape)
  for i in range(imgs.shape[2]):
    imgs_normalized[:,:,i] =cv2.equalizeHist(imgs[:,:,i])
  return imgs_normalized

def adjust_gamma(imgs, gamma=1.0):
  invGamma = 1.0 / gamma
  table = np.array([((i / 255.0) ** invGamma) * 255 for i in np.arange(0, 256)]).astyp
e("uint8")
  new_imgs = np.empty(imgs.shape)
  for i in range(imgs.shape[2]):
    new_imgs[:,:,i] = cv2.LUT(np.array(imgs[:,:,i], dtype = np.uint8), table)
  return new_imgs

def preprocess(image,mask):

  assert np.max(mask)==1
  image=np.array(image)
  image[:,:,0]=image[:,:,0]*mask
  image[:,:,1]=image[:,:,1]*mask
  image[:,:,2]=image[:,:,2]*mask

  image=restrict_normalized(image,mask)
  image=clahe_equalized(image)
  image=adjust_gamma(image,1.2)
  image=image/255.0
  return image
```

Створення патчів зображень/масок:

```
def check_coord(x,y,h,w,patch_size):
  if x-patch_size/2>0 and x+patch_size/2<h and y-patch_size/2>0 and y+patch_size/2<w:
    return True
  return False

def image2patch(image_path,patch_num,patch_size,training=True,show=True):
  image_name=image_path.split("/")[-1].split("_")[0]

  image=plt.imread(image_path)

  groundtruth=plt.imread(train_groundtruth_dir+image_name+"_manual1.gif")
  groundtruth=np.where(groundtruth>0,1,0)

  mask=plt.imread(train_mask_dir+image_name+"_training_mask.gif")
  mask=np.where(mask>0,1,0)

  image=preprocess(image,mask)
  image_show=image.copy()
  groundtruth_show=np.zeros_like(image)
  groundtruth_show[:,:,0]=groundtruth.copy()
  groundtruth_show[:,:,1]=groundtruth.copy()
  groundtruth_show[:,:,2]=groundtruth.copy()

  sample_count=0
  sample_index=0

  sample_point=np.where(groundtruth==1)

  state = np.random.get_state()
  np.random.shuffle(sample_point[0])
  np.random.set_state(state)
  np.random.shuffle(sample_point[1])

  patch_image_list=[]
  patch_groundtruth_list=[]

  while sample_count<patch_num and sample_index<len(sample_point[0]):
    x,y=sample_point[0][sample_index],sample_point[1][sample_index]
    if check_coord(x,y,image.shape[0],image.shape[1],patch_size):
      if np.sum(mask[x-patch_size//2:x+patch_size//2,y-
patch_size//2:y+patch_size//2])>patch_threshold:

        patch_image_binary=image[x-patch_size//2:x+patch_size//2,y-
patch_size//2:y+patch_size//2,:]
        patch_groundtruth=groundtruth[x-patch_size//2:x+patch_size//2,y-
patch_size//2:y+patch_size//2
        patch_groundtruth=np.where(patch_groundtruth>0,255,0)
```

```
        patch_image_list.append(patch_image_binary)
        patch_groundtruth_list.append(patch_groundtruth)
          cv2.rectangle(image_show, (y-patch_size//2,x-
patch_size//2,), (y+patch_size//2,x+patch_size//2), (0,1,0), 2)
          cv2.rectangle(groundtruth_show, (y-patch_size//2,x-
patch_size//2,), (y+patch_size//2,x+patch_size//2), (0,1,0), 2)
        sample_count+=1

    if show:
      plt.figure(figsize=(15,15))
      plt.title("processing: %s"%image_name)
      plt.subplot(121)
      plt.imshow(image_show,cmap=plt.cm.gray)
      plt.subplot(122)
      plt.imshow(groundtruth_show,cmap=plt.cm.gray)
      display.clear_output(wait=True)
    sample_index+=1

  for i in range(len(patch_image_list)):
    if training==True:
        plt.imsave(train_patch_dir+image_name+"-"+str(i)+"-
img.jpg",patch_image_list[i])

        plt.imsave(train_patch_dir+image_name+"-"+str(i)+"-
groundtruth.jpg",(patch_groundtruth_list[i]/225.0).astype(np.uint8),cmap = plt.cm.gray
)
    else:
        plt.imsave(train_patch_dir+image_name+"_"+str(i)+"_val_img.jpg",patch_image_li
st[i])

        plt.imsave(train_patch_dir+image_name+"_"+str(i)+"_val_groundtruth.jpg",(patch
_groundtruth_list[i]/225.0).astype(np.uint8),cmap = plt.cm.gray)


if not os.path.exists(train_patch_dir):
  os.mkdir(train_patch_dir)
else:
  shutil.rmtree(train_patch_dir)
  os.mkdir(train_patch_dir)

if not os.path.exists(test_save_dir):
  os.mkdir(test_save_dir)

for i in tqdm(range(len(train_image_path_list)),desc="Generate the training patches: "
):
  image2patch(train_image_path_list[i],patch_num,patch_size,training=True,show=False)


for i in tqdm(range(len(val_image_path_list)),desc="Generate the val patches: "):
```

```
  image2patch(val_image_path_list[i],patch_num,patch_size,training=False,show=False)
```

Визначення моделі:

```
import tensorflow as tf
from tensorflow.keras import backend as K
from tensorflow.keras.layers import AveragePooling2D,Conv2DTranspose,Input,Add,Conv2D,
 BatchNormalization,LeakyReLU, Activation, MaxPool2D, Dropout, Flatten, Dense,UpSampli
ng2D,Concatenate,Softmax


class LinearTransform(tf.keras.Model):
  def __init__(self, name="LinearTransform"):
    super(LinearTransform, self).__init__(self,name=name)

    self.conv_r=Conv2D(1,kernel_size=3,strides=1,padding='same',use_bias=False)
    self.conv_g=Conv2D(1,kernel_size=3,strides=1,padding='same',use_bias=False)
    self.conv_b=Conv2D(1,kernel_size=3,strides=1,padding='same',use_bias=False)

    self.pool_rc=AveragePooling2D(pool_size=(patch_size,patch_size),strides=1)
    self.pool_gc=AveragePooling2D(pool_size=(patch_size,patch_size),strides=1)
    self.pool_bc=AveragePooling2D(pool_size=(patch_size,patch_size),strides=1)

    self.bn=BatchNormalization()
    self.sigmoid=Activation('sigmoid')
    self.softmax=Activation('softmax')

  def call(self, input,training=True):
    r,g,b=input[:,:,:,0:1],input[:,:,:,1:2],input[:,:,:,2:3]

    rs=self.conv_r(r)
    gs=self.conv_g(g)
    bs=self.conv_r(b)

    rc=tf.reshape(self.pool_rc(rs),[-1,1])
    gc=tf.reshape(self.pool_gc(gs),[-1,1])
    bc=tf.reshape(self.pool_bc(bs),[-1,1])

    merge=Concatenate(axis=-1)([rc,gc,bc])
    merge=tf.expand_dims(merge,axis=1)
    merge=tf.expand_dims(merge,axis=1)
    merge=self.softmax(merge)
    merge=tf.repeat(merge,repeats=48,axis=2)
    merge=tf.repeat(merge,repeats=48,axis=1)

    r=r*(1+self.sigmoid(rs))
    g=g*(1+self.sigmoid(gs))
    b=b*(1+self.sigmoid(bs))

    output=self.bn(merge[:,:,:,0:1]*r+merge[:,:,:,1:2]*g+merge[:,:,:,2:3]*b,training=t
raining)
    return output
```

```
class ResBlock(tf.keras.Model):
  def __init__(self,out_ch,residual_path=False,stride=1):
    super(ResBlock,self).__init__(self)
    self.residual_path=residual_path

    self.conv1=Conv2D(out_ch,kernel_size=3,strides=stride,padding='same', use_bias=Fal
se,data_format="channels_last")
    self.bn1=BatchNormalization()
    self.relu1=LeakyReLU()

    self.conv2=Conv2D(out_ch,kernel_size=3,strides=1,padding='same', use_bias=False,da
ta_format="channels_last")
    self.bn2=BatchNormalization()

    if residual_path:
      self.conv_shortcut=Conv2D(out_ch,kernel_size=1,strides=stride,padding='same',use
_bias=False)
      self.bn_shortcut=BatchNormalization()

    self.relu2=LeakyReLU()

  def call(self,x,training=True):
    xs=self.relu1(self.bn1(self.conv1(x),training=training))
    xs=self.bn2(self.conv2(xs),training=training)

    if self.residual_path:
      x=self.bn_shortcut(self.conv_shortcut(x),training=training)
    xs=x+xs
    return self.relu2(xs)

class Unet(tf.keras.Model):
  def __init__(self):
    super(Unet,self).__init__(self)
    self.conv_init=LinearTransform()
    self.resinit=ResBlock(16,residual_path=True)
    self.up_sample=UpSampling2D(size=(2,2),interpolation='bilinear')
    self.resup=ResBlock(32,residual_path=True)

    self.pool1=MaxPool2D(pool_size=(2,2))

    self.resblock_down1=ResBlock(64,residual_path=True)
    self.resblock_down11=ResBlock(64,residual_path=False)
    self.pool2=MaxPool2D(pool_size=(2,2))

    self.resblock_down2=ResBlock(128,residual_path=True)
    self.resblock_down21=ResBlock(128,residual_path=False)
    self.pool3=MaxPool2D(pool_size=(2,2))

    self.resblock_down3=ResBlock(256,residual_path=True)
    self.resblock_down31=ResBlock(256,residual_path=False)
```

```
    self.pool4=MaxPool2D(pool_size=(2,2))

    self.resblock=ResBlock(512,residual_path=True)

    self.unpool3=UpSampling2D(size=(2,2),interpolation='bilinear')
    self.resblock_up3=ResBlock(256,residual_path=True)
    self.resblock_up31=ResBlock(256,residual_path=False)

    self.unpool2=UpSampling2D(size=(2,2),interpolation='bilinear')
    self.resblock_up2=ResBlock(128,residual_path=True)
    self.resblock_up21=ResBlock(128,residual_path=False)

    self.unpool1=UpSampling2D(size=(2,2),interpolation='bilinear')
    self.resblock_up1=ResBlock(64,residual_path=True)

    self.unpool_final=UpSampling2D(size=(2,2),interpolation='bilinear')
    self.resblock2=ResBlock(32,residual_path=True)

    self.pool_final=MaxPool2D(pool_size=(2,2))
    self.resfinal=ResBlock(32)

    self.conv_final=Conv2D(1,kernel_size=1,strides=1,padding='same',use_bias=False)
    self.bn_final=BatchNormalization()
    self.act=Activation('sigmoid')

  def call(self,x,training=True):
    x_linear=self.conv_init(x,training=training)
    x=self.resinit(x_linear,training=training)
    x=self.up_sample(x)
    x=self.resup(x,training=training)

    stage1=self.pool1(x)
    stage1=self.resblock_down1(stage1,training=training)
    stage1=self.resblock_down11(stage1,training=training)

    stage2=self.pool2(stage1)
    stage2=self.resblock_down2(stage2,training=training)
    stage2=self.resblock_down21(stage2,training=training)

    stage3=self.pool3(stage2)
    stage3=self.resblock_down3(stage3,training=training)
    stage3=self.resblock_down31(stage3,training=training)

    stage4=self.pool4(stage3)
    stage4=self.resblock(stage4,training=training)

    stage3=Concatenate(axis=3)([stage3,self.unpool3(stage4)])
    stage3=self.resblock_up3(stage3,training=training)
    stage3=self.resblock_up31(stage3,training=training)

    stage2=Concatenate(axis=3)([stage2,self.unpool2(stage3)])
```

```
    stage2=self.resblock_up2(stage2,training=training)
    stage2=self.resblock_up21(stage2,training=training)

    stage1=Concatenate(axis=3)([stage1,self.unpool1(stage2)])
    stage1=self.resblock_up1(stage1,training=training)

    x=Concatenate(axis=3)([x,self.unpool_final(stage1)])
    x=self.resblock2(x,training=training)

    x=self.pool_final(x)
    x=self.resfinal(x,training=training)

    seg_result=self.act(self.bn_final(self.conv_final(x),training=training))

    return x_linear,seg_result
```

Навчання моделі:

```
EPOCHS=200
VAL_TIME=2
LR=0.0003
BATCH_SIZE=64

checkpoint_path=dataset_path+"ckpt/"
log_path=dataset_path+"logs/"

if not os.path.exists(checkpoint_path):
  os.mkdir(checkpoint_path)

if not os.path.exists(log_path):
  os.mkdir(log_path)
```

Навчання:

```
def load_image_groundtruth(img_path,groundtruth_path):
  img=tf.io.read_file(img_path)
  img=tf.image.decode_jpeg(img,channels=3)
  img=tf.image.resize(img,[patch_size,patch_size])

  groundtruth=tf.io.read_file(groundtruth_path)
  groundtruth=tf.image.decode_jpeg(groundtruth,channels=1)

  if random.uniform(0,1)>=0.5:
    img=tf.image.flip_left_right(img)
    groundtruth=tf.image.flip_left_right(groundtruth)

  img=tf.image.resize(img,[patch_size,patch_size])
  groundtruth=tf.image.resize(groundtruth,[patch_size,patch_size])

  img/=255.0
  groundtruth=(groundtruth+40)/255.0
```

```
  groundtruth=tf.cast(groundtruth,dtype=tf.uint8)

  return img,groundtruth

train_patch_img_path_list=sorted(glob(train_patch_dir+"*-*-img.jpg"))
train_patch_groundtruth_path_list=sorted(glob(train_patch_dir+"*-*-groundtruth.jpg"))
train_patch_img_path_list,train_patch_groundtruth_path_list=shuffle(train_patch_img_pa
th_list,train_patch_groundtruth_path_list,random_state=0)

print(len(train_patch_img_path_list),len(train_patch_groundtruth_path_list))
print(train_patch_img_path_list[:2])
print(train_patch_groundtruth_path_list[:2])

val_patch_img_path_list=sorted(glob(train_patch_dir+"*_*_val_img.jpg"))
val_patch_groundtruth_path_list=sorted(glob(train_patch_dir+"*_*_val_groundtruth.jpg")
)

print(val_patch_img_path_list[:2])
print(val_patch_groundtruth_path_list[:2])

train_dataset=tf.data.Dataset.from_tensor_slices((train_patch_img_path_list,train_patc
h_groundtruth_path_list))
train_dataset=train_dataset.map(load_image_groundtruth,num_parallel_calls=tf.data.expe
rimental.AUTOTUNE)
train_dataset = train_dataset.shuffle(buffer_size=1300).prefetch(BATCH_SIZE).batch(BAT
CH_SIZE)

val_dataset=tf.data.Dataset.from_tensor_slices((val_patch_img_path_list,val_patch_grou
ndtruth_path_list))
val_dataset=val_dataset.map(load_image_groundtruth,num_parallel_calls=tf.data.experime
ntal.AUTOTUNE)
val_dataset =val_dataset.shuffle(buffer_size=1300).prefetch(BATCH_SIZE).batch(BATCH_SI
ZE)
```

Завантаження і формування моделі:

```
model=Unet()

cosine_decay = tf.keras.experimental.CosineDecayRestarts(initial_learning_rate=LR, fir
st_decay_steps=12000,t_mul=1000,m_mul=0.5,alpha=1e-5)
optimizer=tf.keras.optimizers.Adam(learning_rate=cosine_decay)

loss=tf.keras.losses.BinaryCrossentropy(from_logits=False)

train_loss = tf.keras.metrics.Mean(name='train_loss')
train_acc=tf.keras.metrics.Mean(name='train_acc')
current_accuracy = tf.keras.metrics.BinaryAccuracy(name='train_accuracy')

val_loss = tf.keras.metrics.Mean(name='val_loss')
val_acc=tf.keras.metrics.Mean(name='val_acc')
```

```
val_accuracy = tf.keras.metrics.BinaryAccuracy(name='val_accuracy')

ckpt = tf.train.Checkpoint(model=model)


log_dir=log_path+ datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
log_writer = tf.summary.create_file_writer(log_dir)
```

Крок навчання і валідації:

```
def train_step(step,patch,groundtruth):
  with tf.GradientTape() as tape:

    linear,pred_seg=model(patch,training=True)
    losses = dice_loss(groundtruth, pred_seg)

  grads = tape.gradient(losses, model.trainable_variables)
  optimizer.apply_gradients(zip(grads, model.trainable_variables))

  train_loss.update_state(losses)
  train_acc.update_state(dice(groundtruth, pred_seg))

def val_step(step,patch,groundtruth):

  linear,pred_seg=model(patch,training=False)
  losses = dice_loss(groundtruth, pred_seg)

  val_loss.update_state(losses)
  val_acc.update_state(dice(groundtruth, pred_seg))

  tf.summary.image("image",patch,step=step)
  tf.summary.image("image transform",linear,step=step)
  tf.summary.image("groundtruth",groundtruth*255,step=step)
  tf.summary.image("pred",pred_seg,step=step)
  log_writer.flush()
```

Основна функція навчання:

```
lr_step=0
last_val_loss=2e10
with log_writer.as_default():
  for epoch in range(EPOCHS):
    train_loss.reset_states()
    train_acc.reset_states()
    val_loss.reset_states()
    val_acc.reset_states()

    for tstep, (patch,groundtruth) in enumerate(train_dataset):
      train_step(lr_step,patch,groundtruth)
```

```
      tf.summary.scalar("learning_rate", optimizer._decayed_lr(tf.float32).numpy(),
step=lr_step)
      print('\repoch {}, batch {}, loss:{:.4f}, dice:{:.4f}'.format(epoch + 1, tstep,
train_loss.result(), train_acc.result()),end="")
      lr_step+=1

    if (epoch + 1) % VAL_TIME == 0:
      for vstep, (patch,groundtruth) in enumerate(val_dataset):

        val_step(lr_step,patch,groundtruth)

      print('\repoch {}, batch {}, train_loss:{:.4f}, train_dice:{:.4f},
val_loss:{:.4f}, val_dice:{:.4f}'.format(epoch + 1, vstep, train_loss.result(),
train_acc.result(),val_loss.result(), val_acc.result()),end="")
      tf.summary.scalar("val_loss", val_loss.result(), step=epoch)
      tf.summary.scalar("val_acc", val_acc.result(), step=epoch)

      if val_loss.result()<last_val_loss:
        ckpt.save(checkpoint_path)
        last_val_loss=val_loss.result()
    print("")
    tf.summary.scalar("train_loss", train_loss.result(), step=epoch)
    tf.summary.scalar("train_acc", train_acc.result(), step=epoch)
    log_writer.flush()
```

Підготовка тестових зображень:

```
def padding_images(image,mask,stride):
    h,w=image.shape[:2]
    new_h,new_w=h,w
    while (new_h-patch_size)%stride!=0:
        new_h+=1
    while (new_w-patch_size)%stride!=0:
        new_w+=1
    pad_image=np.zeros((new_h,new_w,3))
    pad_image[:h,:w,:]=image

    pad_mask=np.zeros((new_h,new_w))
    pad_mask[:h,:w]=mask

    return pad_image,pad_mask

def img2patch_list(image,stride=patch_size):
    patch_list=[]
    #image_binary=0.8*image[:,:,1:2]+0.2*image[:,:,2:3]
    for j in range(0,image.shape[1]-patch_size+1,stride):
        for i in range(0,image.shape[0]-patch_size+1,stride):
            patch=image[i:i+patch_size,j:j+patch_size,:]
            patch_list.append(patch)
    return patch_list
```

```
def patchlist2image(patch_list,stride,image_shape):
    result=np.zeros(image_shape[:2])
    sum_matrix=np.zeros(image_shape[:2])
    index_x,index_y=0,0
    for i in range(patch_list.shape[0]):
        patch=patch_list[i,:,:,0]
        #patch=np.where(patch>0.5,1,0)
        #print(patch)
        result[index_x:index_x+patch_size,index_y:index_y+patch_size]+=patch
        sum_matrix[index_x:index_x+patch_size,index_y:index_y+patch_size]+=1
        index_x+=stride
        if index_x+patch_size>image_shape[0]:
            index_x=0
            index_y+=stride
    return result/sum_matrix
```

Функція тестування зображень:

```
testmodel=Unet()
ckpts = tf.train.Checkpoint(model=testmodel)
ckpts.restore(tf.train.latest_checkpoint(checkpoint_path)).expect_partial()

stride=5

def load_test_data(image):
  #image=tf.image.decode_jpeg(image,channels=1)
  #print(image.shape)
  image=tf.image.resize(image,[patch_size,patch_size])
  #image/=255.0
  return image

for i in range(len(test_image_path_list)):
  image_path=test_image_path_list[i]
  image_name=image_path.split("/")[-1].split("_")[0]

  image=plt.imread(image_path)
  original_shape=image.shape
  mask=plt.imread(test_mask_dir+image_name+"_test_mask.gif")
  mask=np.where(mask>0,1,0)

  image,pad_mask=padding_images(image,mask,stride)
  image=preprocess(image,pad_mask)
  test_patch_list=img2patch_list(image,stride)

  test_dataset=tf.data.Dataset.from_tensor_slices(test_patch_list)
  test_dataset=test_dataset.map(load_test_data)
  test_dataset=test_dataset.batch(64)
  pred_result=[]
```

```
  print("testing image:",int(image_name))
  for batch, patch in enumerate(test_dataset):
    _,pred=testmodel(patch,training=False)

    pred=pred.numpy()
    pred_result.append(pred)
  pred_result=np.concatenate(pred_result,axis=0)

  print("post processing:",image_name)
  pred_image=patchlist2image(pred_result,stride,image.shape)

  pred_image=pred_image[:original_shape[0],:original_shape[1]]

  kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (9, 9))
  print(mask.shape)
  mask = cv2.erode(mask.astype(np.uint8), kernel)

  pred_image=pred_image*mask
  pred_image=np.where(pred_image>0.5,1,0)

  plt.figure(figsize=(8,8))
  plt.title(image_name+"-("+str(image.shape[0])+","+str(image.shape[1])+")")
  plt.imshow(pred_image,cmap=plt.cm.gray)
  plt.show()

  plt.imsave(test_save_dir+str(int(image_name))+".png",pred_image,cmap = plt.cm.gray)
```

Оцінка:

```
test_groundtruth_path_list=sorted(glob(test_groundtruth_dir+"*.gif"))
test_pred_path_list=sorted(glob(test_save_dir+"*.png"))

dice_list=[]
roc_list=[]
pr_list=[]
tpr_list=[]
tnr_list=[]
sum_tp=0
sum_tn=0
sum_fp=0
sum_fn=0

roc = tf.keras.metrics.AUC(num_thresholds=200,curve="ROC")
pr = tf.keras.metrics.AUC(num_thresholds=200,curve="PR")
recall=tf.keras.metrics.Recall()


tn=tf.keras.metrics.TrueNegatives()
tp=tf.keras.metrics.TruePositives()
fn=tf.keras.metrics.FalseNegatives()
```

```
fp=tf.keras.metrics.FalsePositives()

for idx in range(len(test_groundtruth_path_list)):
    name=test_groundtruth_path_list[idx].split("/")[-1].split(".")[0].split("_")[0]
    roc.reset_states()
    pr.reset_states()
    tn.reset_states()
    fn.reset_states()
    tp.reset_states()
    fp.reset_states()

    groundtruth=plt.imread(test_groundtruth_path_list[idx])
    preds=plt.imread(test_save_dir+str(int(name))+".png")
    groundtruth=np.array(groundtruth,dtype=np.float32)
    groundtruth=groundtruth/255.0

    roc.update_state(groundtruth,preds[:,:,0])
    pr.update_state(groundtruth,preds[:,:,0])
    tn.update_state(groundtruth,preds[:,:,0])
    tp.update_state(groundtruth,preds[:,:,0])
    fn.update_state(groundtruth,preds[:,:,0])
    fp.update_state(groundtruth,preds[:,:,0])

    dice_list.append(dice(groundtruth,preds[:,:,0]).numpy())
    roc_list.append(roc.result().numpy())
    pr_list.append(pr.result().numpy())

    current_tn=tn.result().numpy()
    current_tp=tp.result().numpy()
    current_fn=fn.result().numpy()
    current_fp=fp.result().numpy()

    sum_tp+=current_tp
    sum_tn+=current_tn
    sum_fp+=current_fp
    sum_fn+=current_fn

    tpr_list.append(current_tp/(current_tp+current_fn))
    tnr_list.append(current_tn/(current_tn+current_fp))

print("average dice score for all predict vessel masks:",np.mean(dice_list))
print("average AUC for all predict vessel masks:",np.mean(roc_list))
print("average PR for all predict vessel masks:",np.mean(pr_list))
print("average recall(sensitivity) for all predict vessel masks:",np.mean(tpr_list))
print("average specificity for all predict vessel masks:",np.mean(tnr_list))
```

Матриця помилок:

```
predict = ["Positive","Negative"]
actual = [ "Positive","Negative"]
```

```
classes = list(set(actual))
classes.sort(reverse=True)
confusion_matrix=[[sum_tp,sum_fn],[sum_fp,sum_tn]]
print("confusion",confusion_matrix)

plt.figure(figsize=(8,8))
font_size=18
plt.imshow(confusion_matrix, cmap=plt.cm.Blues)
indices = range(len(confusion_matrix))
plt.xticks(indices, classes,rotation=40,fontsize=font_size)
plt.yticks([0.00,1.00], classes,fontsize=font_size)
plt.ylim(1.5 , -0.5)

plt.title("Confusion matrix (Pixel Level)",fontdict={'weight':'normal','size':
font_size})
plt.xlabel('Predict Label',fontsize=font_size)
plt.ylabel('Actual Label',fontsize=font_size)

for first_index in range(len(confusion_matrix)):
    for second_index in range(len(confusion_matrix[first_index])):
        if confusion_matrix[first_index][second_index]>2e6:
            text_color="w"
        else:
            text_color="black"
        plt.text(first_index, second_index,
confusion_matrix[first_index][second_index],fontsize=font_size, color =
text_color,verticalalignment='center',horizontalalignment='center',)

cb=plt.colorbar()
cb.ax.tick_params(labelsize=font_size)
plt.show()
```

ДОДАТОК В

Приклад реалізації наукового машинного навчання для класифікації кіберфічних біосенсорних систем

Приклад конкретної реалізації визначення оптимальних параметрів кіберфізичної біосенсорної системи полягає в розгляді моделі 4.2 при: $N = 32,\quad \beta = 2\min^{-1},\quad \gamma = 2\frac{mL}{\min\cdot\mu g},\quad \mu_f = 1\min^{-1},\quad \eta = 0.8/\gamma,\quad \delta_\upsilon = 0.5\frac{mL}{\min\cdot\mu g},$

$\delta_f = 0.5\frac{mL}{\min\cdot\mu g}\quad D = 0.2\frac{nm^2}{\min},\cdot\Delta = 0.3\,nm.$

Результати чисельного моделювання були реалізовані для різних значень $n \in [0,1]$. Наприклад, при $n = 0.9$, моделювання для $\tau > 0\,(\min)$ представлені на рис. 5.4, 5.5.

Результати чисельного моделювання, представлені на рис. 5.1, 5.2 були побудовані для $t \in [0, 100]$. Як видно з рис. 5.1 розв'язок сходиться до не ідентичного сталого стану, який є стійким фокусом. На рис. 5.2 розв'язок сходиться до стійкого граничного циклу з двома локальними екстремумами в циклі.

На рис. 5.1, 5.2 видно, що при зміні значення τ, маємо зміни якісної поведінки пікселів і усього біосенсора, аналізуючи вигляд електричного сигналу. Було розглянуто множину значень параметрів, які задані вище та проведено обчислення довготривалої поведінки моделі (4.2) для $\tau = 0.2$, $\tau = 0.28$. Фазові діаграми антитіл, відносно популяцій антигенів для пікселів довкола пікселя (15,15) для цих значень τ наведені на рис. 5.1, 5.2.

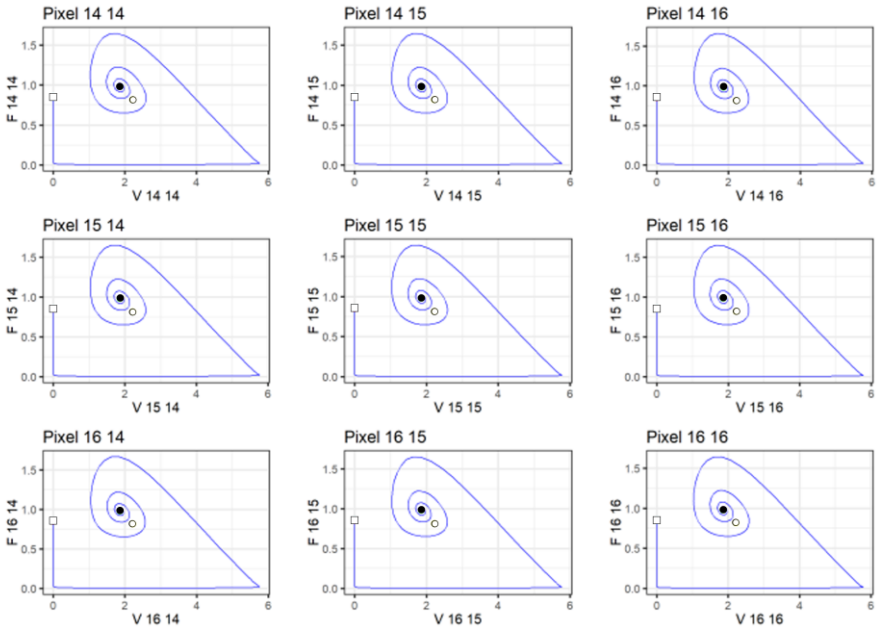


Рисунок 5.1 – Зображення фазових площин системи (4.2) для популяції

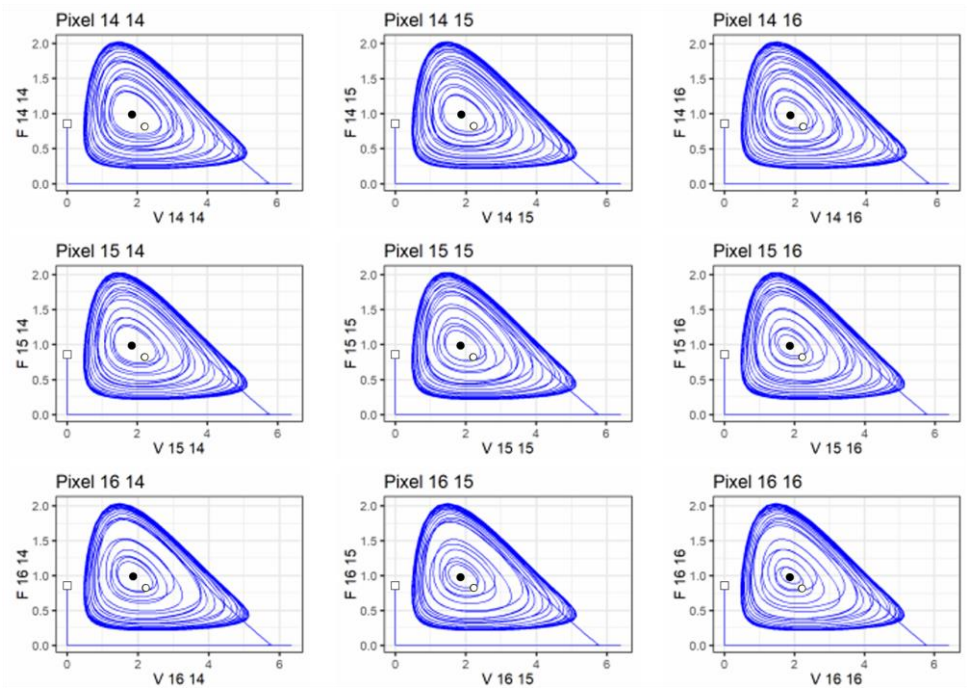


Рисунок 5.2 – Зображення фазових площин системи (4.2) для популяцій

На рис. 5.1, при $\tau \in [0, 0.22]$, видно траєкторії, які відповідають стійкому фокусу для всіх пікселів. Для значень τ біля 0,2223 хв. виникає біфуркація Хопфа [118] і наступні траєкторії відповідають стійким граничним циклам еліпсоїдальної форми для всіх пікселів. Для того, щоб

чисельні розв'язки, які відповідають біфуркації Хопфа, були узгодженими з теоретичними результатами, потрібно відшукати пару комплексно-спряжених уявних розв'язків відповідного характеристичного рівняння лінералізованої системи.

В роботі [118] модель (4.2) була досліджена з використанням фазових діаграм при $N = 4$. Для $\tau = 0.23$ фазові діаграми показують, що розв'язок є граничним циклом з двома локальними екстремумами (один локальний максимум та один локальний мінімум на цикл). Для $\tau = 0.2825$ розв'язок є граничним циклом з чотирьма локальними екстремумами на цикл, для $\tau = 0.2868$, 0.2869, 0.28695 і т.д., розв'язки є граничними циклами з 8, 16 і 32 локальними екстремумами на цикл відповідно. При $\tau = 0.28725$ спостерігаємо хаотичну поведінку при відсутності жодної періодичної поведінки на великому інтервалі часу. Для того, щоб перевірити, що розв'язок є хаотичним для $\tau = 0.28725$, початкові умови було збурено з метою тестування чутливості системи. Порівняння розв'язків для популяції антигенів $V_{1,3}$ із початковими умовами $V_{1,3}(t) = 1$, $V_{1,3}(t) = 1.001, t \in [\tau, 0]$ та усіма іншими ідентичними початковими умовами, які показують хаотичну поведінку. Біля початкового часу два розв'язки виглядають як однакові, але із збільшенням часу відзначається різниця між розв'язками, що підтверджує висновок, що поведінка системи є хаотичною при $\tau = 0.28725$. Проведена чисельна перевірка, що розв'язки для граничних циклів є періодичними та розраховано періоди циклів для кожного локального максимуму і мінімуму.

У випадку $N = 32$, проаналізували модель біосенсора за допомогою решітчатого графіку, що представляє флуоресцентні пікселі, при аналізі моделювання флуоресценції, відповідно до умови (5.1), покладали $\Theta_{fl} = 1{,}5$. Спочатку було отримано решітчасті зображення, що вказують на число зв'язків антигенів і антитіл всередині пікселя (Рис. 5.3).

$$k_{fl} V_{i,j}(t) F_{i,j}(t) \geq \Theta_{fl} \tag{5.1}$$

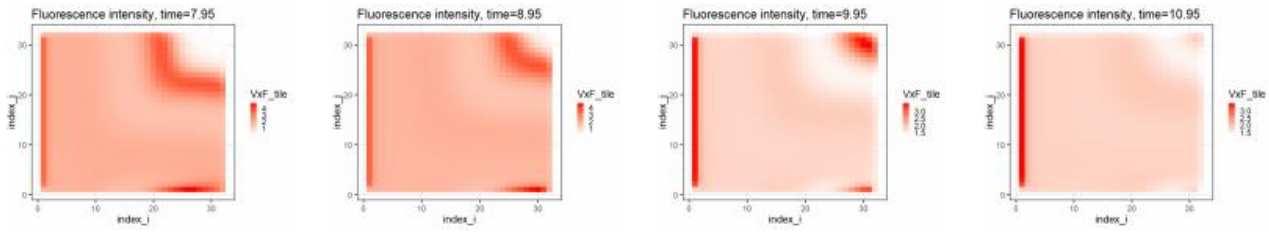


Рисунок 5.3 – Решітчасті зображення зв`язків антигенів з антитілами в пікселях системи (4.2) при $n = 0.9, \tau = 0.05$.

Дослідження флуоресціюючих пікселів і вихідного сигналу, величина якого залежить від їх кількості, повністю відповідає попереднім дослідженням стійкості масиву біосенсорів.

Порівняння проводилося з пороговим значенням Θ_{fl}, було отримано флуоресцентні пікселі для дослідження. Отже, у випадку $\tau = 0.05$ спостерігаємо самостійність. Так само при $t = 11$, спостерігаємо усі пікселі, у яких відбувається явище флуоресценції (рис. 5.4, 5.5 для кількості флуоресціюючих пікселів).

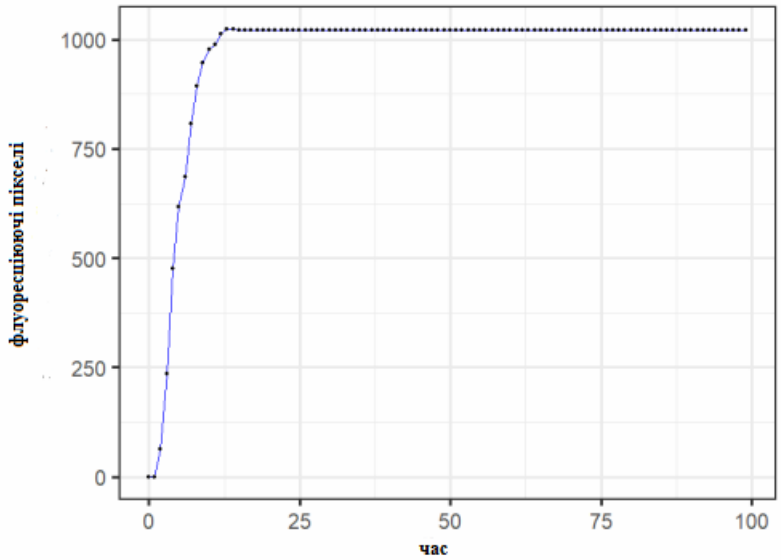


Рисунок 5.4 – Елекричний сигнал з трансдюсера, який характеризує число флуоресціюючих пікселів при $\tau = 0.05$.

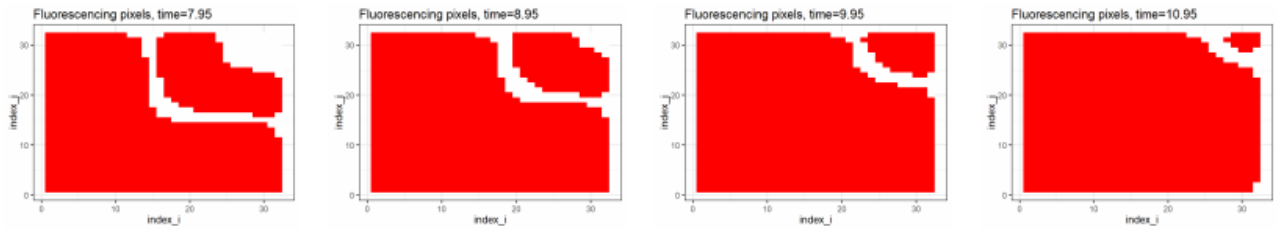


Рисунок 5.5 – Зображення флуоресценції системи (4.2), як результат чисельного моделювання, при $n = 0.9, \tau = 0.05$.

На рис. 5.5 наведено результат чисельного моделювання системи (4.2) при $n = 0.9, \tau = 0.05$, який відповідає стійкому фокусу (спостерігається біжуча хвиля не флуоресціюючих пікселів). Порогове значення для флуоресценції при цьому становить $\Theta_{fl} = 1{,}5$.

Далі, збільшуючи запізнення часу, ми отримуємо біфуркацію Хопфа, яка вже відзначалась вище і отримуємо відповідно 1 періодичний граничний цикл (рис. 5.6, 5.7 для τ = 0.24).

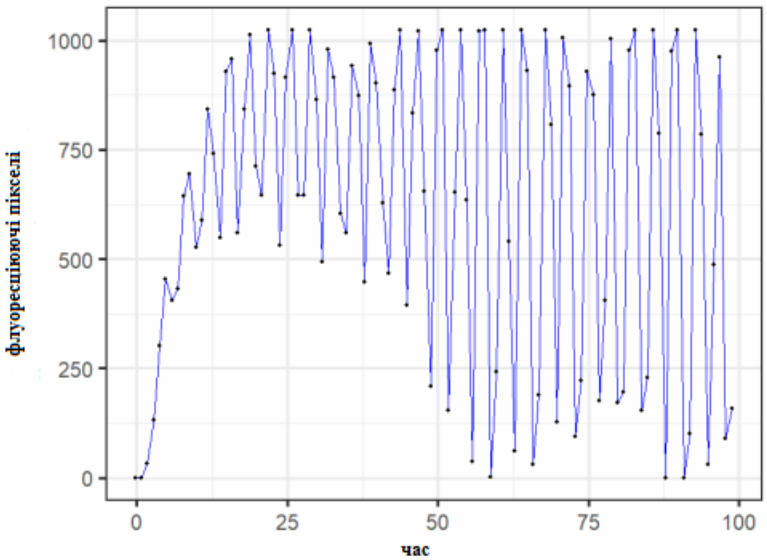
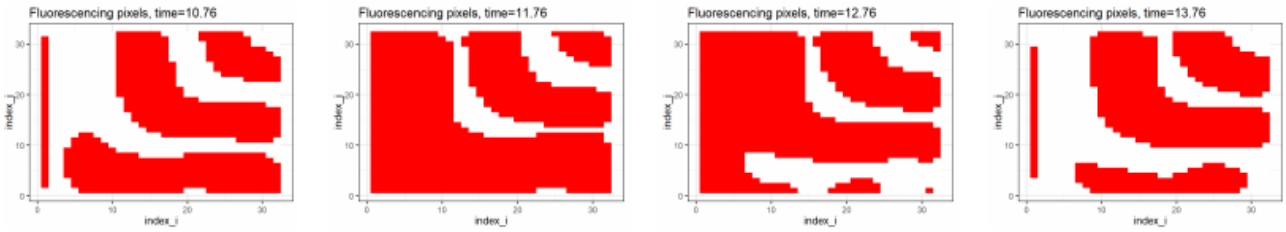


Рисунок 5.6 – Елекричний сигнал з трансдюсера, який характеризує число флуоресціюючих пікселів при τ = 0.24.



Рисунку 5.7 – Зображення флуоресценції системи (4.2), як результат чисельного моделювання, при $n = 0.9, \tau = 0.24$.

На рис. 5.6 наведено результат чисельного моделювання системи (4.2) при $n = 0.9, \tau = 0.24$, який відповідає прагненню до граничного циклу (спостерігається біжуча хвиля флуоресціюючих пікселів).

На рис. 5.8 наведено результат чисельного моделювання системи (4.2) при $n = 0.9, \tau = 0.28$, також спостерігається біжуча хвиля флуоресціюючих пікселів.

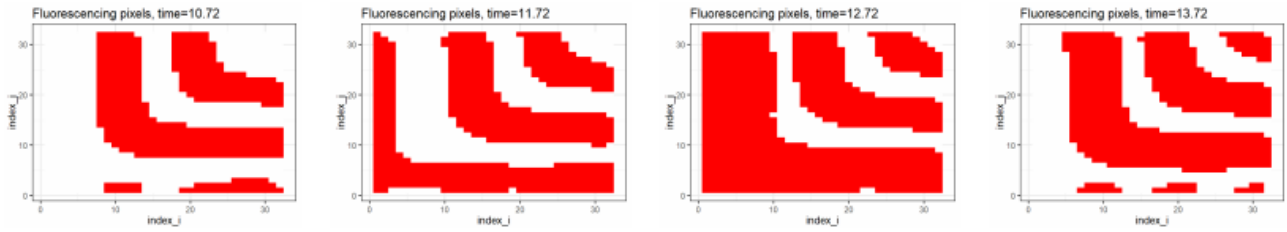


Рисунок 5.8 – Зображення флуоресценсії системи (4.2), як результат чисельного моделювання, при $n = 0.9, \tau = 0.28$.

Електричний сигнал який характеризується числом флуоресціюючих пікселів обгрунтовує детерміністичний ефект (Рис. 5.8).

У випадку τ =0.28 спостерігаємо хаотичну поведінку, яка починається з хвилеподібних змін у флуоресціюючих пікселях (Рис. 5.8) та швидко переходить до хаотичних змін (Рис. 5.9).

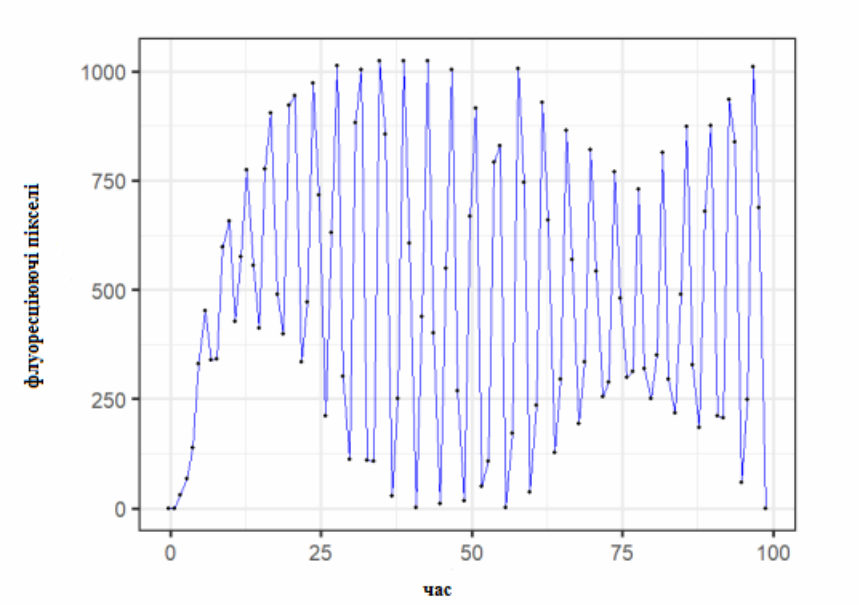


Рисунок 5.9 – Елекричний сигнал з трансдюсера, який характеризує число флуоресціюючих пікселів при $\tau = 0.28$.

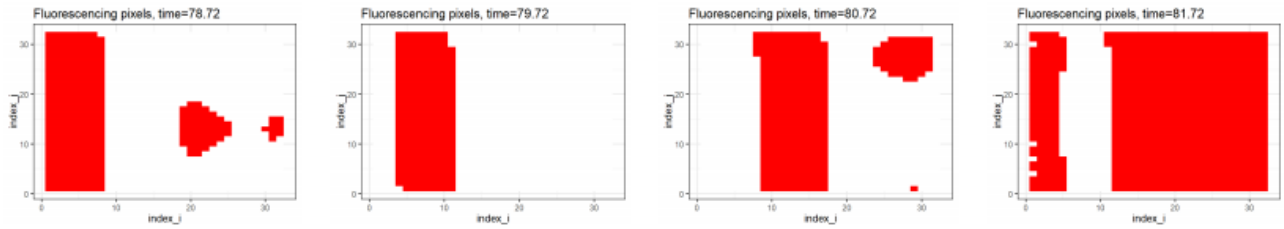


Рисунок 5.10 – Забраження флуоресценсії системи (4.2), як результат чисельного моделювання, при $n = 0.9$, $\tau = 0.28725$.

На рис. 5.10 наведено результат чисельного моделювання системи (4.2) $(n = 0.9), \tau = 0.28725$ при якому спостерігається хаотична хвиля флуоресціюючих пікселів.

Таким чином запропонований спосіб дозволяє оптимальним чином підібрати параметри кіберфізичної імуносенсорної системи (початкові умови, швидкісні параметри та час імунної відповіді), при цьому забезпечивши операційну або самостійкість.

ДОДАТОК С
АКТИ ВПРОВАДЖЕННЯ

„ЗАТВЕРДЖУЮ"
Проректор з наукової роботи
Тернопільського національного
медичного університету
імені І.Я. Горбачевського,
заслужений діяч науки і техніки України,
доктор біологічних наук, професор
Кліщ І.М.

„4" лютого 2021 р.

Акт впровадження

1. *Назва пропозиції для впровадження:* Підхід до вирішення проблем машинного навчання, інтегрованих з даними систем із відкритим кодом електронних медичних записів.
2. *Ким запропоновано, виконавець, адреса:* Мілян Н.В. аспірант кафедри кібербезпеки Тернопільського національного технічного університету імені Івана Пулюя, кафедра кібербезпеки, м. Тернопіль, вул. Руська 56.
3. *Джерело інформації:* стаття Мілян Н.В. Марценюк В. П. Approach to Solving Machine Learning Problems Integrated with Data of the Systems with Open Code of Electronic Medical Records. Вісник ТНТУ. Т. : ТНТУ, 2019. Том 95. № 3. С. 105–116.
4. *Де впроваджено:* Тернопільський національний медичний університет імені І. Я. Горбачевського, м. Тернопіль, вул. Руська, 12, кафедра медичної інформатики.
5. *Термін впровадження:* 2021 р.
6. *Загальна кількість спостережень:* 45.
7. *Висновок по впровадженню:* Використання підходу до розробки моделі машинного навчання для медичних досліджень який ґрунтується на використанні програмного забезпечення з відкритим кодом. Підхід використовується в навчальному процесі на кафедрі медичної інформатики при вивченні дисципліни "Медична інформатика" (тема: "Методи біостатистики. Аналіз біосигналів. Методи обробки біосигналів").

Доцент кафедри
медичної інформатики,
доктор технічних наук, доцент Сверстюк А.С.

„ЗАТВЕРДЖУЮ"
Проректор з наукової роботи
Тернопільського національного
медичного університету
імені І.Я. Горбачевського,
заслужений діяч науки і техніки України,
доктор біологічних наук, професор
Кліщ І.М.

"4" лютого 2021 р.

Акт впровадження

1. *Назва пропозиції для впровадження:* Реалізація алгоритму індукції дерева рішень на хмарних платформах.
2. *Ким запропоновано, виконавець, адреса:* Мілян Н.В. аспірант кафедри кібербезпеки Тернопільського національного технічного університету імені Івана Пулюя, кафедра кібербезпеки, м. Тернопіль, вул. Руська, 56.
3. *Джерело інформації:* стаття V. Martsenyuk, Z. Mayhruk, M. Karpinski, N. Milian, I. Andrushchak and O. Veselska, "On implementation of decision tree induction in cloud platforms," 2019 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 2019, pp. 1-6, doi: 10.1109/ICASET.2019.8714450.
4. *Де впроваджено:* Тернопільський національний медичний університет імені І.Я. Горбачевського, м. Тернопіль, вул. Дорошенка, 2, кафедра медичної фізики діагностичного та лікувального обладнання.
5. *Термін впровадження:* 2021 р.
6. *Загальна кількість спостережень:* 48.
7. *Висновок по впровадженню:* Використання запропонованої програмної реалізації дає змогу проводити автоматизований аналіз категоризованих даних на платформах хмарних обчислень. Рекомендуємо до включення в педагогічний процес при викладанні навчальної дисципліни «Медична фізика діагностичного та лікувального обладнанння», «Медична та біологічна фізика» на відповідних практичних заняттях «Фізичні основи електрокардіографії. Електрокардіографія».

Завідувач кафедри медичної фізики
діагностичного та лікувального обладнанння,
доктор технічних наук, доцент Рудяк Ю.А.

ЗАТВЕРДЖУЮ
Проректор з наукової роботи
Тернопільського національного
технічного університету ім. І. Пулюя
_____ П. О. Марущак
10 лютого 2021р.

АКТ

про впровадження результатів дисертаційної роботи «Моделі і методи прийняття рішень при створенні клінічних діагностичних систем» аспіранта кафедри кібербезпеки Міляна Назара Васильовича, представленої на здобуття наукового ступеня доктора філософії, в навчальному процесі Тернопільського національного технічного університету ім. І. Пулюя.

Цим актом підтверджується, що результати дисертаційної роботи аспіранта кафедри кібербезпеки Міляна Н.В. використано при проведенні лабораторних занять з дисциплін «Інтелектуальний аналіз даних» (освітній рівень - бакалавр) та «Методи та системи підтримки прийняття рішень» (освітній рівень - магістр) для студентів спеціальності 125 «Кібербезпека».

Мілян Н.В. запропонував метод розв'язування задач класичного і глибокого машинного навчання в умовах невизначеності на основі мінімаксного підходу.

Використання вказаних результатів у навчальному процесі дало змогу студентам ознайомитись із сучасними підходами до розв'язку задач класичного та глибокого машинного навчання, зокрема використовуючи мінімаксний підхід.

Завідувач кафедри кібербезпеки,
к.т.н., доцент Н. В. Загородна