Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)
Дослідження методів машинного навчання для прогнозування розвитку діабету
(тема)

Виконав:
здобувач 2 року навчання
групи ІПЗм-23-3

Олег ЛЯПОТА
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник доц. Олексій НАЗАРОВ
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Ляпоті Олегу Владиславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів машинного навчання для прогнозування розвитку діабету
Затверджена наказом по університету від 15.04. 2025р. № 290 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16.06.2025
3. Вихідні дані до роботи опис досліджуваних методів машинного навчання для прогнозування діабету, структура бази даних для зберігання медичних показників пацієнтів, результатів прогнозування та моделей, використані мова програмування Python, технології Flask, scikit-learn, TensorFlow, XGBoost, СУБД MySQL, MongoDB, середовище розробки PyCharm 2024.
4. Перелік питань, що потрібно опрацювати в роботі
аналіз сучасних методів прогнозування діабету, вибір алгоритмів машинного навчання, проектування схеми бази даних і логічної архітектури системи, реалізація REST API, навчання моделей, оцінка точності, порівняння результатів, формування висновків та рекомендацій щодо впровадження.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	16.04.2025	*виконано*
2	Аналіз предметної галузі і постановка задачі	17.04.2025-21.04.2025	*виконано*
3	Огляд наукової літератури, аналіз стану проблеми, формулювання задачі	22.04.2025-26.04.2025	*виконано*
4	Теоретичні основи дослідження, опис методів машинного навчання	27.04.2025-01.05.2025	*виконано*
5	Підготовка до апробації результатів дослідження. Публікація матеріалів	02.05.2025-06.05.2025	*виконано*
6	Проведення експериментів, аналіз результатів, обґрунтування вибору моделей	07.05.2025-11.05.2025	*виконано*
7	Підготовка пояснювальної записки	12.05.2025-16.05.2025	*виконано*
8	Підготовка презентації та доповіді	17.05.2025-21.05.2025	*виконано*
9	Перевірка на плагіат	22.05.2025-26.05.2025	*виконано*
10	Нормоконтроль	27.05.2025-31.05.2025	*виконано*
11	Рецензування	01.06.2025-05.06.2025	*виконано*
12	Попередній захист	06.06.2025-10.06.2025	*виконано*
13	Занесення диплома в електронний архів	11.06.2025-14.06.2025	*виконано*
14	Допуск до захисту у зав. кафедри	15.06.2025	*виконано*

Дата видачі завдання 16 квітня 2025р.

Здобувач _____ Олег ЛЯПОТА
(підпис)

Керівник роботи _____ доц. Олексій НАЗАРОВ
(підпис) (посада, Власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 66 с., 5 рис., 8 табл., 11 джерел.

ДІАБЕТ, КЛАСИФІКАЦІЯ, МАШИННЕ НАВЧАННЯ, НЕЙРОННА МЕРЕЖА, ПРОГНОЗУВАННЯ, PYTHON, SMOTE.

Об’єктом дослідження є процеси ранньої діагностики та прогнозування ризику розвитку діабету.

Метою роботи є проектування ефективної системи прогнозування ризику розвитку діабету на основі алгоритмів машинного навчання для підвищення точності діагностики та прийняття рішень у медичній практиці.

Методами розробки та проектування є аналіз існуючих алгоритмів машинного навчання, таких як дерево прийняття рішень, логістична регресія, випадковий ліс, екстремальне посилення градієнту, багатошаровий персептрон та згорточна нейронна мережа.

У результаті роботи було розроблено архітектуру системи, включаючи UML-діаграми, схему бази даних і моделі взаємодії компонентів. Реалізовано процес обробки даних для підготовки до прогнозування, обґрунтовано вибір мови Python як платформи реалізації завдяки її високій популярності, великій кількості спеціалізованих бібліотек та фінансовій доступності. Реалізовано RESTful API для навчання моделей, отримання прогнозів та керування даними пацієнтів і результатами. Сформовано рекомендації щодо впровадження розробленої системи в клінічну практику для ранньої діагностики діабету.

DIABETES, CLASSIFICATION, MACHINE LEARNING, NEURAL NETWORK, PREDICTION, PYTHON, SMOTE.

The object of the study is the processes of early diagnosis and prediction of the risk of developing diabetes.

The aim of the project is to design an effective diabetes risk prediction system based on machine learning algorithms to improve diagnostic accuracy and decision-making in medical practice.

The development and design methods include the analysis of existing machine learning algorithms such as decision trees, logistic regression, random forest, extreme gradient boosting, multilayer perceptron, and convolutional neural network.

As a result of the work, the system architecture was developed, including UML diagrams, a database schema, and models of component interaction. The data processing pipeline was modeled for prediction preparation, and the choice of Python as the implementation platform was justified due to its high popularity, the availability of specialized libraries, and cost-efficiency. A RESTful API was implemented for model training, prediction, and management of patient data and results. Recommendations were formed for integrating the developed system into clinical practice for early diabetes diagnosis.

Завідувачу кафедри
ПІ_____
(скорочена назва кафедри)
проф. Кирилу СМЕЛЯКОВУ_
(вчене звання, сласне ім'я, прізвище)

ЗАЯВА
щодо самостійності виконання кваліфікаційної роботи та можливості її публікації (та/або публікації анотації кваліфікаційної роботи) в електронному архіві відкритого доступу ElAr KhNURE

Я, Ляпота Олег Владиславович, студент гр. ІПЗм-23-3, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів машинного навчання для прогнозування розвитку діабету», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(на) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата Підпис

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

AI – Artificial Intelligence

API – Application Programming Interface

AUC – Area Under Curve

BMI – Body Mass Index

CNN – Convolutional Neural Network

CRUD – Create Read Update Delete

DB – Database

JSON – JavaScript Object Notation

ML – Machine Learning

MLP – Multilayer Perceptron

NoSQL – Not Only SQL

ORM – Object-Relational Mapping

REST API – Representational State Transfer Application Programming Interface

ROC – Receiver Operating Characteristic

SMOTE – Synthetic Minority Over-sampling Technique

SQL – Structured Query Language

UML – Unified Modeling Language

XGBoost – Extreme Gradient Boosting

ВСТУП

Цукровий діабет залишається однією з найпоширеніших хронічних хвороб у світі, щороку зростаючи в масштабах і негативно впливаючи на якість життя мільйонів людей. За даними ВООЗ, проблема діабету набуває глобального характеру, що обумовлює необхідність раннього виявлення ризику захворювання та вжиття профілактичних заходів. Традиційні методи діагностики базуються на ручному аналізі медичних показників і не завжди забезпечують своєчасність та точність. У цьому контексті застосування методів машинного навчання (ML) дозволяє автоматизувати процес обробки медичних даних, виявляти приховані закономірності та значно підвищити ефективність прогнозування розвитку діабету.

Розробка інтелектуальних систем для прогнозування ризику розвитку діабету є важливою складовою сучасної медичної аналітики. Використання машинного навчання в поєднанні з клінічними даними дозволяє створити гнучкі, точні та масштабовані рішення, що можуть бути інтегровані в медичні інформаційні системи для прийняття обґрунтованих рішень.

Робота виконується в межах наукового напряму кафедри Програмної інженерії ХНУРЕ, пов’язаного з інтелектуальним аналізом даних, медичним прогнозуванням, розробкою програмного забезпечення для підтримки прийняття рішень та систем реального часу. Отримані результати можуть бути використані в подальших наукових розробках у рамках проєктів кафедри.

Метою роботи є розробка програмної системи прогнозування ризику розвитку діабету, що поєднує сучасні алгоритми машинного навчання, обробку медичних даних і REST-архітектуру.

Для досягнення мети необхідно провести аналіз предметної області та наукових джерел, обґрунтувати вибір алгоритмів машинного навчання, розробити архітектуру програмної системи, реалізувати модулі обробки даних, навчання моделей та прогнозування, провести порівняльний аналіз точності моделей, інтегрувати систему у вигляді RESTful API.

Об’єкт дослідження – процеси діагностики та прогнозування розвитку діабету на основі клінічних даних.

Предмет дослідження – алгоритми машинного навчання та програмні засоби для аналізу медичних ознак (рівень глюкози, індекс маси тіла, вік, спадковість) з метою прогнозування розвитку діабету.

У роботі застосовано аналітичні методи для огляду літератури, методи машинного навчання: логістична регресія, дерево рішень, Random Forest, XGBoost, MLP, CNN, методи обробки даних: нормалізація, SMOTE, інженерія ознак, програмні засоби реалізації: Python, Flask, SQLAlchemy, scikit-learn, TensorFlow, MongoDB, MySQL, методи оцінки моделей: Accuracy, Precision, Recall, F1-score, ROC AUC.

Вперше розроблено модульну архітектуру системи прогнозування діабету з можливістю гнучкої інтеграції нових моделей ML, застосовано інженерію ознак та метод SMOTE для покращення результатів класифікації. Отримано модель, яка поєднує ефективність ансамблевих підходів із зручністю використання в медичному середовищі.

Розроблений прототип системи може бути впроваджений у заклади охорони здоров’я як інструмент підтримки прийняття клінічних рішень. Його використання сприяє підвищенню точності діагностики, зменшенню часу аналізу даних та оптимізації лікувального процесу.

Основні результати дослідження опубліковано у тезах міжнародної науково-практичної конференції “Mathematical and Information Technologies in Applied and Information Systems” (MIT&AISs 2025), м. Харків, Україна.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі дослідження

Цукровий діабет є хронічним метаболічним захворюванням, яке характеризується стійким підвищенням рівня глюкози в крові внаслідок недостатньої продукції інсуліну або зниження чутливості тканин до нього. За даними Всесвітньої організації охорони здоров'я, кількість хворих на діабет з кожним роком зростає, досягаючи епідемічного рівня. Станом на початок 2020-х років кількість людей із діабетом перевищила 460 мільйонів, і прогнози вказують на подальше зростання цієї кількості.

Медична значущість діабету полягає не лише у безпосередній шкоді, яку завдає порушення вуглеводного обміну, але й у високому ризику розвитку ускладнень, таких як інсульт, інфаркт, хронічна ниркова недостатність, сліпота та ампутації. Зважаючи на це, критично важливою є своєчасна діагностика та оцінка ризику розвитку діабету на ранніх етапах, до появи клінічних симптомів.

У клінічній практиці застосовуються кілька основних методів діагностики діабету:

- аналіз рівня глюкози натще;
- пероральний глюкозотолерантний тест;
- оцінка рівня глікованого гемоглобіну – важливий довготривалий показник;
- моніторинг індексу маси тіла;
- аналіз спадковості.

Проте традиційні методи діагностики здебільшого є реактивними: вони виявляють діабет, коли вже сформовані клінічні ознаки. Це обмежує їхню ефективність для раннього виявлення осіб з підвищеним ризиком, що могло б дозволити вжити профілактичних заходів ще до розвитку захворювання.

У цьому контексті особливого значення набувають інтелектуальні системи прогнозування, що ґрунтуються на методах машинного навчання. Ці системи здатні аналізувати багатовимірні медичні дані – включаючи рівень глюкози, ІМТ, вік, стать, спадкову схильність, артеріальний тиск та інші параметри – з метою

виявлення прихованих закономірностей і побудови моделей оцінки ризику розвитку діабету з високою точністю

Однією з основних переваг таких підходів є можливість індивідуалізації прогнозу – адаптації моделі до конкретного пацієнта, враховуючи унікальне поєднання факторів ризику. При цьому особливу складність становить проблема незбалансованості медичних даних, що може призводити до помилкових негативних результатів. Для розв'язання цієї проблеми ефективно застосовуються алгоритми синтетичного збагачення, зокрема SMOTE, який дозволяє підвищити чутливість моделі до рідкісних випадків хвороби.

Загалом, сучасні системи прогнозування, що базуються на машинному навчанні, відкривають нові можливості для профілактики, ранньої діагностики та персоналізованого медичного обслуговування, підвищуючи якість життя пацієнтів та знижуючи навантаження на систему охорони здоров'я.

1.2 Постановка задачі

На сучасному етапі розвитку медицини актуальною проблемою залишається раннє виявлення ризику розвитку цукрового діабету. Традиційні підходи до діагностики, що ґрунтуються переважно на аналізі рівня глюкози в крові та глікованого гемоглобіну, хоча й ефективні, не дозволяють повною мірою виявити приховані закономірності в багатовимірних медичних даних, особливо на доклінічній стадії захворювання.

У відповідь на це, в останні роки зросла увага до використання методів машинного навчання для автоматизованої оцінки ризику розвитку діабету. Такі методи здатні працювати з великими обсягами різнорідних даних, виявляти нелінійні залежності між показниками стану здоров'я пацієнтів і формувати точні прогнози. У результаті дослідження необхідно вирішити набір проблем, що виникають перед існуючими рішеннях з використанням методів машинного навчання:

– необхідність вибору релевантних моделей машинного навчання для задач медичного прогнозування;

- обробка незбалансованих даних, у яких кількість пацієнтів без діабету значно перевищує кількість хворих;
- забезпечення точності, чутливості та інтерпретованості моделей для використання в реальних клінічних умовах;
- інтеграція моделей у програмну систему, зручну для лікарів і медичного персоналу.

Для вирішення наведених проблем сформуємо завдання дослідження як розробка ефективної системи прогнозування ризику розвитку діабету на основі алгоритмів машинного навчання, здатної обробляти медичні дані, формувати точні прогнози для індивідуальних пацієнтів та забезпечувати інтеграцію в клінічну інформаційну систему.

Для вирішення задачі необхідно:

- провести огляд наукової літератури та аналіз існуючих моделей прогнозування діабету;
- обґрунтувати вибір методів машинного навчання для розв’язання задачі;
- сформувати та підготувати медичний датасет, провести нормалізацію, обробку пропущених значень, застосувати техніку SMOTE для вирівнювання класів;
- реалізувати та навчити кілька моделей машинного навчання;
- здійснити порівняльний аналіз моделей за критеріями;
- інтегрувати моделі у програмну систему, реалізувавши REST API для взаємодії з клієнтом;
- сформулювати практичні рекомендації щодо використання системи у клінічній практиці.

Таким чином, робота орієнтована не лише на теоретичне дослідження ефективності ML-методів для медичного прогнозування, а й на створення функціонального прототипу програмної системи, яка може бути використана для підвищення якості медичної діагностики та підтримки клінічних рішень.

1.3 Порівняльний аналіз технологій

Для розробки проекту необхідно обрати технологію, що буде використовуватись для реалізації методів машинного навчання. Було розглянуто наступний набір найбільш поширених технологій, що використовуються для реалізації методів машинного навчання:

- Python;
- R;
- MATLAB;
- Java;
- Julia.

Для вибору найкращої технології було використано лінійну адитивну згортку з ваговими коефіцієнтами. Для виконання задачі вибору необхідно було сформувати множину критеріїв вибору для порівняння технологій та вагові коефіцієнти, що відповідали-б цим критеріям:

- кількість бібліотек що розширюють функціонал технології і вказують на гнучкість роботи з технологією;
- ціна ліцензії для комерційного використання обрана для того, щоб забезпечити подальшу впроваджуваність розробленого рішення;
- кількість репозиторіїв на GitHub, що вказує на популярність та розмір спільноти та простоту подальшої підтримки такого додатку;
- відсоток на щорічному Stack Overflow Developer Survey, що дозволяє свідчити про актуальність використання технології;
- кількість публікацій на Google Scholar дозволить розглянути престижність технологій, ступінь її використання у наукових дослідженнях.

Значення усіх критеріїв лежать на шкалах відношень в початку відліку на нулі. Внесемо значення критеріїв у порівняльну таблицю (див. табл. 1.1).

Таблиця 1.1 – Критерії вибору для технології машинного навчання (таблиця виконана самостійно)

	Кількість бібліотек	Ціна ліцензії (USD)	Кількість тегів на GitHub	Відсоток на Stack Overflow Developer Survey	Кількість публікацій на Google Scholar
Python	137 000	0	18 200 000	46.9	610 000
R	21 926	0	927 000	3.1	3 870 000
MATLAB	81	1015	413 000	3	324 000
Java	224	0	17 500 000	30	1 970 000
Julia	10 000	0	76 400	0.8	148 000

Для використання коефіцієнтів необхідно було провести певні перетворення. Ціну ліцензії треба інвертувати так, щоб найбільше значення відповідало найбільш бажаному, для цього віднімемо значення ціни від максимального, отримавши таким чином економію покупки ліцензії (див. таб. 1.2).

Таблиця 1.2 – Критерії вибору для технології машинного навчання з інвертованою ціною ліцензій (таблиця виконана самостійно)

	Кількість бібліотек	Економія покупки ліцензії (USD)	Кількість тегів на GitHub	Відсоток на Stack Overflow Developer Survey	Кількість публікацій на Google Scholar
Python	137 000	1015	18 200 000	46.9	610 000
R	21 926	1015	927 000	3.1	3 870 000
MATLAB	81	0	413 000	3	324 000
Java	224	1015	17 500 000	30	1 970 000
Julia	10 000	1015	76 400	0.8	148 000

Для використання лінійної адитивної згортки критерії необхідно нормалізувати по принципу “до максимуму”.

Нормалізація “до максимуму” розраховується відносно максимального та мінімального значення відповідного критерію. (див. табл. 1.3).

Таблиця 1.3 – Нормалізовані критерії вибору для технології машинного навчання (таблиця виконана самостійно)

	Кількість бібліотек	Економія покупки ліцензії (USD)	Кількість тегів на GitHub	Відсоток на Stack Overflow Developer Survey	Кількість публікацій на Google Scholar
Python	1	1	1	1	0.124
R	0.16	1	0.047	0.05	1
MATLAB	0	0	0.019	0.05	0.047
Java	0,001	1	0.961	0.633	0.49
Julia	0,072	1	0	0	0

Метод Парето дозволяє зменшити кількість варіантів, що розглядаються. Метод Парето має наступне визначення: “Варіант a краще варіанту b згідно з відношенням Парето, якщо a хоча б за одним критерієм краще ніж b, а по іншим критеріям не гірше, ніж b“. Варіант MATLAB можна виключити так як за відношенням Парето він є гіршим за інші варіанти:

- Python домінує Matlab по всім критеріям;
- R домінує Matlab по всім критеріям окрім Відсоток на Stack Overflow Developer Survey, який є рівним;
- Java домінує Matlab по всім критеріям;
- Julia програє по трьмо критеріям і домінує по 2.

У результаті бачимо, що 3 мови повністю домінують MATLAB, чого достатньо для виключення за методом Паретто (див. табл. 1.4).

Таблиця 1.4 – Нормалізовані критерії вибору для технології машинного навчання після використання методу Парето.

	Кількість бібліотек	Економія покупки ліцензії (USD)	Кількість тегів на GitHub	Відсоток на Stack Overflow Developer Survey	Кількість публікацій на Google Scholar
Python	1	1	1	1	0.124
R	0.16	1	0.047	0.05	1
Java	0,001	1	0.961	0.633	0.49
Julia	0,072	1	0	0	0

Для використання лінійної адитивної згортки необхідно ввести вагові коефіцієнти. Визначимо коефіцієнти методом простого ранжування. Для цього розподілимо критерії по вагомості.

Найбільш важливою визнано кількість бібліотек, оскільки вона безпосередньо впливає на функціональні можливості та підтримку технології. Далі йдуть кількість тегів на GitHub, економія на ліцензії, популярність серед розробників за даними Stack Overflow і, найменш важлива це кількість публікацій у Google Scholar, яка скоріше відображає академічну, ніж прикладну значущість. (див. табл. 1.5).

Таблиця 1.5 – Вагові коефіцієнти за методом простого ранжування.

Важливість критерію	Назва критерію	Розрахунок коефіцієнту
1	Кількість публікацій на Google Scholar	$\frac{1}{15} = 0{,}067$
2	Відсоток на Stack Overflow Developer Survey	$\frac{2}{15} = 0{,}133$
3	Економія покупки ліцензії (USD)	$\frac{3}{15} = 0{,}2$
4	Кількість тегів на GitHub	$\frac{4}{15} = 0{,}267$
5	Кількість бібліотек	$\frac{5}{15} = 0{,}333$

Використовуючи присвоєні коефіцієнти розрахуємо значення лінійної адитивної згортки для кожної з технологій, та оберемо набійльш перспективну:

- Python – 0,941;
- R – 0,339;
- Java – 0,574;
- Julia – 0,224.

Бачимо, що найбільш перспективною технологією для виконання проекту є Python. Основне значення у вибір технології внесла наявність великої кількості бібліотек, що використовуються для машинного навчання та обробки даних. Велика кількість бібліотек дозволить досягти гнучкості у реалізації системи для подальших досліджень.

2 ОГЛЯД ТА АНАЛІЗ ЛІТЕРАТУРНИХ І ПАТЕНТНИХ ДЖЕРЕЛ

2.1 Сучасні підходи до медичного прогнозування з використанням ML

У досліджених джерелах було розглянуто різні аспекти діагностики, класифікації та прогнозування діабету за допомогою методів машинного навчання. Джерела можна розподілити на декілька груп:

– прогнозування діабету за допомогою машинного навчання [1-6];
– статистичні данні по глобальній ситуації з розвитком діабету [7];
– розробка додатків для прогнозування діабету [8-9];
– особливості комп'ютерної обробки медичних даних [10-11].

Актуальність джерел досягнута шляхом вибору публікацій, що були випущенні не раніше ніж у 2014 році, тобто за 10 років до виконання курсового проекту.

Дані про авторитетність ресурсів було засновано на даних сайту scimagojr.com:

– Journal of Personalized Medicine має квартиль Q2 у галузі медицини та високий h-індекс 51;
– Journal of Physics: Conference Series має квартиль Q4 у галузі фізики та астрономії, проте h-індекс 99 є дуже високим;
– International Journal of Intelligent Systems має кватиль Q1 у галузях штучного інтелекту, людино-машиних інтерацій, програмного забезпечення, теоретичних комп'ютерних наук, та дуже високий h-індекс в 106;
– International Journal of Cognitive Computing in Engineering має квартиль Q1 у галузях комп'ютерних наук, інженерії, інформаційних систем та менеджменту, проте h-індекс в 16 є невисоким;
– PLoS ONE має квартиль Q1 у мультидисциплінарній галузі та неймовірно високий h-індекс 435;
– Journal of diabetes science and technology має квартилі Q1 у галузях біоінженерії, біомедичної інженерії та внутрішньої медицини та квартиль Q2 у галузі ендокринології, діабету та метаболізму та високий h-індекс 93.

Як ми бачимо, джерела є престижними та багатократно цитованими, що визначає їх авторитетність.

У наведених наукових працях доведено ефективність застосування ML-алгоритмів для задач діагностики та прогнозування діабету. Наприклад, у дослідженні Alghamdi M було запропоновано використання ансамблевих моделей у поєднанні з технікою SMOTE для підвищення чутливості моделі до рідкісних випадків. Модель показала значне покращення метрик класифікації, зокрема Recall, що критично важливо для клінічної практики.

Серед розглянутих підходів ML можна виділити:

– логістичну регресію – базовий, але ефективний метод для бінарної класифікації, що може використовуватись як еталонний для більш складних моделей;
– дерева рішень та випадковий ліс – забезпечують хорошу узагальненість та здатні моделювати нелінійні залежності навіть на незбалансованих наборах даних;
– XGBoost – метод градієнтного бустингу, який отримав популярність через свою ефективність у табличних медичних даних;
– MLP та CNN – сучасні нейронні архітектури, які добре працюють з великими наборами даних і дозволяють враховувати складні кореляції між параметрами.

Огляд сучасної літератури підтверджує, що ML-моделі ефективно прогнозують розвиток діабету, особливо за умови правильної підготовки даних. У дослідженнях широко застосовується попередня обробка даних, включаючи:

– нормалізацію ознак;
– заміну або усунення пропущених значень;
– синтетичне збалансування класів з використанням методу SMOTE, який дозволяє збільшити кількість випадків меншості без втрати якості навчання.

Крім технічної ефективності, у сучасних роботах все більше уваги приділяється питанням інтерпретованості моделей та їх адаптації до клінічного використання. Наприклад, методи логістичної регресії або дерева рішень надають лікарям чітке уявлення про значення кожного медичного параметра, що важливо з точки зору клінічного довіри до системи.

Azure Cosmos DB широко використовується у власних платформах електронної комерції Microsoft, на яких працюють Windows Store і XBox Live. Він також використовується в галузі роздрібної торгівлі для зберігання даних каталогу та пошуку подій у конвеєрах обробки замовлень [3].

2.2 Математичні моделі алгоритмів

У рамках проведення дослідження необхідно розглянути та порівняти різноманітні методи машинного навчання.

Дерево прийняття рішень – це ієрархічна структура, яка розділяє набір даних на менші підгрупи, базуючись на ознаках. Основний алгоритм побудови дерева використовує принцип мінімізації ентропії або максимізації інформаційного зиску.

Розрахунок інформаційного зиску наведено у формулі:

$$IG(S,A) = Entropy(S) - \sum_{v \in Values(A)} |S_v||S| \cdot Entropy(S_v) \tag{2.1}$$

де S – поточний набір даних,

S_v – підмножина S, яка відповідає значенню .

Розширення дерев, такі як Random Forest і Gradient Boosted Trees, значно підвищують точність завдяки ансамблевим методам.

Логістична регресія використовується для моделювання ймовірності належності до певного класу. Основою є логістична функція:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{2.2}$$

де x – вектор вхідних ознак,

θ – вектор ваг моделі.

Функція втрат для логістичної регресії:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] \quad (2.3)$$

Завдяки простоті та інтерпретованості, логістична регресія слугує базовим методом для порівняння складніших алгоритмів.

Екстремальне посилення градієнту – це ансамблевий метод, який використовує ітеративне посилення слабких моделей (дерев). Алгоритм мінімізує функцію втрат:

$$L = \sum_{i=1}^{n}\ell(y_i,\, \dot{y}_i.) + \sum_{k=1}^{K}\Omega(f_k) \quad (2.4)$$

де ℓ – функція втрат,

$\Omega(f_k)$ – регуляризація для контролю складності.

Завдяки оптимізації обчислень і регуляризації, XGBoost демонструє високу точність і масштабованість.

Багатошаровий персептрон є базовим видом нейронних мереж. Він складається з шарів нейронів, які використовують функцію активації для нелінійного перетворення вхідних даних.

Математичний опис представлено функцією:

$$h^{(l)} = f\left(W^{(l)}h^{(l-1)} + b^{(l)}\right) \quad (2.5)$$

де $W^{(l)}$ – матриця ваг,

$b^{(l)}$ – вектор зміщення,

f – функція активації.

Функція втрат для задач класифікації:

$$J(W, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log y^{(i)} + \left(1 - y^{(i)}\right)\log\left(1 - y^{(i)}\right)\right] \quad (2.6)$$

Використання сучасних оптимізаторів значно покращує продуктивність.

Згорточна нейронна мережа застосовується для аналізу структурованих даних, наприклад, зображень або складних табличних даних. Основна операція описується формулою:

$$h_{ij}^{(l)} = f\left(\sum_{m=1}^{M}\sum_{n=1}^{N} \omega_{mn}^{(l)}\, x_{(i+m)(j+n)}^{(l-1)} + b^{(l)}\right) \tag{2.7}$$

де $\omega_{mn}^{(l)}$– ядро згортки,

$x_{ij}^{(l-1)}$– вхідні дані.

Завдяки спільному використанню фільтрів і пулінгу, згорточна нейронна мережа є дуже ефективними для багатовимірних даних.

Обрані алгоритми мають широку сферу застосування та демонструють нові підходи до розв'язання задач прогнозування. Їхня комбінація дозволяє створити точні та адаптивні моделі для прогнозування діабету.

3 ОБҐРУНТУВАННЯ ТА ВИБІР МЕТОДІВ ДОСЛІДЖЕННЯ

3.1 Вибір критеріїв для оцінки моделей

Для оцінки ефективності моделей машинного навчання, які застосовуються у медичному прогнозуванні, зокрема для передбачення розвитку діабету, необхідно використовувати набір метрик, що здатні комплексно охарактеризувати якість класифікації. Особливо важливим є коректний вибір показників у контексті незбалансованих даних, де кількість здорових пацієнтів значно перевищує кількість хворих, що типово для медичних датасетів (Pima Indians Diabetes Dataset).

Accuracy – загальна точність показує частку правильно класифікованих випадків серед усіх прикладів. Проте ця метрика може бути оманливо високою у випадку значної дисбалансованості даних, коли модель переважно прогнозує більшість класу. Математичний сенс описано формулою:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3.1}$$

де TP – вірно визначені позитивні випадки,

TN – вірно визначені негативні випадки,

FP – невірно визначені позитивні випадки,

FN – вірно визначені позитивні випадки.

Precision – точність позитивного прогнозу визначає частку правильних позитивних передбачень серед усіх передбачених як позитивні. У контексті діабету Precision важливий для зменшення кількості хибно позитивних випадків, що може знизити емоційне і ресурсне навантаження на пацієнта і систему охорони здоров’я. Математичний сенс описано формулою:

$$Pr\,e\,cision = \frac{(TP)}{(TP + FP)} \tag{3.2}$$

де TP – вірно визначені позитивні випадки,

FP – невірно визначені позитивні випадки.

Recall – чутливість відображає здатність моделі виявляти всі справжні позитивні випадки. Високий Recall критично важливий для медичних задач, де пропущений діагноз може мати тяжкі наслідки. Математичний сенс описано формулою:

$$Recall = \frac{(TP)}{(TP + FN)} \tag{3.3}$$

де TP – вірно визначені позитивні випадки,

FN – невірно визначені негативані випадки.

F1 Score – Гармонійне середнє між Precision та Recall. Застосовується для пошуку компромісу між точністю та чутливістю моделі. Ідеальний інструмент для задач з незбалансованими даними. Математичний сенс описано формулою:

$$F1 = 2 \cdot \frac{(\Pr e\,cision \cdot Recall)}{\Pr e\,cision + Recall} \tag{3.4}$$

ROC AUC – Оцінює здатність моделі відрізняти позитивні та негативні класи при зміні порогу прийняття рішення, де наближення значення ROC AUC до 1 означає якість класифікатору. Математичний сенс описано формулою:

$$AUC = \int_0^{-1} TPR(FPR)dFPR \tag{3.5}$$

де TPR – True Positive Rate,

FPR – False Positive Rate.

У задачах, пов’язаних із діагностикою, перевагу зазвичай надають Recall, щоб не пропустити хворих та AUC як інтегральному показнику якості класифікації. Разом із тим, F1 Score використовується як збалансована метрика для порівняння різних моделей у складних умовах.

У межах даної роботи було використано всі п’ять критеріїв, що дозволило об’єктивно оцінити моделі з різних аспектів.

3.2 Методика експериментального дослідження

Для побудови моделей було використано відкритий медичний датасет – Pima Indians Diabetes Dataset, що містить 768 записів з ознаками здоров'я пацієнтів, зокрема:

- рівень глюкози у крові;
- артеріальний тиск;
- товщина шкірної складки;
- індекс маси тіла;
- кількість вагітностей;
- вік;
- рівень інсуліну;
- спадкова схильність.

Перед роботою з даними виконується видалення або заміна нульових та пропущених значень у клінічних полях. Данні нормалізуються, тобто приводяться до діапазону [0, 1] за допомогою Min-Max Scaling, що забезпечує рівномірний вплив усіх ознак при навчанні моделей. Через значний дисбаланс: 65% здорових і 35% хворих, було використано техніку SMOTE для синтетичного збагачення меншого класу, що підвищує чутливість моделей до рідкісних випадків.

Для підвищення інтерпретованості моделей було проведено інженерію ознак та додано синтетичні ознаки на базі існуючих:

- glucose_BMI_ratio – співвідношення між рівнем глюкози та індексом маси тіла, що дозволяє визначати аномальні випадки підвищеного рівня глюкози в крові;
- is_obese – бінарний індикатор, що показує чи відповідає BMI порогу ожиріння, та полегшує роботу по катигоризації моделям;
- age_group – категоризація пацієнтів за віком також полегшує задачу категоризації для моделей, так як ризик виникнення діабету залежить саме від вікової групи, а не конкретного віку пацієнта;
- HbA1c_level – оцінка глікованого гемоглобіну на основі рівня глюкози, додає моделям інтерпретованості так як являється найпоширенішим методом визначення діабету.

Навчання моделей проводилось з використанням бібліотек scikit-learn, TensorFlow/Keras, XGBoost. Параметри оптимізувалися з використанням крос-валідації (K=5) для уникнення переобучення та перевірки узагальнювальної здатності моделей.

Кожну модель тестували на незалежній вибірці після навчання. Метрики є стандартними для задач медичної класифікації, де критично важливо уникнути як хибнонегативних, так і хибнопозитивних випадків.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ

4.1 Архітектура програмного забезпечення

Розроблена система прогнозування розвитку діабету реалізована у вигляді вебсервісу з використанням фреймворку Flask. Програмна архітектура підтримує як інтерфейс користувача через HTML-шаблони, так і зовнішній REST API для інтеграції з іншими медичними інформаційними системами. Реалізація базується на модульному підході з чітким розподілом відповідальності між компонентами.

Складемо схему бази даних. Вона має зберігати основну інформацію про пацієнтів, медичні записи, прогнози та моделі машинного навчання (див. рис. 4.1).

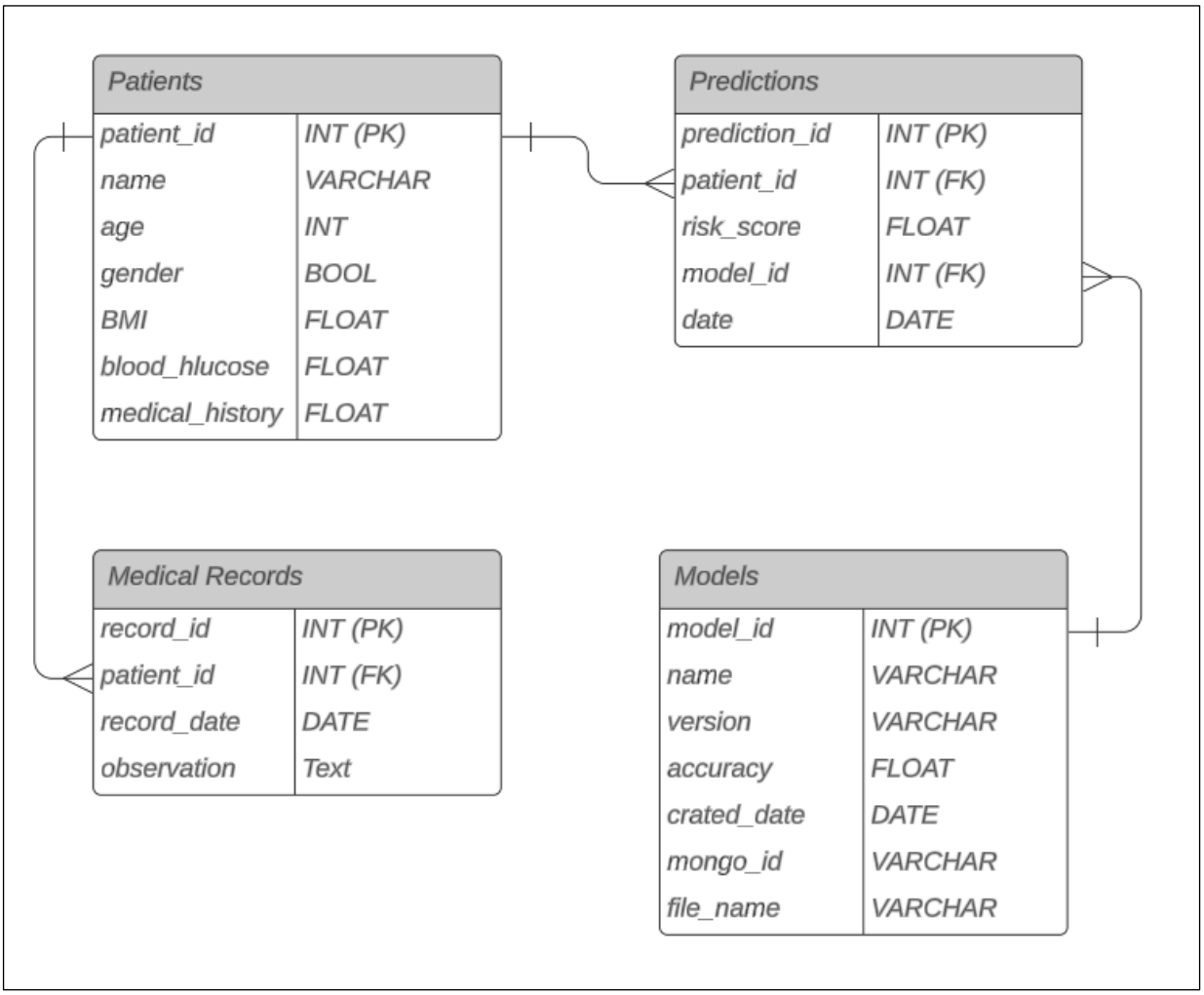

Рисунок 4.1 – Схема реляційної бази даних

Інформація про пацієнта містить набір особистої інформацій та факторів ризику, що можуть вплинути на результат прогнозу:

- patient_id, ключ таблиці, ціле число;
- name, ім’я пацієнту, строкове значення;
- age, ціла кількість років пацієнту, ціле число;
- gender, біологічна стать пацієнту, булеве значення;
- BMI, індекс маси тіла, число з плаваючою точкою;
- blood_hlucose, середній рівень глюкози у крові, число з плаваючою точкою;
- medical_history, вірогідність діабету на базі сімейної історії хвороби, число з плаваючою точкою.

Інформація про медичні записи містить інформацію про результатти певного огляду певного пацієнту:

- record_id, ключ таблиці, ціле число;
- patient_id, зовнішній ключ з таблиці пацієнтів, ціле число;
- record_date, дата запису, дата та час;
- observation, результат запису, текст.

У таблиці models міститься інформація про певну версію моделі:

- model_id, ключ таблиці, ціле число;
- name, назва моделі, текст;
- version, версія моделі, текст;
- accuracy, точність моделі, число з плаваючою точкою;
- created_date, дата створення моделі, дата та час;
- mongo_id, ідентифікатор для строки MongoDB, що містить параметри конкретної моделі, текст;
- filename, назва файлу в якому зберігається відповідна модель.

Таблиці patients та models поєднує між собою таблиця predictions, реалізуючи зв’язок багато до багатьох:

- prediction_id, ключ таблиці, ціле число;
- patient_id, зовнішній ключ з таблиці пацієнтів, ціле число;
- model_id, зовнішній ключ з таблиці моделей, ціле число;
- risk_score, вірогідність наявності діабету, число з плаваючою точкою;

– date, дата прогнозу, дата та час.

Для керування користувачами використовується таблиця users. (див. рис. 4.2)

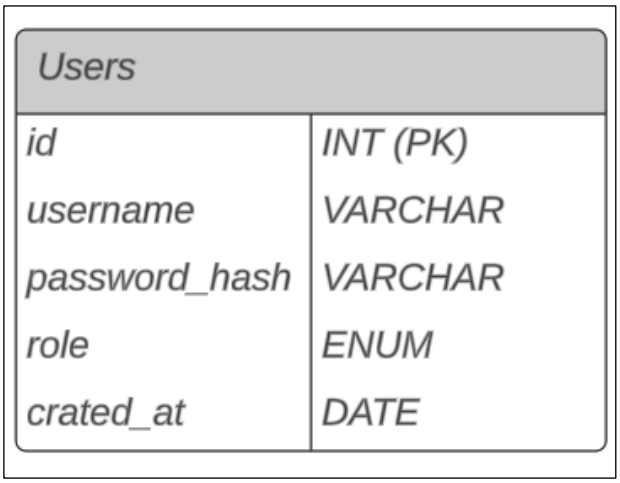

Рисунок 4.2 – Таблиця users

Таблиця users зберігає інформацію про користувачів системи:

– id, унікальний ідентифікатор;

– username, ім’я користувача;

– password_hash, пароль зашифрований через алгоритм bcrypt;

– role, роль користувача у системі Admin чи Doctor;

– created_at, дата створення користувача.

Моделі навчаються, валідуються та тестуються на наборі даних, що зберігається у таблицях бази даних (див. рис. 4.3).

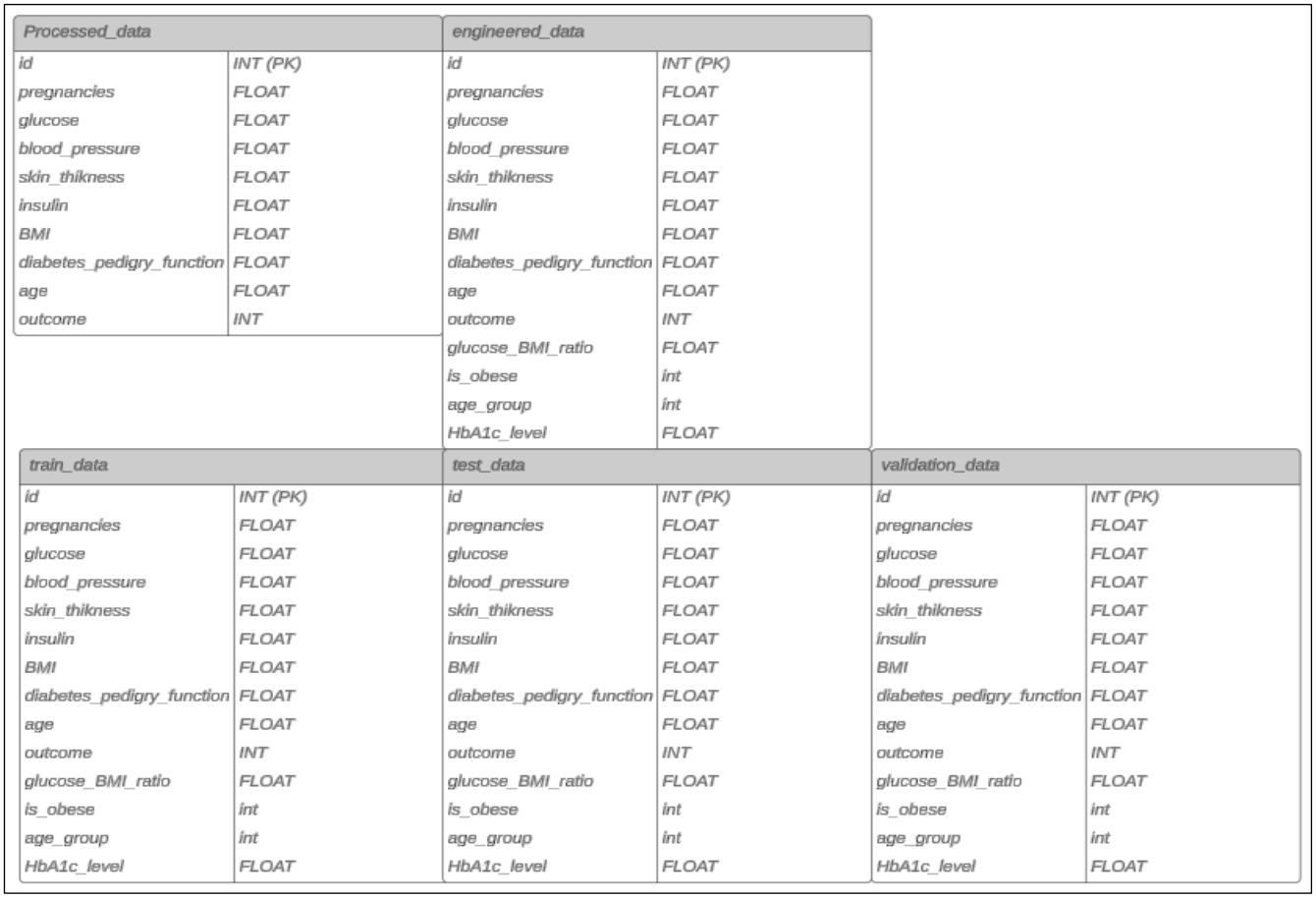

Рисунок 4.3 – Таблиці для машинного навчання

Таблиця processed_data містить набір даних з нормалізованими, та відсутніми пропущеними даними:

- id, ключ таблиці, ціле число;
- pregnancies, кількість перенесених пацієнтом вагітностей, число з плаваючою точкою;
- glucose, середній рівень цукру в крові, число з плаваючою точкою;
- blood_pressure, середя величина кров’яного тиску, число з плаваючою точкою;
- skin_thikness, товщина шкіри, число з плаваючою точкою;
- insullin, середній рівень інсуліну у крові, число з плаваючою точкою;
- BMI, індекс маси тіла, число з плаваючою точкою;
- diabetes_pedigri_function, вірогідність розвитку діабету згідно сімейної історії хвороби, число з плаваючою точкою;
- age, вік пацієнта, число з плаваючою точкою;

- outcome, наявність, або відсутність діабету, ціле число 1 чи 0 відповідно.

Таблиця engineered_data представляє собою дата сет після проведення інженерії залежностей, у результаті якого розраховано набір додаткових ознак:

- glucose_BMI_ratio, відношення рівня цукру до індексу маси тіла, допомагає з визначенням аномальних відхилень, число з плаваючою точкою;
- is_obese, вказує чи є у пацієнта зайва вага, значно прискорює роботу по навчанню моделі, ціле число;
- age_group, категоріальна ознака, що групує вік пацієнта у діапазоні, допомагає визначати групи ризику, ціле число;
- HbA1c_level, рівень глікозильованого гемоглобіну, критична довгострокова характеристика, число з плаваючою точкою.

Таблиці test_data, train_data, validation_data представляють датсет розбитий відповідно на тестову, тренувальну та валідаційну вибірку у відношенні 70/15/15.

База даних MongoDB представляє собою колекцію записів, кожен з яких зберігає параметри та метрики відповідної моделі БД, варіант записо представлено у вигляді JSON строки.

Для моделювання ключових компонентів системи було використання UML моделі. Діаграма класів використовується для опису зв'язків між об'єктами програмного забезпечення (див. рис. 4.4).

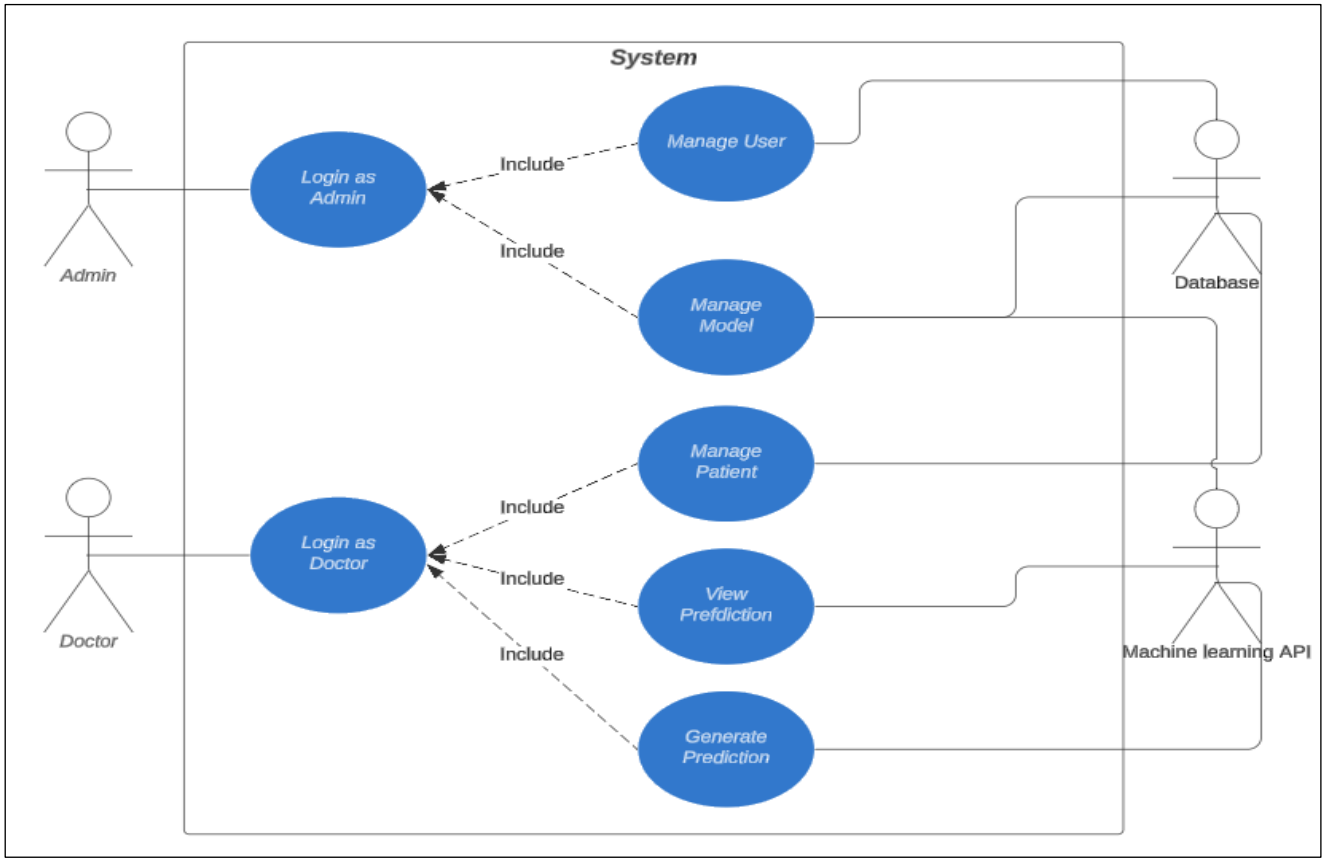

Рисунок 4.4 – Use Case сценарії

У системі наявно 4 актори:

– Admin, актор, що виконує управління системою;
– Doctor, актор, що використовує систему;
– Database, актор, що зберігає данні;
– Machine learning API, модуль, що реалізує машине навчання та прогнозування розвитку діабету.

У актора Admin наявно наступні сценарії виконання:

– Login as Admin, основний сценарій виконання, що вимагається для інших сценаріїв, передбачає вхід в акаунт з правами адміністратора;
– Manage User, додавання, видалення та оновлення інформації про інших користувачів системи, вимагає з'єднання з актором Database;
– Manage Model, додавання, видалення та оновлення характеристик для моделей машинного навчання, тренування нових моделей, вимагає з'єднання з акторами Database та Machine learning API.

У актора Doctor наявно наступні сценарії виконання:

– Login as Doctor, основний сценарій виконання, що вимагається для інших сценаріїв, передбачає вхід в акаунт з правами доктора;
– Manage User, додавання, видалення та оновлення інформації про пацієнтів, вимагає з'єднання з актором Database;
– View Prediction, перегляд прогнозів сгенерованних для відповідного пацієнту, вимагає з'єднання з акторами Database та Machine learning API;
– Generate Prediction, генерація прогнозів розвитку діабету для відповідного пацієнту, вимагає з'єднання з акторами Database та Machine learning API.

Проектування інтерфейсу користувача виконується з урахуванням потреб медичного персоналу. Основний інтерфейс включатиме інтуїтивно зрозумілий дизайн для введення даних пацієнтів і відображення результатів прогнозів.

Для перегляду структури майбутнього додатку розглянемо діаграму розгортання (див. рис. 4.5).

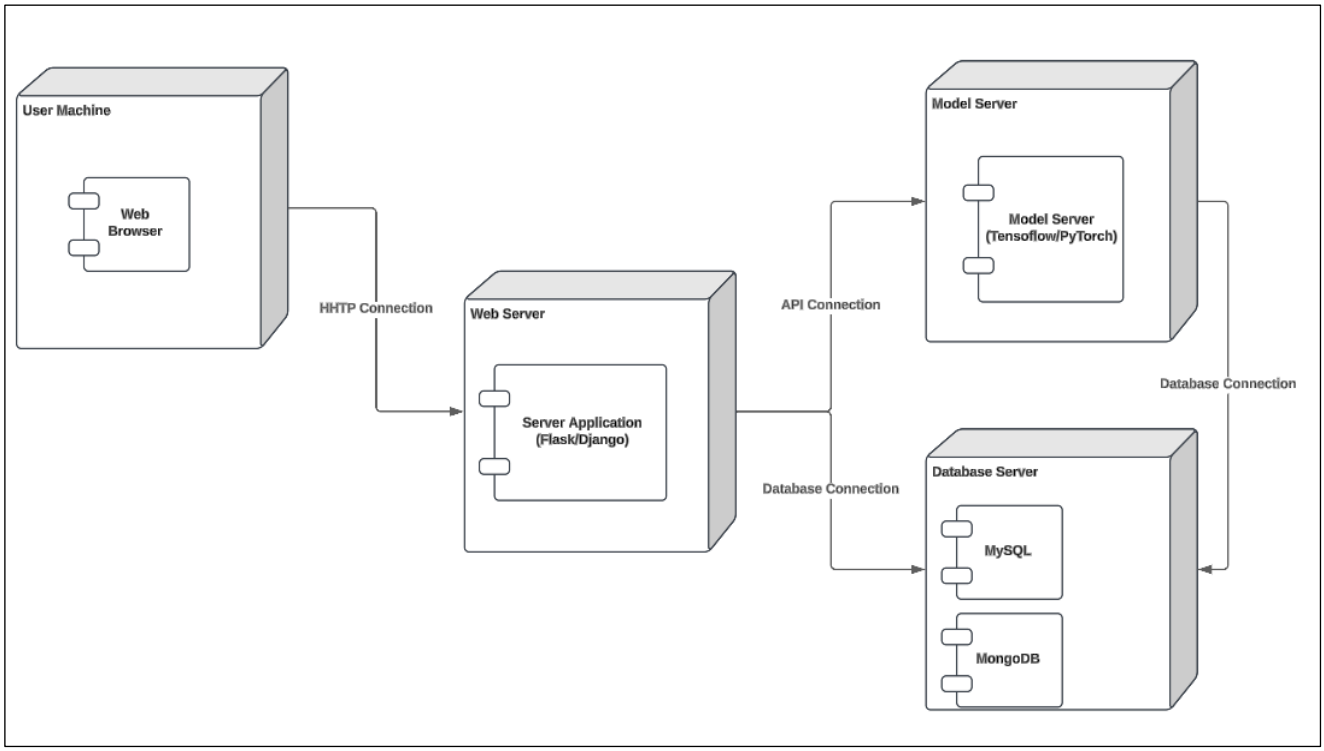

Рисунок 4.5 – Діаграма розгортання

Система складається з двох частин: клієнтської та серверної. Клієнтська частина працює на машині користувача, у вигляді веб браузера.

Серверна частина розподіляється на веб-сервер, сервер бази даних та API сервер.

Веб-сервер розгортає серверний додаток, що опрацьовує HTTP запити.

API сервер розгортає систему обробки моделей машинного навчання. Центральний контролер реалізовано у модулі app.py. Контролер запускає серверний Flask додаток, реєструє методи, що будуть обробляти кінцеві точки API, обробляє підключення до бази даних, реалізує механізм авторизації за допомогою токенізації

Логіка моделей ML реалізовано у модулі model_utils.py. Модуль реалізує навчання моделей ML, прогнозування ризику розвитку діабету для пацієнту та отримання метрик моделей, необхідних для їх аналітичного порівняння.

Найкращі версії моделей зберігаються у форматі .pkl у рамках директорії /models.

У модулі db_utils.py реалізується ORM через бібліотеку SQLAlchemy для сутностей User, Patient, Prediction та CRUD операції для них.

4.2 Інтеграція моделей ML у REST-сервіс

Інтеграція алгоритмів машинного навчання в систему прогнозування реалізована через модуль model_utils.py, який тісно пов'язаний із REST-сервером на Flask. Архітектура побудована таким чином, щоб забезпечити автоматичне навчання, збереження, використання і оновлення моделей без потреби в прямому втручанні користувача в ML-код.

У model_utils.py реалізовано навчання моделей LogisticRegression, DecisionTreeClassifier, RandomForestClassifier, XGBClassifier, MLPClassifier, Sequential.

Моделі створюються, навчаються та оцінюються в єдиному інтерфейсі train_best_model():

```
  if model_type == "LogisticRegression":
params.pop("max_iter", None)
model = LogisticRegression(**params)
```

```
elif model_type == "RandomForest":
    model = RandomForestClassifier(**params)

elif model_type == "XGBoost":
    model = XGBClassifier(eval_metric="logloss", **params)

elif model_type == "DecisionTree":
    model = DecisionTreeClassifier(**params)

elif model_type == "MLP":
    model = MLPClassifier(**params)

elif model_type == "CNN":
```

Після навчання моделі зберігаються у файл за допомогою joblib.dump() для всіх моделей, окрім CNN, яка зберігається за допомогою model.save() бібліотеки Keras.

Метадані моделі: ім’я, гіперпараметри, хеш, характеристики для порівняльного аналізу записуються у Mongo DB і SQL.

Метод save_model_metadata() зберігає дані моделі у SQL:

```
      def save_model_metadata(name, version, accuracy, mongo_id=None,
filename=None):
    session = Session()

    stmt = select(Model).where(Model.name == name, Model.version == version)
    existing_model = session.execute(stmt).scalars().first()

    if existing_model:
        existing_model.accuracy = accuracy
        existing_model.created_date = datetime.now()
        existing_model.mongo_id = mongo_id
        existing_model.filename = filename
        model_id = existing_model.model_id
    else:
        new_model = Model(
            name=name,
```

```
            version=version,
            accuracy=accuracy,
            mongo_id=mongo_id,
            filename=filename
        )
        session.add(new_model)
        session.commit()
        model_id = new_model.model_id

    if not existing_model:
        session.commit()

    session.close()
    return model_id
```

Метод update_model_metrics_in_mongo() обновляє метрики моделі у разі, якщо після оновлення даних метрики змінились:

```
      d def update_model_metrics_in_mongo(mongo_id, metrics):
    db.models.update_one(
        {"_id": ObjectId(mongo_id)},
        {"$set": {"metrics": metrics}}
    )
```

Метод save_model_to_mongo() зберігає метрики та параметри моделі у MongoDB:

```
      def save_model_to_mongo(model_type, params, filename, accuracy,
metrics=None):
    doc = {
        "model_type": model_type,
        "params": params,
        "accuracy": accuracy,
        "filename": filename,
        "created_at": datetime.now()
    }
    if metrics:
        doc["metrics"] = metrics
```

```
    existing = db.models.find_one({
        "model_type": model_type,
        "params": params
    })

    if existing:
        db.models.update_one(
            {"_id": existing["_id"]},
            {"$set": doc}
        )
        return str(existing["_id"])
    else:
        result = db.models.insert_one(doc)
        return str(result.inserted_id)
```

Метод predict_risk() обирає вже наявну навчену модель, завантажує її з файлу та повертає вірогідність розвитку діабету для наведеного набору характеристик:

```
      if filename.endswith(".h5") or filename.endswith(".keras"):
    model = load_model(filename)
    df = preprocess_features(df)
    X = df[FEATURES].values
    X = np.expand_dims(X, axis=-1)  # (samples, features, 1)
    prob = model.predict(X)[0][0]
    return {"risk_score": float(prob), "model_file": filename, "model_type":
"CNN"}

else:
    model = joblib.load(filename)
    df = preprocess_features(df)
    X = df[FEATURES]
    prob = model.predict_proba(X)[:, 1]
    return {"risk_score": float(prob[0]), "model_file": filename}
```

Flask додаток визначає методи, що обробляють HTTP запити. /predict приймає POST запит з JSON-об’єктом та повертає результат виконання метода predict_risk():

```
@app.route('/predict/<int:model_id>', methods=['POST'])
def predict_with_model(model_id):
    data = request.json
    df = pd.DataFrame([data])
    prediction = predict_risk(df, model_id)
    return jsonify(prediction)
```

/train викликає метод train_best_model() для навчання нової моделі:

```
@app.route('/train', methods=['POST'])
def train():
    data = request.json or {}
    model_type = data.get("model_type", "LogisticRegression")
    params = data.get("params", {})

    result = train_best_model(model_type, params)
    return jsonify(result)
```

Інтеграція моделей машинного навчання у REST-сервіс реалізована на практично орієнтованому рівні. Вона забезпечує автоматизоване навчання, зберігання, обробку запитів прогнозування і доступ до результатів через захищений API. Такий підхід дозволяє розгорнути сервіс у медичному середовищі та інтегрувати його у більші IT-інфраструктури з мінімальними зусиллями.

Після завершення планування експериментів та написання програм для їх проведення можна перейти до замірів усіх необхідних параметрів, а саме швидкості виконання запитів у мс, кількості витраченої програмою пам’яті на виконання запиту та кількості RU зі сторони БД. Для кожного експерименту проводилося виконання запитів та розраховувалося їх середнє значення.

5 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ

В рамках експериментальних досліджень для кожної моделі після її створення проводилась перевірка на валідаційній вибірці, по результатам котрої було отримано метрики для порівняльного аналізу.

5.1 Порівняльний аналіз результатів

Найкращі показники для кожного методу машинного навчання (див. табл. 5.1).

Таблиця 5.1 – Характеритсики для порівняльного аналізу методів машиного навчання для прогнозування розвитку діабету (таблиця виконана самостійно)

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	84.81%	83.33%	87.50%	85.37%	91.54%
Decision Tree	79.75%	74.00%	92.50%	82.22%	84.71%
Random Forest	86.08%	83.72%	90.00%	86.75%	94.78%
XGBoost	86.08%	82.22%	92.50%	87.06%	94.17%
MLP	82.28%	80.95%	85.00%	82.93%	90.96%
CNN	83.54%	81.40%	87.50%	84.34%	90.38%

Через реалізації усіх параметрів у відсотковому діапазоні [00.00% - 100.00 %] додатковою нормалізації наші данні не вимагають. Метод Парето дозволяє зменшити кількість варіантів, що розглядаються. Метод Парето має наступне визначення: “Варіант a краще варіанту b згідно з відношенням Парето, якщо a хоча б за одним критерієм краще ніж b, а по іншим критеріям не гірше, ніж b“. Шляхом Парето фільтрації можна виключити з подальшого аналізу методи Decision Tree та MLP, що повністю домінуються конкурентами (див. табл. 5.2).

Таблиця 5.2 – Характеристики для порівняльного аналізу методів машинного навчання для прогнозування розвитку діабету після використання методу Парето (таблиця виконана самостійно)

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	84.81%	83.33%	87.50%	85.37%	91.54%
Random Forest	86.08%	83.72%	90.00%	86.75%	94.78%
XGBoost	86.08%	82.22%	92.50%	87.06%	94.17%
CNN	83.54%	81.40%	87.50%	84.34%	90.38%

Для використання лінійної адитивної згортки необхідно ввести вагові коефіцієнти. Визначимо коефіцієнти методом простого ранжування. Для цього розподілимо критерії по вагомості та розрахуємо значення вагових коефіцієнтів в залежності від загальної кількості (див. табл. 5.3).

Таблиця 5.3 – Вагові коефіцієнти за методом простого ранжування (таблиця виконана самостійно)

Важливість критерію	Назва критерію	Розрахунок коефіцієнту
1	ROC AUC	$\frac{1}{15} = 0.067$
2	Recal	$\frac{2}{15} = 0.133$
3	Precision	$\frac{3}{15} = 0.2$
4	Accuracy	$\frac{4}{15} = 0.267$
5	F1 Score	$\frac{5}{15} = 0.333$

Використовуючи сформовані коефіцієнти розрахуємо значення лінійної адитивної згортки для кожного метода ML та оберемо найбільш перспективну:

- Logistic Regresion – 85.48;
- Random Forest – 86.91;

– XGBoost – 87.04;

– CNN – 84.33.

У результаті бачимо, що найкращі результати дають асамблеві методи XGBoost та Random Forest, так як вони найкраще підходять для задачі мультикритеріального вибору і продемонстрували високі показники при виконанні медичних задач, зокрема в задачі прогнозування розвитку діабету.

5.2 Опрацювання результатів

Результати експериментального дослідження дозволили оцінити ефективність різних моделей машинного навчання при розв’язанні задачі прогнозування розвитку діабету на основі реальних медичних даних. Аналіз результатів за основними метриками показав, що найбільш стабільно працюють ансамблеві методи – Random Forest та XGBoost.

XGBoost продемонстрував найкращий баланс між точністю, чутливістю та узагальнюваністю. Його F1 Score склав 87.06%, Recall – 92.50%, що свідчить про високу здатність моделі виявляти хворих, не створюючи при цьому надлишкових хибнопозитивних спрацювань.

Random Forest показав ще вищу ROC AUC – 94.78%, що підтверджує його надійність у класифікації пацієнтів із різними рівнями ризику. Проте загальна збалансованість метрик була дещо нижча порівняно з XGBoost.

Нейронні мережі показали конкурентні, але не провідні результати. Це можна пояснити обмеженим обсягом даних, що обмежує здатність глибоких моделей до узагальнення.

Decision Tree мала високу Recall – 92.50%, але значно нижчу точність, що робить її менш надійною для практичного застосування через велику кількість хибних діагнозів.

Logistic Regression хоча і поступається ансамблевим методам за F1 Score, демонструє високу інтерпретованість і швидкість, що робить її корисною у контексті попереднього скринінгу або як частину гібридної системи.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено методику побудови інтелектуальної системи прогнозування розвитку діабету з використанням сучасних методів машинного навчання. Методика передбачає етапи попередньої обробки медичних даних, синтетичного балансування класів, генерації нових ознак, навчання кількох моделей класифікації та порівняльної оцінки за ключовими метриками.

Експериментальна частина дослідження показала, що ансамблеві методи, зокрема XGBoost та Random Forest, демонструють найвищу точність класифікації ризику розвитку діабету на основі Pima Indians Diabetes Dataset. Модель XGBoost досягла F1 Score 87.06% і Recall 92.5%, що свідчить про її ефективність у виявленні хворих без втрати загальної точності.

Інтеграція моделей у REST-сервіс на базі Flask дозволила створити гнучку та масштабовану систему з API-доступом до функцій навчання, прогнозування та управління пацієнтами. Це відкриває можливості для інтеграції системи у медичні інформаційні платформи, мобільні додатки або системи підтримки прийняття рішень у лікарській практиці.

Практична значущість дослідження полягає у створенні універсального інструменту ранньої діагностики діабету, що може використовуватись для медичного скринінгу у клініках первинної ланки, попереднього самодіагностування в мобільних застосунках, аналітики у профілактичних програмах охорони здоров’я.

Очікувана ефективність від впровадження системи – це зниження частки пропущених випадків діабету, підвищення точності медичних рішень, автоматизація рутинних діагностичних задач та зменшення навантаження на медичний персонал.

Подальші напрями дослідження включають використання глибших нейронних мереж на розширених датасетах, так як розміри датасету не дозволили використати повний функціонал нейронних мереж.

Можливо провести валідацію системи на даних з інших популяцій та медичних закладів, так як виконання роботи було обмежене наявністю відповідних датасетів у відкритому доступу.

Можливе розширення системи до мультидіагностичних задач, для уможливлювання діагностування суміжних ендокринологічних задач.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Chou, C. Y., Hsu, D. Y., & Chou, C. H. (2023). Predicting the Onset of Diabetes with Machine Learning Methods. Journal of personalized medicine, 13(3), 406. https://doi.org/10.3390/jpm13030406.

2. Abdelhafez, H., & Amer, A. (2024). Machine Learning Techniques for Diabetes Prediction: A Comparative Analysis. Journal of Applied Data Sciences, 5(2), 792-807. doi:https://doi.org/10.47738/jads.v5i2.219.

3. Jingyu Xue, Fanchao Min and Fengying Ma, "Research on Diabetes Prediction Method Based on Machine Learning", у 2020 Journal of Physics Conference Series 1684:012062, https://iopscience.iop.org/article/10.1088/1742-6596/1684/1/012062#references.

4. Alghamdi M, Al-Mallah M, Keteyian S, Brawner C, Ehrman J, et al. (2017) Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford ExercIse Testing (FIT) project. PLOS ONE 12(7): e0179805. https://doi.org/10.1371/journal.pone.0179805.

5. Dagliati A, Marini S, Sacchi L, et al. Machine Learning Methods to Predict Diabetes Complications. Journal of Diabetes Science and Technology. 2018;12(2):295-302. doi:10.1177/1932296817706375.

6. Diabetes. URL: https://www.who.int/news-room/fact-sheets/detail/diabetes (дата звернення: 17.12.2024.).

7. El-Sofany, Hosam, El-Seoud, Samir A., Karam, Omar H., Abd El-Latif, Yasser M., Taj-Eddin, Islam A. T. F., A Proposed Technique Using Machine Learning for the Prediction of Diabetes Disease through a Mobile App, International Journal of Intelligent Systems, 2024, 6688934, 13 pages, 2024. https://doi.org/10.1155/2024/6688934.

8. Nazin Ahmed, Rayhan Ahammed, Md. Manowarul Islam, Md. Ashraf Uddin, Arnisha Akhter, Md. Alamin Talukder, Bikash Kumar Paul, "Machine learning based diabetes prediction and development of smart web application" 2021 International Journal of Cognitive Computing in Engineering Volume 2, Pages 229-241, https://doi.org/10.1016/j.ijcce.2021.12.001.

9. Єрохін, А. Л., Турута, О. П., Нечипоренко, А. С., Бабій, А. С. Proc. of the

International Conference on Computer Sciences and Information Technologies, Lviv, Ukraine, 2017, 332-335. http://openarchive.nure.ua/handle/document/4137

10. Synthesis of Structured Models of Computer Systems in Medical Diagnosis / N. Bilous, A. Povoroznyuk, O. Kozina // International Book Series «Information Science and Computing» Intelligent Information and Engineering Systems - 2009. - №13 – c. 201-209 http://openarchive.nure.ua/handle/document/6438.

11. https://github.com/OlehLiapota/graduation-project

ПЕРЕЛІК ДЖЕРЕД ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

9. Єрохін, А. Л., Турута, О. П., Нечипоренко, А. С., Бабій, А. С. Proc. of the International Conference on Computer Sciences and Information Technologies, Lviv, Ukraine, 2017, 332-335. http://openarchive.nure.ua/handle/document/4137.

10. Synthesis of Structured Models of Computer Systems in Medical Diagnosis / N. Bilous, A. Povoroznyuk, O. Kozina // International Book Series «Information Science and Computing» Intelligent Information and Engineering Systems - 2009. - №13 – с. 201-209 http://openarchive.nure.ua/handle/document/6438.