

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

## КВАЛІФІКАЦІЙНА РОБОТА

Метод підтримки якості програмного забезпечення для діагностики захворювань нервової системи на основі машинного навчання

Назва теми

Рівень вищої освіти Другий (магістерський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр КвРІПЗ.2301116.01.08.ПЗ

Виконав студент 2 курсу, група ІПЗм-23-1

  
Підпис

Дмитро ЯМБОРКО

Ініціали, прізвище

Керівник к. техн. наук, доцент

Науковий ступінь, звання

  
Підпис

Оксана ЯШИНА

Ініціали, прізвище

Нормоконтролер к. пед. наук, доцент

  
Підпис

Оксана ОНИШКО

Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії програмного забезпечення

  
Підпис

Леонід БЕДРАТЮК

Ініціали, прізвище

9 грудня 2024 р.

Хмельницький 2024

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Другий (магістерський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Л. П. Бедратюк

01 09 2024 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Ямборку Дмитру Анатолійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод підтримки якості програмного забезпечення для діагностики захворювань нервової системи на основі машинного навчання

Керівник проекту (роботи) канд. техн. наук, доцент Яшина О.М.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 12.08.2024 р. № 104-КР

2. Строк подання студентом проекту (роботи) на кафедру 01.12.2024 р.

3. Вихідні дані до проекту (роботи) Матеріали науково-дослідної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1 Аналіз предметної області та рішень з програмного забезпечення.

2 Удосконалення методу підтримки якості програмного коду на основі автоматичного тестування

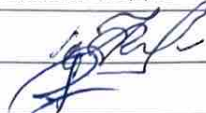
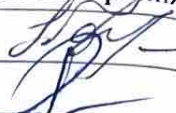
3 Архітектура програмної реалізації

4 Програмна реалізація

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Антиплагіат	Форкун Ю.В., доцент		
Нормоконтроль	Онишко О.Г., доцент		

7. Дата видачі завдання «01» вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1. Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження; визначення структури дипломної роботи	01.09 - 07.09.2024	
2. Аналіз предметної області, актуальних технологій, моделей та методів	08.09 - 25.09.2024	
3 Розробка моделей та методів вирішення завдання	26.09 - 10.10.2024	
4. Робота над науковими публікаціями	11.10 - 20.10.2024	
5. Проектування архітектури системи для вирішення задачі, розробка вимог.	11.10 - 26.10.2024	
6 Детальний опис реалізації та оцінки системи	27.10 - 15.11.2024	
7 Оформлення пояснювальної записки згідно вимог чинних стандартів	16.11 - 30.11.2024	
8 Попередній захист дипломної роботи	17.11.2024	
9 Перевірка роботи на наявність плагіату; норм контроль; брошурування пояснювальної записки; підготовка супровідних документів	01.12 - 04.12.2024	
10 Підготовка до захисту дипломної роботи	05.12 - 08.12.2024	

Студент

  
Підпис

Дмитро ЯМБОРКО  
Ініціали, прізвище

Керівник проекту (роботи)

  
Підпис

Оксана ЯШИНА  
Ініціали, прізвище

## РЕФЕРАТ

Тема дипломної роботи: «Метод підтримки якості програмного забезпечення для діагностики захворювань нервової системи на основі машинного навчання».

Автор роботи: Ямборко Дмитро Анатолійович.

Керівник роботи: Яшина Оксана Миколаївна.

Пояснювальна записка: 83 с., 7 рис., 2 дод., 25 джерел.

**ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ, ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ДІАГНОСТИКА ХВОРОБ, ГЛИБИННЕ НАВЧАННЯ.**

**Мета дослідження** - удосконалення методики використання машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

**Об'єктом дослідження** є процес використання методу машинного навчання для досягнення та підтримки якості програмного забезпечення діагностичного призначення.

**Предметом дослідження** є методи машинного навчання, що використовуються для досягнення та підтримки якості програмного забезпечення діагностичного призначення.

Враховуючи мету, предмет, а також об'єкт дослідження можна сформулювати такі завдання:

1. Здійснення аналізу предметної області досліджуваної проблеми.
2. Здійснення дослідження методів машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.
3. Удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.
4. Розроблення алгоритму роботи методу підтримки якості програмного забезпечення на основі машинного навчання.

Наукова новизна:

1. Удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.
2. Розроблено алгоритм роботи методу підтримки якості програмного забезпечення на основі машинного навчання.

Практичне значення отриманих результатів. У цій кваліфікаційній роботі показано удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення. Результатом даного дослідження став алгоритм роботи методу підтримки якості програмного забезпечення на основі машинного навчання. Це в свою чергу дозволить програмістам використовувати даний метод та алгоритм під час розробки програмного забезпечення в медичній галузі для продуктів діагностичного спрямування. Потрібність і актуальність подібних досліджень продемонстровано у перших двох розділах.

В ході проведення даного дослідження використано методи аналізу та синтезу, абстракції, порівняльні критерії.

26.11.2024



## ABSTRACT

Master's thesis: «Machine learning-based quality assurance method for software for diagnosing nervous system diseases».

Author: Dmytro Yamborko.

Head of work: Oksana Yashyna.

Master's thesis consists of: 83 pages of the general text, 7 graphics, 2 supplements, 25 literature sources.

ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, SOFTWARE QUALITY, DISEASE DIAGNOSTICS, DEEP LEARNING.

The purpose of the study is to improve the methodology of using machine learning to achieve the quality of diagnostic software.

The object of research is the process of using machine learning to achieve the quality of diagnostic software.

The subject of the study is machine learning methods used to achieve the quality of diagnostic software.

Given the purpose, subject, and object of the study, the following tasks can be formulated:

1. Analyze the subject area of the problem under study.
2. To study machine learning methods for achieving and maintaining the quality of program code for diagnostic software.
3. Improving the machine learning method for improving the quality of diagnostic software.

Scientific novelty:

1. Improving the machine learning method to improve the quality of diagnostic software.
2. The algorithm of the method of software quality support based on machine learning was developed.

Practical significance of the results. This master's thesis shows the improvement of the machine learning method to improve the quality of diagnostic software. The result of this study is an algorithm for the operation of a machine learning-based software quality assurance method. This, in turn, will allow programmers to use this method and algorithm when developing software in the medical field for diagnostic products. The necessity and relevance of such research is demonstrated in the first two chapters. .... 17

In the course of this study, the methods of analysis and synthesis, abstraction, and comparative criteria were used. .... 33

1.5. Висновки до 1-го розділу ..... 34

2. Метод підтримки якості програмного забезпечення діагностичного призначення ..... 36

26.11.24

ГМД

2.1. Консигуаційна модель підтримки програмного забезпечення діагностичного призначення ..... 36

2.2. Метод підтримки якості програмного забезпечення діагностичного призначення ..... 39

2.3. Алгоритм процесу програмного забезпечення діагностичного призначення ..... 44

2.4. Висновки до 2-го розділу ..... 48

3. Архітектура системи ..... 47

3.1. Формування та аналіз вимог програмної реалізації програмної системи ..... 49

3.2. Проектування архітектури програмної системи діагностичного призначення ..... 51

3.3. Висновки до 3-го розділу ..... 55

4. Система системи підтримки якості програмного забезпечення діагностичного призначення ..... 56

4.1. Система системи якості програмного забезпечення діагностичного призначення ..... 56

4.2. Матриця взаємодія програмного забезпечення діагностичного призначення ..... 58

4.3. Висновки до 4-го розділу ..... 62

## ЗМІСТ

Вступ.....	10
1.Теоретичний виклад досліджуваної проблеми .....	13
1.1.Аналіз предметної області.....	13
1.2.Аналіз існуючих рішень .....	17
1.3.Огляд методів вирішення проблеми.....	27
1.4.Постановка задачі.....	33
1.5.Висновки до 1-го розділу.....	34
2.Метод підтримки якості програмного забезпечення діагностичного призначення .....	36
2.1.Концептуальна модель підтримки програмного забезпечення діагностичного призначення .....	36
2.2.Метод підтримки якості програмного забезпечення діагностичного призначення .....	39
2.3.Алгоритм процесу програмного забезпечення діагностичного призначення .....	44
2.4.Висновки до 2-го розділу.....	48
3.Архітектура системи .....	47
3.1.Формування та аналіз вимог програмної реалізації програмної системи .....	49
3.2.Проектування архітектури програмної системи діагностичного призначення .....	51
3.3.Висновки до 3-го розділу.....	55
4.Оцінка системи підтримки якості програмного забезпечення діагностичного призначення .....	56
4.1.Оцінка системи якості програмного забезпечення діагностичного призначення .....	56
4.2.Метрика впровадження програмного забезпечення діагностичного призначення .....	58
4.3.Висновки до 4-го розділу.....	62

Висновки .....	64
Перелік джерел посилання .....	68
Додаток А .....	71
Додаток Б.....	73

## ВСТУП

Актуальність дослідження. Хвороби нервової системи завдають подекуди непоправної шкоди здоров'ю людини. Наприклад інсульт є основною причиною тривалої непрацездатності та смерті в усьому світі [10, 13]. Інсульт є другою за значимістю причиною смерті після серцево-судинних захворювань, що становить 10% випадків у всьому світі [9, 15]. Рентгенологи зазвичай діагностують 2 типи інсульту: ішемічний і геморагічний інсульт, використовуючи тривимірну комп'ютерну томографію (КТ) або магнітно-резонансну томографію (МРТ) і надаючи клінічну оцінку своїх результатів, обґрунтування та рекомендації щодо лікування та реабілітації.

Разом з тим, попит на програмне забезпечення зростає з кожним днем у різних сферах, зокрема й у медичній сфері, а саме діагностики різних захворювань. Розробники програмного забезпечення докладають більше зусиль для розробки та перевірки якості програмного забезпечення, а також його валідності та надійності перед його випуском. Високоякісні програмні модулі розробляються для того, щоб дозволити іншим повторно використовувати компоненти.

Щоб забезпечити якість програмного забезпечення широко використовують штучний інтелект, а саме такий його аспект як машинне навчання.

В останні роки штучний інтелект увійшов у різні сфери життя. У сфері виробництва замість людей зайнято багато роботів, що запрограмовані відповідати конкретних потреб та мають спеціально навчені алгоритми машинного навчання (МН). Взаємозв'язок між штучним інтелектом (ШІ) та розробкою програмного забезпечення (ПЗ) з кожним днем стає ближчим, хоча розрив між інтеграцією цих двох галузей все ще великий порівняно зі зв'язком між ШІ та іншими науками. Алгоритми машинного навчання нещодавно почали відігравати важливу роль у індустрії програмного забезпечення, а також у житті розробників програмного забезпечення. Ці розробники програмного забезпечення прагнули розробити програмне забезпечення на основі МН, яке допомагає їм ефективно виконувати свої

завдання [1]. Існують моделі для часткового генерування коду або оцінки вартості, зусиль і якості необхідного програмного забезпечення. Багато досліджень довели здатність методів машинного навчання надавати інформацію та вживати заходів, які прискорять процес розробки програмного забезпечення та підтримають його ефективність. Не тільки на рівні процесів, а й на рівні проекту в цілому, включаючи розрахунок зусиль, ресурсів і фінансових питань, а також часу, необхідного для випуску [2].

**Об'єктом дослідження** є процес використання методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

**Мета дослідження** - удосконалення методики використання машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

**Предметом дослідження** є методи машинного навчання, що використовуються для підтримки якості програмного забезпечення діагностичного призначення.

Враховуючи мету, предмет, а також об'єкт дослідження можна сформулювати такі завдання:

1. Здійснення аналізу предметної області досліджуваної проблеми.
2. Здійснення дослідження методів машинного навчання для досягнення та підтримки якості програмного забезпечення діагностичного призначення.
3. Удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

Наукова новизна:

1. Удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.
2. Розроблено алгоритм роботи методу підтримки якості програмного забезпечення на основі машинного навчання.

Практичне значення отриманих результатів. У цій кваліфікаційній роботі магістра показано удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення. Результатом даного

дослідження став алгоритм роботи методу підтримки якості програмного забезпечення на основі машинного навчання. Це в свою чергу дозволить програмістам використовувати даний метод та алгоритм під час розробки програмного забезпечення в медичній галузі для продуктів діагностичного спрямування. Потрібність і актуальність подібних досліджень продемонстровано у перших двох розділах.

В ході проведення даного дослідження використано методи аналізу та синтезу, абстракції, порівняльні критерії.

Відповідно до теми кваліфікаційної роботи опубліковані тези «Метод підтримки якості програмного забезпечення на основі машинного навчання» на конференції «Актуальні проблеми комп'ютерних наук (АПКН-2024)».

# 1 ТЕОРЕТИЧНИЙ ВИКЛАД ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

## 1.1 Аналіз предметної області

В сучасних умовах стрімко розвивається методологія розробки програмних систем, застосовуються гнучкі технології та нові мови програмування. Розвиток індустрії програмного забезпечення зробив питання якості програмних систем актуальною проблемою. На сьогоднішній день для визначення якості програмних систем запропоновано низку моделей якості, які визначають параметри та критерії оцінки якості [17].

Надійність програмного забезпечення є одним з ключових показників серед параметрів якості програмних систем, оскільки вона кількісно оцінює збої програмного забезпечення, які можуть вивести з ладу навіть найпотужнішу систему, гарантуючи, що програмні системи працюють коректно і не виникають непередбачувані інциденти. Зростаюча складність програмних систем, розширення кола завдань, що на них покладаються, і, як наслідок, значне збільшення обсягу та складності програмних систем зробили проблему надійності програмних систем ще більш актуальною. Суть питання полягає у виявленні основних факторів, що впливають на надійність програмних систем, демонстрації існуючих проблем у цій галузі та розробці математичних моделей для оцінювання надійності. Математичні моделі оцінюють кількість помилок, що залишилися в програмній системі до введення в експлуатацію, прогнозують час виникнення наступного збою і коли закінчиться процес тестування [17].

Питання якості програмного забезпечення загалом дуже важливе, а коли це стосується діагностики різноманітних захворювань, зокрема таких, що пов'язані із хворобами нервової системи, то тут якість підіграє ключову роль. Оскільки невірна постановка діагнозу тягне за собою і дуже погані наслідки, подекуди навіть летальні.

Неврологічні захворювання - це стани, які впливають на нервову систему. Нервова система включає головний і спинний мозок, а також усі нерви, які

розгалужуються до решти тіла. Існує багато різних типів захворювань нервової системи, кожна з яких має різні причини.

Нервова система відповідає за надсилання сигналів від мозку до решти тіла і навпаки, що є життєво важливим для регуляції багатьох функцій організму, а також для обробки та реагування на різні внутрішні фактори та фактори навколишнього середовища.

Нервова система допомагає різним частинам тіла спілкуватися одна з одною. Сигнали у нервовій системі надсилаються вздовж складної мережі нервових клітин, які називаються нейронами.

Нервова система розділена на дві різні частини: центральна нервова система (ЦНС), периферична нервова система (ПНС).

Центральна нервова система складається з головного та спинного мозку. На базовому рівні він приймає, аналізує та реагує на внутрішню інформацію та інформацію про навколишнє середовище. Це є головним комунікаційним центром людського тіла.

Периферична нервова система включає решту нервової тканини за межами головного та спинного мозку. Це допомагає передавати інформацію від ЦНС до решти тіла і навпаки. Вона в свою чергу також складається з двох частин, які включають: соматичну нервову систему (яка надсилає сигнали від ЦНС до скелетних м'язів, які дозволяють реагувати на речі оточуючого середовища) та вегетативну нервову систему, що контролює мимовільні функції організму, такі як частота серцевих скорочень, артеріальний тиск і травлення.

За даними Національної медичної бібліотеки, існує понад 600 захворювань, які можуть вплинути на нервову систему. Ці захворювання також називаються неврологічними.

Дослідження неврологічних захворювань Trusted Source у Сполучених Штатах 2020 року показало, що кількість людей із цими захворюваннями зростає з 1990 по 2017 рік. Ймовірно, це пов'язано зі старінням населення. Трьома захворюваннями, що мають найбільший вплив, були:

- інсульт;

- хвороба Альцгеймера та інші деменції;
- мігрень.

Існує багато різних типів неврологічних захворювань. Нижче вони розбиті за кожним типом разом із прикладами поширених захворювань для кожного типу захворювання.

Травми нервової системи. Можна пошкодити нервову систему. Ці типи травм зазвичай трапляються через нещасні випадки, спортивні травми чи акти насильства.

Пошкодження ЦНС можуть включати черепно-мозкову травму та травму спинного мозку.

Черепно-мозкова травма може призвести до таких фізичних симптомів, як:

- запаморочення;
- головний біль;
- слабкість;
- втрата зору;
- судоми;
- втрата свідомості.

Це також іноді може впливати на пізнання, пам'ять і настрої. Травми спинного мозку можуть викликати такі симптоми, як:

- біль;
- оніміння і поколювання;
- м'язова слабкість або параліч.

Також можуть бути пошкоджені нерви ПНС. Це може статися, коли нерв розтягується, стискається, запалюється або розривається. Симптоми можуть включати:

- невропатичний біль;
- оніміння і поколювання;
- м'язову слабкість або параліч.

Також до хвороб нервової системи відносяться цереброваскулярні захворювання. Цереброваскулярна хвороба - це порушення кровотоку в мозку. Це може статися, коли у мозку крововилив або коли мозок не отримує достатньо збагаченої киснем крові.

Інсульт є поширеним типом цереброваскулярного захворювання. Це може бути спричинено перериванням кровотоку в ділянці мозку (ішемічний інсульт), наприклад через тромб, або крововиливом у мозок (геморагічний інсульт).

Згідно із інформацією Центру з контролю та профілактики захворювань (CDC), інсульт є основною причиною смерті в Сполучених Штатах. Насправді кожна шоста смерть від серцево-судинних захворювань у 2020 році сталася через інсульт.

Існують і інші види цереброваскулярних захворювань. Ось кілька прикладів:

- аневризма головного мозку;
- судинні мальформації;
- внутрішньочерепний стеноз, звуження артерій у мозку.

Нейродегенеративні захворювання - це коли нервові клітини перестають працювати належним чином або починають відмирати. У деяких випадках це може статися через накопичення запальних клітин або аномальних білків у мозку. Основна причина цих змін невідома.

Ці стани зазвичай прогресують і це означає, що вони продовжують погіршуватися з часом.

Деякі приклади нейрогенеративних захворювань включають:

- хвороба Альцгеймера та інші види деменції;
- хвороба Паркінсона;
- бічний аміотрофічний склероз (БАС).

Також дуже поширеними є головні болі і більшість людей відчувають їх час від часу. Але коли головні болі сильні, виникають часто або постійні, то це може свідчити про мігрень або наявність якоїсь патології кровопостачання чи головного мозку.

Взагалі кажучи, головний біль виникає, коли чутливі до болю нерви реагують на тригер, посилюючи біль.

## 1.2 Аналіз існуючих рішень

Машинне навчання (МН) - це область інформаційних технологій, що використовує дані та алгоритми для імітації способу навчання людей штучним інтелектом із поступовим підвищенням його точності. Машинне навчання поділяється на три частини.

1. Функція прийняття рішень. Оскільки алгоритми машинного навчання передбачають в основному використання для того аби здійснювати прогнозування або класифікації, то використовуючи певні вхідні дані (вони можуть бути позначеними чи не позначеними) алгоритм буде оцінювати шаблон в даних.

2. Функція помилок. Це частина, що здійснює оцінювання прогнозу моделі на основі вже знайомих прикладів, зробивши порівняння для оцінки точності моделі.

3. Частина, що відповідає за процес оптимізації моделі. Здійснюється коригування вагових коефіцієнтів для зменшення розбіжності між знайомим прикладом та оцінкою моделі у випадку, якщо модель може краще відповідати точкам даних у навчальному наборі. Алгоритм повторюватиме цей ітеративний процес «оцінки та оптимізації», при цьому здійснюючи автономне оновлення ваг аж до досягнення порогу точності.

Дуже часто глибоке навчання та машинне навчання плутають та використовують взаємозамінюючи ці означення. Однак, слід звернути увагу на те, що між ними існують певні відмінності.

До поняття штучного інтелекту відносять і машинне навчання, і глибоке навчання і нейронні мережі. Однак, слід зауважити, що нейронні мережі відносяться до галузі машинного навчання, в той час, як глибоке навчання – це підобласть нейронних мереж. Відмінність глибокого та машинного навчання

полягає у способі навчання алгоритму. Глибоке навчання широко використовує так зване контрольоване навчання (мічені набори даних) з метою інформування свого алгоритму. Разом з тим даний вид навчання може не потребувати обов'язкового міченого набору даних.

Сам безпосередній процес глибокого навчання може отримувати текст чи зображення або будь-які неструктуровані дані необробленими. Глибоке навчання здатне здійснювати автоматичне визначення набору ознак, що відрізняють різні категорії даних одна від одної. Завдяки цьому моменту процес навчання може обходитись без участі людини та працювати із великими об'ємами даних.

Під час проведення машинного навчання у класичному розумінні (неглибокого навчання) сам процес відбувається з участю людини. Для цього експерти здійснюють визначення набору функцій для розуміння як відрізняються вхідні дані. Щоб вивчити це питання при контрольованому навчанні необхідно, щоб дані були структурованими.

Будова нейронних мереж, які ще називають штучними нейронними мережами (ШНМ) містить вузлові рівні: вхідний, певну кількість прихованих рівнів, вихідний рівень. Кожен такий вузол (штучний нейрон) має з'єднання з іншим нейроном, при цьому маючи відповідну вагу та поріг. Коли на виході якогось вузла (нейрона) відбувається перевищення вказаного порогового значення, то відбувається активація цього вузла із надсиланням даних на наступний рівень мережі.

Кількість шарів у нейромережі показує ту саму «глибину» у глибокому навчанні. У випадку, якщо нейромережа має у своїй будові більше ніж три рівні, що включають вхідні та вихідні дані, то такий алгоритм називають алгоритмом глибокого навчання (глибокою нейромережею). Якщо нейромережа складається тільки з трьох шарів, то вона називається базовою нейронною мережею. Загалом, глибоке навчання та нейронні мережі впливають на розвиток у таких областях, як комп'ютерна діагностика, обробка природної мови, розпізнавання мовлення тощо.

З розвитком технології машинного навчання це, безумовно, полегшило наше життя. Однак впровадження машинного навчання в бізнесі також викликало низку етичних проблем щодо технологій штучного інтелекту (ШІ).

Деякі з них включають:

1. Технологічну особливість.
2. Вплив штучного інтелекту на робочі місця.
3. Конфіденційність
4. Відповідальність.

Хоча ця тема привертає велику увагу громадськості, багатьох дослідників не хвилює ідея того, що ШІ перевершить людський інтелект у найближчому майбутньому. Технологічну унікальність також називають сильним ШІ або суперінтелектом. Філософ Нік Бострум визначає суперінтелект як «будь-який інтелект, який значно перевершує найкращий людський мозок практично в усіх сферах, включаючи наукову творчість, загальну мудрість і соціальні навички». Незважаючи на те, що суперінтелект не є неминучим у суспільстві, його ідея викликає деякі цікаві питання, оскільки ми розглядаємо використання автономних систем, таких як безпілотні автомобілі. Нереалістично думати, що безпілотний автомобіль ніколи не потрапить у аварію, але хто несе відповідальність за таких обставин? Чи слід нам усе ще розробляти автономні транспортні засоби, чи ми обмежимо цю технологію напівавтономними транспортними засобами, які допомагають людям безпечно керувати автомобілем? Присяжні все ще не знають про це, але це типи етичних дебатів, які виникають у міру розвитку нових інноваційних технологій штучного інтелекту.

2. Незважаючи на те, що громадське сприйняття штучного інтелекту зосереджено навколо втрати робочих місць, це занепокоєння, ймовірно, слід переосмислити. З кожною революційною новою технологією ми бачимо, що ринковий попит на конкретні посади змінюється. Наприклад, коли ми дивимося на автомобільну промисловість, багато виробників, як-от GM, зосереджуються на виробництві електромобілів, щоб відповідати екологічним ініціативам. Енергетична галузь не зникає, але джерело енергії переходить від економії палива

до електроенергії. Подібним чином штучний інтелект перенесе попит на робочі місця в інші сфери. Потрібні будуть люди, які допоможуть керувати системами ШІ. Все ще будуть потрібні люди для вирішення складніших проблем у галузях, на які найімовірніше вплинуть зміни попиту на роботу, наприклад обслуговування клієнтів. Найбільшою проблемою зі штучним інтелектом і його впливом на ринок праці буде допомога людям у переході на нові ролі, які користуються попитом.

3. Конфіденційність, як правило, обговорюється в контексті конфіденційності даних, захисту даних і безпеки даних. Ці побоювання дозволили політикам зробити більше кроків в останні роки. Наприклад, у 2016 році було створено законодавство GDPR для захисту персональних даних людей у Європейському Союзі та Європейській економічній зоні, надаючи людям більше контролю над своїми даними. У Сполучених Штатах окремі штати розробляють політику, наприклад Каліфорнійський закон про конфіденційність споживачів, який був запроваджений у 2018 році та вимагає від компаній інформувати споживачів про збір їхніх даних. Подібне законодавство змусило компанії переглянути спосіб зберігання та використання особистої інформації. У результаті інвестиції в безпеку стають все більш пріоритетними для компаній, оскільки вони прагнуть усунути будь-які вразливості та можливості для стеження, злому та кібератак. Упередженість і дискримінація. Випадки упередженості та дискримінації в ряді систем машинного навчання викликали багато етичних питань щодо використання штучного інтелекту. Як людина може захиститися від 25 упередженості та дискримінації, коли самі навчальні дані можуть бути створені упередженими людськими процесами. Хоча компанії, як правило, мають добрі наміри щодо своїх зусиль щодо автоматизації, існують деякі непередбачені наслідки впровадження штучного інтелекту в практику найму. Прагнучи автоматизувати та спростити процес, Amazon ненавмисно дискримінувала кандидатів на технічні посади за статтю, і зрештою компанії довелося скасувати проект. Упередженість і дискримінація також не обмежуються функцією людських ресурсів; їх можна знайти в ряді додатків від програмного забезпечення для розпізнавання обличчя до алгоритмів соціальних мереж. Оскільки підприємства все більше усвідомлюють ризики,

пов'язані зі штучним інтелектом, вони також стають більш активними в обговоренні етики та цінностей штучного інтелекту.

Наприклад, IBM припинила випуск продуктів розпізнавання та аналізу обличчя загального призначення. Генеральний директор IBM Арвінд Крішна написав: «IBM рішуче виступає проти використання будь-якої технології, включно з технологією розпізнавання обличчя, що пропонується іншими постачальниками, для масового стеження, расового профілювання, порушення основних прав і свобод людини або з будь-якою іншою несумісною метою. з нашими цінностями та принципами довіри та прозорості» [3]. Оскільки немає значного законодавства, яке б регулювало практику штучного інтелекту, немає реального механізму примусу, який би гарантував, що практикується етичний штучний інтелект. Нинішні стимули для компаній бути етичними – це негативні наслідки неетичної системи ШІ. Щоб заповнити цю прогалину, етичні рамки з'явилися як частина співпраці між етиками та дослідниками для управління створенням і розповсюдженням моделей ШІ в суспільстві. Однак, на даний момент вони служать лише орієнтиром. Деякі показують, що поєднання розподіленої відповідальності та відсутності передбачення потенційних наслідків не сприяє запобіганню шкоди суспільству.

Машинне навчання має значний вплив на те, як програмне забезпечення, щоб воно могло йти в ногу зі змінами в бізнесі. Машинне навчання настільки важливе, тому що воно допомагає вам використовувати дані для створення бізнесправил та логіки. У традиційних моделях розробки програмного забезпечення програмісти писали логіку на основі поточного стану бізнесу а потім додавали відповідні дані. Однак, зміни в бізнесі стали нормою. Практично неможливо передбачити, які зміни трансформують ринок. Цінність машинного навчання полягає в тому, що воно дозволяє постійно вчитися на даних і прогнозувати майбутнє. Цей потужний набір алгоритмів і моделей використовується в різних галузях для вдосконалення процесів та отримання уявлення про закономірності та аномалії в даних. Але машинне навчання - це не самотня робота, це командний процес. який вимагає співпраці науковців, інженерів даних, бізнес-аналітиків та бізнес-лідерів для співпраці. Машинне навчання - це форма ШІ, яка дозволяє системі вчитися на

основі даних, а не за допомогою явного програмування. Однак, машинне навчання не є простим процесом. Машинне навчання використовує різноманітні алгоритми, які ітеративно навчаються на даних, щоб покращувати, описувати дані та прогнозувати результати. Коли алгоритми отримують навчальні дані, вони можуть створювати більш точні моделі на основі цих даних.

Модель машинного навчання - це результат, отриманий під час навчання алгоритму машинного навчання за допомогою даних. Після навчання, коли надаєте моделі вхідні дані, ви отримаєте вихідні дані. Наприклад, алгоритм прогнозування створює прогнозуючу модель. Потім, коли прогнозуюча модель отримує дані, то результатом буде відповідний прогноз, що ґрунтується на даних, на яких навчалася модель. Саме тому машинне навчання в сучасних реаліях відіграє досить важливу роль для розробки аналітичних моделей, зокрема й таких, що стосуються діагностичних систем.

Методи машинного навчання класифікують таким чином: контрольоване машинне навчання, неконтрольоване машинне навчання (без втручання людини), напівконтрольоване машинне навчання (із залученням людського фактору).

Контрольоване навчанням (кероване машинне навчання) – це такий метод машинного навчання при якому відбувається визначення позначених наборів даних у навчанні алгоритмів для класифікації даних чи точного прогнозування результатів. Після того, як вхідні дані поступають у модель відбувається коригування нею відповідної ваги, до тих пір, поки не буде встановлено належним чином. Це здійснюється в рамках процесу так званої перехресної перевірки аби було переконання, що модель не переобладнана або не підігнана.

Контрольоване навчання слугує для вирішення різного роду реальних проблем в масштабі, до прикладу, здійснити класифікацію за певним параметром.

Кероване навчання використовує такі методи:

- нейронні мережі,
- наївну байєсівську регресію,
- логістичну регресію,
- лінійну регресію,

- випадкові ліси,
- машину опорних векторів.

Під час неконтрольоване машинного навчання використовуються алгоритми для здійснення аналізу та кластеризації наборів даних, що непозначені. Такі алгоритми працюють без людини-експерта, виявляючи приховані шаблони чи групи даних. Оскільки цей метод може здійснювати виявлення подібностей та відмінностей під час аналізу інформації, то його доцільно використовувати при аналізі даних дослідницького характеру, стратегіях перехресних продажів, таргетування клієнтів, для розпізнавання зображень чи шаблонів. Також є можливим його використання, щоб зменшити кількість функцій у моделі за допомогою процесу зменшення розмірності.

Найбільш поширеними та використовуваними підходами є: аналіз головних компонентів та сингулярне розкладання. Неконтрольоване навчання також використовує такі алгоритми: включають нейронних мереж, кластеризації k-середніх, ймовірнісних методів кластеризації.

Напівконтрольоване навчання – в процесі навчання використовується зменшений набір даних, що мають мітки. Це дозволяє здійснювати керування класифікацією та виділенням ознак із більшого набору даних без міток. Напівконтрольоване навчання може вирішити проблему відсутності достатньої кількості позначених даних для алгоритму контрольованого навчання. Це також допомагає, якщо надто дорого позначити достатню кількість даних.

Машинне навчання з підкріпленням – це така модель машинного навчання, де навчання алгоритму відбувається без допомоги вибіркового даних. Така модель здійснює навчання в процесі на основі проб і помилок. При цьому послідовність тих результатів, що були успішними посилюється. Це в свою чергу допомагає у знаходженні та розробці найкращої рекомендації чи політики відносно конкретної проблеми. Наприклад, навчання з підкріпленням використовується для того, щоб дізнатися, коли спробувати відповісти на запитання.

Зазвичай використовується ряд алгоритмів машинного навчання. До них належать:

- нейронні мережі;
- лінійна регресія;
- логістична регресія;
- кластеризація;
- дерева рішень;
- випадкові ліси.

Нейронні мережі, як уже зазначалось здійснюють імітацію роботи мозку та мають у своїй будові пов'язаних між собою вузлів. Нейронні мережі працюють таким чином, що за їх допомогою доцільно здійснювати розпізнавання шаблонів, обробку та переклад природної мови, а також розпізнавання зображень, мовлення та створення зображень.

Лінійна регресія - даний алгоритм використовується, щоб здійснити прогнозування числових значень на основі лінійної залежності між різними значеннями. Наприклад, даний алгоритм доцільно застосовувати, щоб спрогнозувати ціни на житло, беручи до уваги історичні дані конкретного регіону.

Логістична регресія – це такий контрольований алгоритм навчання (контрольований), який здійснює прогнозування для категоричних відповідей, що змінюються (відповіді «так/ні» на запитання). Даний тип алгоритму доцільно застосовувати для класифікації спаму, а також контролю якості на певній виробничій лінії.

Кластеризація застосовує в своїй основі неконтрольоване навчання, тобто алгоритми кластеризації ідентифікують шаблони в даних, з можливістю подальшого групування. За допомогою даного методу можна дослідити дані та визначити відмінності між їхніми елементами, що залишились би поза увагою людини.

Дерева рішень – ці алгоритми доцільно застосовувати щоб спрогнозувати числові значення (регресія), а також здійснити класифікацію даних за певними категоріями. Цей вид алгоритму використовує розгалужену послідовність пов'язаних рішень. Такі рішення подаються у вигляді деревоподібної діаграми.

Даний алгоритм дуже легкий у перевірці, що є однією із його переваг. І це відрізняє його від методу чорного ящика нейромережі.

Випадковий ліс (Random forest) – у цьому методі закладено такий алгоритм, що може передбачати певні значення чи категорії, при цьому відбувається поєднання результатів з декількох дерев рішень.

Залежно від бюджету, необхідної швидкості та точності, кожен тип алгоритму - керований, неконтрольований, напівконтрольований або підкріплений - має свої переваги та недоліки. Наприклад, алгоритми дерева рішень використовуються як для прогнозування числових значень (проблеми регресії), так і для класифікації даних за категоріями. Дереву рішень використовують розгалужену послідовність пов'язаних рішень, які можна представити за допомогою деревоподібної діаграми. Основна перевага дерев рішень полягає в тому, що їх легше перевірити, ніж нейронні мережі. Однак, що вони можуть бути більш нестабільними, ніж інші засоби прогнозування рішень.

Загалом у машинного навчання є багато переваг, які компанії можуть використати для підвищення ефективності. До них відноситься машинне навчання, що визначає закономірності та тенденції у величезних обсягах даних, які люди можуть взагалі не помітити. І цей аналіз вимагає невеликого втручання людини: просто введіть набір даних, який вас цікавить, і дозвольте системі машинного навчання зібрати й удосконалити власні алгоритми, які з часом будуть постійно вдосконалюватись із збільшенням кількості введених даних. Клієнти та користувачі можуть насолоджуватись більш персоналізованим досвідом, оскільки модель дізнається більше з кожною зустріччю з цією людиною. З іншого боку, машинне навчання потребує великих навчальних наборів даних, які є точними та неупередженими. Збір достатньої кількості даних і наявність достатньо надійної системи для її роботи також може призвести до виснаження ресурсів.

Машинне навчання також може бути схильним до помилок, залежно від вхідних даних. З надто малою вибіркою система може виробити абсолютно логічний алгоритм, який є абсолютно неправильним або вводить в оману. Щоб

уникнути марної витрати бюджету або невдоволення клієнтів, організації повинні діяти на основі відповідей лише тоді, коли є висока впевненість у результатах.

Випадки використання реального машинного навчання: Розпізнавання мовлення: це також відоме як автоматичне розпізнавання мовлення (ASR), комп'ютерне розпізнавання мовлення або перетворення мови в текст, і це можливість, яка використовує обробку природної мови (NLP) для 32 перекладу людського мовлення в письмовий формат. Багато мобільних пристроїв включають у свої системи розпізнавання мовлення для здійснення голосового пошуку (наприклад, Siri) або покращення доступності для текстових повідомлень. Обслуговування клієнтів: онлайн-чат-боти замінюють людей-агентів на шляху клієнта, змінюючи наше уявлення про залучення клієнтів на веб-сайтах і в соціальних мережах. Чат-боти відповідають на поширені запитання (FAQ) про такі теми, як доставка, або надають персоналізовані поради, перехресні продажі продуктів або пропонують розміри для користувачів. Приклади включають віртуальних агентів на сайтах електронної комерції; боти для обміну повідомленнями, використовуючи Slack і Facebook Messenger; і завдання, які зазвичай виконують віртуальні та голосові помічники. Комп'ютерний зір: ця технологія штучного інтелекту дозволяє комп'ютерам отримувати значущу інформацію з цифрових зображень, відео та інших візуальних даних, а потім виконувати відповідні дії. Завдяки згортковим нейронним мережам комп'ютерний зір має застосування для позначення фотографій у соціальних мережах, радіологічної візуалізації в охороні здоров'я та безпілотних автомобілів в автомобільній промисловості. Механізми рекомендацій: використовуючи минулі дані про використання, алгоритми ШІ можуть допомогти виявити тенденції даних, які можна використовувати для розробки ефективніших стратегій перехресних продажів. Механізми рекомендацій використовуються роздрібними торговцями в Інтернеті, щоб давати клієнтам відповідні рекомендації щодо продуктів під час оформлення замовлення.

Роботизована автоматизація процесів (RPA): також відома як програмна робототехніка, RPA використовує інтелектуальні технології автоматизації для

виконання повторюваних ручних завдань. Автоматизована торгівля акціями: створені для оптимізації портфелів акцій високочастотні торгові платформи на основі ШІ здійснюють тисячі або навіть мільйони угод на день без втручання людини.

Банки та інші фінансові установи можуть використовувати машинне навчання для виявлення підозрілих транзакцій. Контрольоване навчання може навчити модель, використовуючи інформацію про відомі шахрайські транзакції. Виявлення аномалій може ідентифікувати транзакції, які виглядають нетиповими та потребують подальшого дослідження.

### 1.3 Огляд методів вирішення проблеми

Однією з головних перешкод для внеску в сучасні методи ШІ для діагностики інсульту є отримання доступу до загальнодоступних наборів даних. Загальнодоступних 2D і 3D клінічно анотованих наборів медичних зображень інсульту з метаданими експертних звітів дуже мало. У деяких сферах конкуренції з медичної візуалізації, протягом кількох років було випущено обмежені загальнодоступні 2D та 2,5D набори даних інсульту з наборами КТ та МРТ зображень із клінічними анотаціями для кожного пацієнта. Однак, ці набори даних про інсульт не забезпечують повний набір 3D-зображень інсульту, що ускладнює розробку сучасних моделей сегментації уражень, які добре працюватимуть у клінічних умовах. Крім того, ці набори даних про інсульт не забезпечують повного набору клінічно анотованих метаданих на тривимірний воксель, таких як опис звіту

рентгенолога, групування уражень після інсульту за загальними характеристиками: судинні території, демографічні дані, розміри ураження тощо [1].

В сучасних реаліях використовуються досить широкий набір можливостей для діагностики інсульту, наприклад використання комп'ютерного зору.

Комп'ютерний зір у діагностиці інсульту, наприклад StrokeSave - це платформа, за допомогою якої користувачі можуть самостійно діагностувати поширеність інсульту. Мобільний додаток постійно оновлюється даними про частоту серцевих скорочень, артеріальний тиск і рівень кисню в крові з датчиків на зап'ясті пацієнта. Коли ці вимірювання досягають порогу можливого інсульту, пацієнт робить зображення обличчя та запис голосу для виявлення паралічу, пов'язаного з інсультом. Спеціально розроблений об'єктив, приєднаний до камери телефону, потім робить зображення сітківки для моделі глибокого навчання для класифікації на основі наявності ретинопатії та надсилає комплексний діагноз. Модель глибокого навчання, яка складається з RNN, навченого на 100 аудіофайлах із невиразним голосом, SVM, навченого на 410 точках судинних даних, і CNN, навченого на 520 зображеннях ретинопатії, досягла цілісної точності 95,0 відсотків під час перевірки на 327 зразках. Це значення перевищує точність клінічного обстеження, яка становить від 40 до 89 відсотків, що додатково демонструє життєво важливу корисність такого медичного пристрою. Завдяки цій автоматизованій платформі користувачі отримують ефективну, високоточну діагностику без професійної медичної допомоги, що революціонує медичну діагностику КС і потенційно рятує мільйони життів.

3D МРТ реконструкція для 3D атласу артеріальних територій мозку [16]. Коли здійснювався пошук під час дослідження в мережі інтернет публікацій про тривимірну реконструкцію інсульту за допомогою людини, то було знайдено знайшли один результат публікації.

Також нижче описано методи та засоби відповідно до аналізу сучасних дослідницьких праць.

Наприклад, Liu та ін. використовують порівняльний алгоритм ітераційної реконструкції ML, відомий як алгоритм максимізації очікування (EM), для

виконання 3D-реконструкції МРТ інсульту, щоб скласти свій 3D-атлас мозку, оскільки вони мають достатньо даних для навчання свого алгоритму ML із великим набором даних 3D МРТ клінічного інсульту [23]. Однак, інші використовують більш традиційні методи CV, не засновані на штучному інтелекті, такі як контури для малювання уражень і картування кожного з них на зрізах МРТ для виконання 3D-реконструкції інсульту, оскільки вони мали доступ до невеликого набору даних МРТ інсульту ATLAS з менше ніж 200 зображень [15] порівняно з Liu та ін., які мали доступ до свого великого приватного клінічного набору даних 3D МРТ інсульту з понад 2888 зображень [23].

У той час як Ліу та інші змогли побудувати глибокий тривимірний атлас мозку з ураженнями ішемічного інсульту, нанесеними на карту їхніх конкретних артеріальних територій (MCA, задня церебральна артерія (PCA), вертебробазиллярна (VB), ACA) [16], N. Palacios-Quesan та інші змогли побудувати частковий 3D атлас мозку з інсультними ураженнями, нанесеними на артеріальну територію SMA [15]. Основна відмінність між цими двома результатами для 3D-реконструкції інсульту для 3D-атласу мозку полягала в тому, що набори даних про інсульт, до яких мали доступ як Liu et al [23], так і N. Palacios-Quesan et al, полягали у величезному коефіцієнті прогалин у розмірі набору даних [15].

У сегментації уражень після інсульту більшість підходів походять із 2D, зокрема UNet [11], [14]. 2D UNet - це двовимірний варіант CNN, удосконалений декількома шарами згортки для зменшення та підвищення дискретизації зображень, що проходять через мережу кодера-декодера, який вчиться класифікувати та сегментувати або відстежувати великі об'єкти, такі як кола, прямокутники, пошкодження тощо. Архітектури UNet можуть виконувати не лише 2D-сегментацію уражень, але існують і 3D-варіанти UNet, які можуть виконувати 3D-сегментацію уражень. UNet, як правило, чудово сегментує великі пошкодження на зображеннях, але має тенденцію пропускати менші пошкодження, оскільки згорткове ядро менше, ніж фактичне зображення, через що він пропускає глобальну інформацію [14]. Хоча UNet використовувався для 3D-сегментації уражень з іншими наборами даних медичної візуалізації, як правило, оскільки більшість

наборів даних інсульту не є повністю 3D, дослідники, які займаються великими проблемами медичної візуалізації, переважно розробили свої варіанти UNet для 2D-сегментації. Однак, ці двовимірні варіанти UNet пропускають усю контекстну інформацію 3D мозку [14]. Ліу та інші [14] створили найбільший обмежений загальнодоступний набір даних 3D MPT інсульту з клінічними анотаціями з 2888 зображеннями, кожне з яких пов'язано з клінічними метаданими [14].

Цими ж дослідниками було розроблено модель TensorFlow Keras 3D DAGMNet для свого набору даних 3D MPT інсульту ICPSR 38464 на основі UNet3+, яка вдосконалена подвійними воротами уваги, тому модель може навчитися сегментувати ураження різних розмірів, об'ємів, місць розташування та отримати повне 3D-зображення мозку. Перед остаточним прогнозуванням виходу, коли 3D DAGMNet декодує функції, щоб навчитися відстежувати різні ураження та зливати ці знання в один вихід, Лю також додає рівень регуляризації L1, щоб запобігти створенню моделлю хибнопозитивних прогнозів [14]. Лю та ін. також додали масштабовану експоненціальну лінійну одиницю (SeLU) і шари пакетної нормалізації до 3D DAGMNet, щоб запобігти проблемі зникнення градієнта, яка є загальною для 3D-мереж [14]. Дані науковці також здійснили перевірку здатності 3D DAGMNet виконувати сегментацію уражень ішемічного інсульту на основі найсучасніших моделей, таких як DeepMedic, UNet, FCN тощо, використовуючи оцінку кубиків, точність, чутливість та інші показники ефективності [14]. Ці тести для 3D DAGMNet порівняно з сучасними моделями також проводилися з групуванням інсультних уражень за загальними характеристиками, такими як демографічні дані, судинна територія, розмір ураження, півкуля мозку, початок симптомів, виробник MPT тощо, щоб виявити, що 3D DAGMNet все-таки працює на зверху [14].

Однак, Ліу та інші попередньо обробили свій набір даних про інсульт, щоб включити лише випадки ішемічного інсульту, тому 3D DAGMNet навчено лише сегментації уражень ішемічного інсульту [14]. Таким чином, 3D Лю та інших не враховує вивчення сегментації уражень для випадків геморагічного інсульту в їх

наборі даних 3D МРТ інсульту. Одним із підходів до розширення 3D для вивчення сегментації уражень геморагічного інсульту є використання перенесення навчання.

Методи НЛП для медичної візуалізації з клінічними анотаціями, що застосовувалися в різних сферах діагностики захворювань, оскільки вони мають достатню кількість клінічно анотованих метаданих на медичне зображення [22], [24]. Однак ці методи НЛП, такі як генерація підписів до зображень CapGAN Л. Сонга та ін. [22], генерація звітів HRNN К. Іня та ін. [24] і пошук зображень Дж. Рамоса та ін. не застосовувалися до медичної візуалізації для діагностики інсульту, оскільки відсутня велика кількість клінічно анотованих метаданих для МРТ інсульту в загальнодоступних наборах даних [14]. Тому у даному дослідженні здійснено детальніше заглиблення в деякі з цих методів НЛП для діагностики хвороб за допомогою медичного зображення.

Моделі DNN із субтитрами до медичних зображень важливі для діагностики інсульту штучним інтелектом, оскільки вони можуть заповнювати зображення субтитрами щодо приміток нейрорадіолога про ураження. Зазвичай CNN, рекурентні нейронні мережі (RNN) і моделі UNet застосовуються для субтитрів медичних зображень, але однією з їхніх труднощів є перевірка створених субтитрів зображень за допомогою експертних метаданих субтитрів клінічних зображень через те, що вони не застосовували навчання з підкріпленням (RL) [22]. Тому L. Song та інші запропонували CapGAN для подолання цієї труднощі, яка є поєднанням загальної змагальної мережі.

Генератор діагностичних звітів на основі LSTM здатний отримувати характеристики із зображень, наприклад рентгенівських знімків грудної клітки, для класифікації та виявлення уражень, одночасно створюючи підписи до клінічних зображень рентгенолога для кожного зображення [22].

Дискримінатор на базі GRU L. Song та інших забезпечує формування та отримання високоякісних підписів до зображень медичного діагнозу [22]. Оцінювач клінічної мови L. Song та інших на базі RL співпрацює з дискримінатором, щоб підтвердити, що стиль клінічної мови діагностичних звітів узгоджується з тим, як радіологи написали б їх, що призводить до більш точного

опису клінічних зображень, як наприклад, інформація про патологію легень [10]. L. Song та ін. виявили, що лише кілька інших дослідників Ю та ін. SeqGAN та Чен та ін. запропонували підхід, подібний до CapGAN, але для природних субтитрів зображень, де використовуються GAN та RL [22]. Однак, цим небагатьом дослідникам, які застосовували підхід GAN і RL, подібний до CapGAN, не вдалося пристосувати його для субтитрів до медичних зображень [22]. Таким чином, L. Song та ін. є потенційно першою командою, яка застосувала свій підхід CapGAN до субтитрів медичних зображень, розширюючи їхній підхід для обробки низької роздільної здатності та розмитих меж між різними органами та тканинами [22]. L. Song та інші оцінили свій CapGAN для медичних зображень із субтитрами за допомогою показників BiLingual Evaluation Understudy (BLEU) порівняно з порівняльними моделями DNN на двох загальнодоступних наборах даних: рентгенівські зображення грудної клітки Open-i із пов'язаним набором метаданих радіологічного звіту та 3 органи LGK (печінка, жовчного міхура, нирок) ультразвукових зображень із набором даних звітів про діагностику [22].

Також у медичній практиці використовуються моделі DNN для створення медичних звітів, що є важливою функцією діагностики інсульту штучним інтелектом, оскільки вони можуть заощадити нейрорадіологам час на написання, створивши шаблон клінічного звіту на основі результатів моделі після аналізу зображення.

Таким чином, нейрорадіолог потім перегляне шаблон клінічного звіту моделі з будь-якими граматичними виправленнями, надлишковими реченнями та ключовими відомостями про ураження, які модель могла пропустити. Генерацію медичних звітів можна розглядати як область генерації клінічних метаданих із пов'язаними медичними зображеннями на основі відображених подібностей.

Створення субтитрів до медичних зображень. Подібні методи штучного інтелекту від субтитрів до медичних зображень за допомогою CNN, RNN, мереж кодерів-декодерів, GAN і RL можна застосовувати для створення медичних звітів. Основна відмінність між субтитрами до медичних зображень і створенням медичного звіту полягає в тому, що замість кількох фраз про аналіз уражень на

зображенні рентгенологами генерується кілька речень, які складають один або кілька абзаців.

Оскільки є декілька речень, то відстеження тем на речення за допомогою RNN і генерування речень для цих тем за допомогою RNN є надзвичайно важливим [24]. Крім того, для створення речень, які добре синхронізуються з високою точністю та якістю, подібною до радіологів, RL має важливе значення та допомагає йому порівнювати створений звіт із версією звіту радіолога.

С. Yin та інші покращують ієрархічну RNN (HRNN), відому для створення детальних медичних звітів, оновлюючи механізм відповідності тем, контексту та семантичної уваги, тому вектори тем і згенеровані речення є більш різноманітними замість схожих речень [22]. С. Yin та ін. також покращують точність і надійність DenseNet HRNN шляхом заміни глобального об'єднання функцій на об'єднання глобальних міток у мультикласифікації відхилень [22]. Оновлена С. Yin та ін. HRNN може виявляти ураження та створювати різноманітні медичні звіти [22]. С. Yin та ін. оцінили свою оновлену HRNN порівняно з порівняльними моделями DNN, використовуючи метрики BLEU, CIDEr, ROUGE-L і METEOR на 2 загальнодоступних наборах даних: набір даних пари рентгенівського медичного зображення IU та пари даних природного зображення [22]. Оновлений С. Yin та ін. HRNN показує найкращі результати в більшості показників, крім CIDEr, порівняно з CNN-RNN, LRCN, Soft ATT, ATT-RK і HRGR-AGR Agent [22]. Перероблена модель HRNN С. Yin та інших також здатна створювати щонайбільше 6 речень довжиною менше 30 слів, що більше, ніж більшість традиційних моделей створення звітів [22]. Проте ще одне вдосконалення, яке можна застосувати до HRNN С. Yin та ін., полягає в тому, щоб включити GAN і RL, щоб допомогти покращити точність і якість різноманітних речень, створених у клінічному звіті [22], [24].

#### 1.4 Постановка задачі

На початку роботи щодо дослідження було встановлено таку мету: удосконалення методики використання машинного навчання для підтримки якості програмного забезпечення діагностичного призначення. Відповідно до визначеної мети виділено наступні завдання:

1. Здійняти аналіз предметної області, а саме програмне забезпечення діагностичного призначення.

2. Провести аналіз тих методів, що вже існують в реальних умовах.

3. Здійснення дослідження методів машинного навчання для досягнення та підтримки якості програмного забезпечення діагностичного призначення.

4. Удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

Отже, в даній кваліфікаційній роботі має бути проведено удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення. Результати даного дослідження можуть використовуватись розробниками під час написання коду для розробки програмного забезпечення діагностичного спрямування із забезпеченням підтримки якості відповідно до стандартів.

## 1.5 Висновки до 1-го розділу

У першому розділі було здійснено аналіз предметної області. Результатом цього аналітичного матеріалу стали стали уявлення про те, що таке машинне навчання та як його можна використовувати під час розробки програмного забезпечення діагностичного призначення.

Також здійснено аналіз методів та засобів машинного навчання та їх використання відповідно до предметної області. В даному дослідженні предметною областю є медична галузь, а саме діагностування хвороб нервової системи.

Було визначено мету даного дослідження, що полягає в удосконаленні методики використання машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

Відповідно до мети були встановлені такі задачі:

1. Здійнити аналіз предметної області, а саме програмне забезпечення діагностичного призначення.

2. Провести аналіз тих методів, що вже існують в реальних умовах.

3. Здійснення дослідження методів машинного навчання для досягнення та підтримки якості програмного забезпечення діагностичного призначення.

4. Удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

В переважній більшості випадків використовуються такі алгоритми машинного навчання:

- нейронні мережі;
- лінійна регресія;
- логістична регресія;
- кластеризація;
- дерева рішень;
- випадкові ліси.

Кожен із цих методів має свої переваги та недоліки і може використовуватись відповідно до тих задач, що потребують виконання та розв'язку.

## 2 МЕТОД ПІДТРИМКИ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДІАГНОСТИЧНОГО ПРИЗНАЧЕННЯ

### 2.1 Концептуальна модель процесу підтримки якості програмного забезпечення діагностичного призначення

Концептуальне моделювання є одним із центральних видів діяльності в галузі інформаційних технологій. Концептуальні моделі в основному використовуються як проміжні артефакти для побудови системи. Вони є схематичними описами системи, теорії або явища походження, таким чином утворюючи модель.

Концептуальна модель - це модель, яка містить доповнення певними концепціями. Процес концептуального моделювання регулюється метою моделювання та моделями. Він базується на ряді актів моделювання, на ряді умов коректності, на принципах і постулатах моделювання, а також на парадигмах фонових або субстанційних теорій. Цілі визначають (додаткову) вартість моделі. Концептуальне моделювання виконує аналітик-програміст, який керує процесом на основі свого досвіду, освіти, розуміння, наміру та ставлення. Концептуальні моделі - це продукти, які використовуються іншими зацікавленими сторонами, такими як програмісти, учні, бізнес-користувачі та оцінювачі. Концептуальні моделі використовують мову як носій артефакту моделювання та обмежені виразністю цього носія.

Моделювання в інженерії програмного забезпечення включає процес збору та аналізу системи, а також вимоги до побудови моделі.

Модель включає конструювання статичних, динамічних та поведінкових представлень. В описі поведінки системи, дій та стани зазвичай використовуються поняття.

Загалом, моделювання в різних його формах є одним із найфундаментальніших процесів людського розуму оскільки це дозволяє бачити шаблони, оцінювати та маніпулювати ними в процесах і речах, а також для вираження значення.

Концептуальний моделі - це моделі, що складаються з концепцій, які формують шлях до реальної розробки програмного продукту. Концептуальна модель є артефактом що допомагає зрозуміти область і, отже, зробити свій внесок у виявлення відповідних функціональних вимог.

На рисунку 2.1 продемонстровано концептуальну модель процесу підтримки якості програмного забезпечення діагностичного спрямування, що в свою чергу передбачає такі блоки як:

- визначення проблеми (детальний опис із визначенням вимог до майбутньої програмної системи);
- збір та обробка даних (здійснюється збір інформації з різних доступних джерел, її опрацювання, класифікація тощо);

Вивчаючи кожен із різних наборів даних МРТ головного мозку, можна побачити, що між ними є якісні відмінності, як видно на їхніх зрізах 2D МРТ. Хоча порівнюємо різні датасети для сегментації знімків черепа МРТ, позначеного «Маскою мозку» видно, що маски сегментації знімків черепа подібні, тоді як попередньо оброблені зрізи 2D МРТ мозку пацієнта виглядають інакше. Подібним чином, коли поповнюються для сегментації ураження інсульту МРТ, позначеного «Маскою ураження», видно, що маски сегментації ураження схожі, тоді як попередньо оброблені зрізи 2D МРТ пацієнта відрізняються.

- вибір потрібного алгоритму машинного навчання;
- навчання моделі згідно обраного алгоритму;

З цих відмінностей, випливає, що потрібно тренувати 3D-варіантні моделі лише на навчальній розділеній частині і також оцінювати їх на розділеній частині перевірки цього набору даних. Якби спочатку навчили варіантну модель 3D на наборі даних для сегментації черепа, а потім оцінили його на іншому наборі даних то, ймовірно, були б отримані погані результати, оскільки зрізи 2D МРТ виглядають інакше, ймовірно, через використання інших наборів даних і різні системи МРТ сканера. Таким чином, не потрібно розгортати модель сегментації видалення черепа з набору даних для видалення черепів із випадків МРТ у наборі даних. Подібним чином, якби відбувалось навчання варіантної моделі 3D для

сегментації уражень після інсульту та оцінка її на наборі даних, то швидше за все, також було б отримано недостатні результати, оскільки зрізи 2D МРТ відрізняються. Однак, якщо досліджуються 2D-зрізи МРТ, вони виглядають схожими, ймовірно, тому, що їхні МРТ-сканери схожі, тому можна розгорнути навчену модель для виконання сегментації черепа, потенційно маючи високі шанси успішно прибравши цей шум, залишивши лише тканину мозку в наборі даних.

Для набору даних потрібно залишатися в межах цього набору даних, щоб виконувати завдання сегментації видалення черепа та сегментації уражень інсульту. У наборі даних про інсульт, якщо виконується сегментація черепа, щоб усунути шум, залишаючи лише тканину мозку, то потрібно переконатися, що тренується лише варіантну модель 3D на підмножині набору даних, наприклад, від 30% до 40%., тож можна гарантувати, що дана модель не буде переобладнана. Таким чином, можна збільшити ймовірність того, що 3D-варіантна модель добре узагальнює значну частину 60% даних, які вона не бачила. Отже, коли відбувається навчання іншої 3D-варіантної моделі на тій підмножині з 60% МРТ, які мають лише тканину мозку, можна належним чином навчити її для сегментації уражень інсульту, знаючи, що ці попередньо оброблені дані, з якими ведеться робота, були попередньо оброблені.

- проведення валідації та тестування;
- блок впровадження та підтримки (моніторингу);
- оновлення у разі необхідності.

Концептуальна картина описує домен реального світу, виключаючи технічні аспекти і служить керівництвом для наступного етапу проектування.

У розробці програмного забезпечення концептуальне моделювання має вирішальне значення у формуванні вимог до розробки програмного продукту.

На рисунку 2.1 представлено концептуальну модель процесу підтримки якості програмного забезпечення діагностичного призначення.

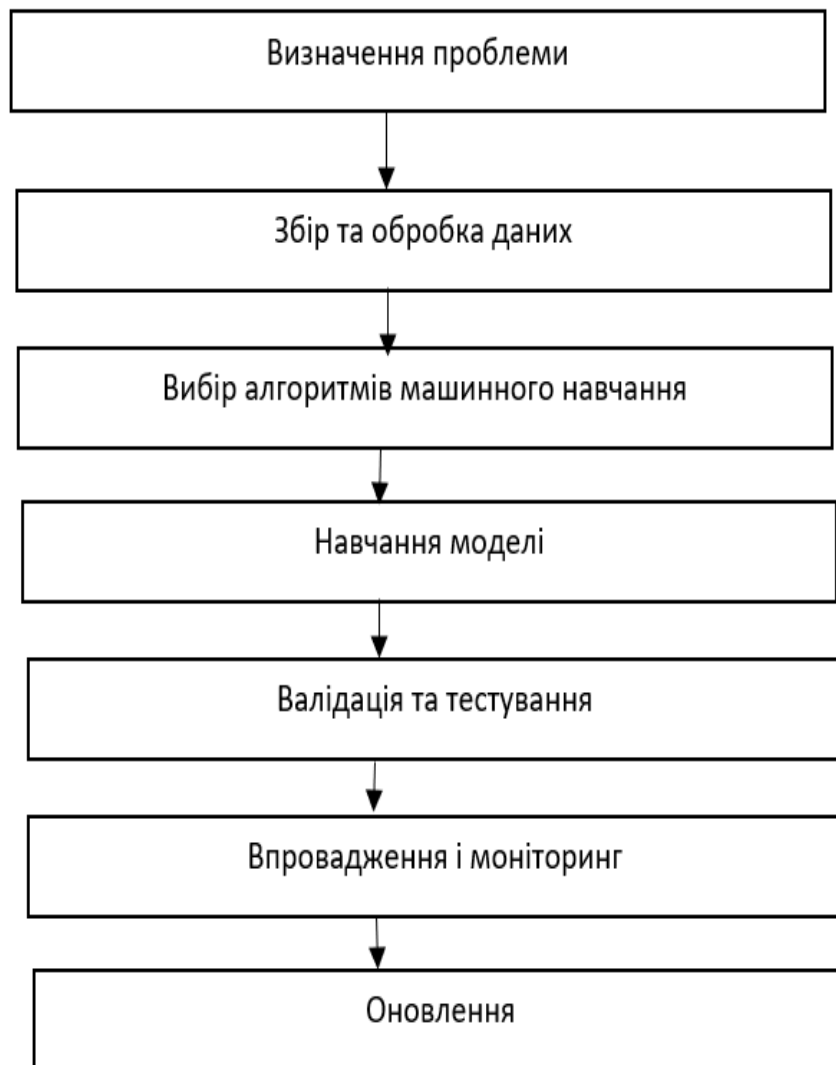


Рисунок 2.1 – Концептуальна модель процесу підтримки якості програмного забезпечення діагностичного призначення

## 2.2 Метод підтримки якості програмного забезпечення діагностичного призначення

Метод підтримки якості програмного забезпечення діагностичного спрямування полягає у тому, що спочатку формується набір даних метаданих. Для

попередньої обробки даних можна виконати різні операції над усім набором даних, а потім навчити нашу модель на загальній сегментації уражень інсульту, ігноруючи невидимі випадки. Один із підходів, який можна використати, це відфільтрувати лише випадки ішемічного інсульту та навчити модель на цих попередньо оброблених випадках для сегментації зачистки черепа та сегментації уражень інсульту, і вона, ймовірно, добре навчиться, оскільки в основному має справу з меншими ураженнями. Подібним чином, інший підхід, який можна використати, полягає в тому, щоб відфільтрувати випадки геморагічного інсульту та навчити нашу модель сегментувати геморагічні ураження, і модель, ймовірно, навчатиметься краще порівняно з ішемічним інсультом, оскільки вона вчиться відстежувати більші геморагічні ураження. Інший підхід, можна використати, полягає в тому, щоб продовжувати тренувати модель на загальній сегментації ураження інсульту, але спочатку збільшити кількість випадків геморагічного інсульту, щоб вони зрівнялися з випадками ішемічного інсульту, і тоді модель, ймовірно, навчиться краще відстежувати ураження різного розміру, оскільки є рівномірний розподіл. Хоча в ідеалі було б, ймовірно, краще збільшити кількість випадків геморагічного інсульту, запропонувавши рентгенологу відібрати більше пацієнтів з інсультом і збільшити розподіл випадків геморагії більш природним шляхом, ніж штучно за допомогою МРТ- доповнення (рисунок 2.2).

Даний метод удосконалений шляхом введення датасетів, що містять зображення МРТ сегментації ураження мозку. На основі цього було здійснено розробку алгоритму підтримки якості програмного забезпечення на основі машинного навчання.

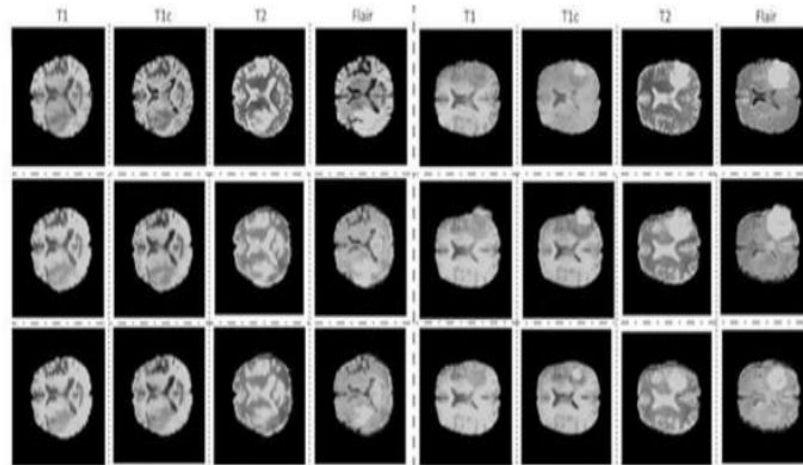


Рисунок 2.1 – Датасети МРТ

Для завдань сегментації МРТ пропонуються варіантні моделі 3D, які вчать витягувати характеристики з 3D-вокселів, а потім відстежувати ці об'єкти в 3D для сегментації зачисток черепа та сегментації уражень інсульту. Ці 3D МРТ-сканування мозку, представлені 3D-вокселями з наборів даних, попередньо обробляються в адаптованому для сегментації глибоким навчанням, і належним чином налаштовуються на основі даних. Ці попередньо оброблені дані МРТ зберігаються у потрібному форматі, а всі шляхи до файлів для підготовлених даних зберігаються у відповідному файлі.

У навчальному сценарії сегментації зчитується відповідний файл csv у фрейм даних, розділяється фрейм даних на 70% для навчання та 30% для тестування або перевірки. Потім передаються ці 2 кадри даних у 2 об'єкти набору даних, витягуються попередньо оброблені МРТ мозку та асоційовані мітки маски сегментації, щоб можна було отримати їх, коли завантажуються об'єкти набору даних за допомогою об'єктів для навчання та перевірки.

Далі встановлюються датасети на перетасування даних, використовуючи розмір партії, коли відбувається навчання варіантної моделі 3D, для виконання сегментації видалення черепа та сегментації уражень після інсульту після вивчення даних МРТ інсульту в наборі даних. Однак, датасет, який використовується для перевірки 3D-варіантної моделі для будь-якого завдання сегментації, не переміщує

дані та перевіряє модель в пакеті даних МРТ на кожному виконаному кроці, доки не досягнеться кінець епохи, зрештою повторюючи всі епохи.

Спочатку, перш ніж навчати варіантну модель, потрібно її ініціалізувати та налаштувати за допомогою каналів, оптимізаторів, функцій втрати та гіперпараметрів. Для сегментації черепа за допомогою МРТ створюється базова 3D-модель, передаючи 1 для вхідних каналів і 1 для вихідних каналів, оскільки навчання відбувається на бінарній масці сегментації. Для сегментації уражень інсульту за допомогою МРТ створюється модель на основі збудження від стискання з шарами та загального середнього об'єднання, оскільки це має допомогти шарам 3D згортки виділити особливості уражень інсульту різного розміру як для ішемічних, так і для геморагічних уражень інсульту.

Крім того, для сегментації уражень після інсульту МРТ модель 3D використовує 1 вхідний канал і 1 вихідний канал для навчання на бінарній масці сегментації. Для обох завдань сегментації МРТ 3D-моделі компілюються та навчаються за допомогою оптимізатора.

Під час навчання та перевірки 3D-моделі також оцінюються за допомогою оцінки кубиків і оцінки якості, оскільки вони є чудовими індикаторами того, наскільки добре модель передбачила сегментацію для пошкоджень черепа чи інсульту щодо перетину маски сегментації правдивості.

Для сегментації зачистки черепа МРТ обчислюється лише середня втрата, оцінка кубиків і оцінку якості, оскільки це дасть нам показник того, наскільки добре наша модель виконує сегментацію в середньому за епоху.

Однак, для сегментації уражень після інсульту МРТ, оскільки це більш складне завдання, то буде обчислено середню метрику для втрати, оцінки кубиків і оцінки якості, ми також обчислюємо найкращу метрику для кожного з них у кожній епосі, щоб ми могли порівнювати обидва випадки.

Потрібно знати в середньому, наскільки добре 3D відстежує ураження інсульту різного розміру за епоху, а також чи не працює він добре за середньою метрикою, якщо ми прагнемо отримати найкращі показники кожного разу за епоху, виконайте будь-які кроки, зроблені в межах епохи за всі епохи, ми бачимо, що

модель досягає найкращих показників для сегментації, але не так часто. Якщо ми можемо бачити, що 3D досягає хороших показників кубиків, просто не так часто, щоразу, коли заново налаштовуються гіперпараметри або повторно налаштовується спосіб попередньої обробки даних або оновлюється архітектура моделі 3D, то видно, що більше найкращих показників повторюються частіше протягом кожної епохи, що зрештою збільшує середній показник.

Для сегментації черепа за допомогою МРТ використовується базова 3D-модель, яка спочатку базувалася на 2D-архітектурі, але після оновлення згортки, максимального об'єднання тощо, отримано 3D-шари. Коли здійснюється перехід до сегментації уражень інсульту МРТ, то вдосконалюється 3D-архітектуру за допомогою блоків збудження стиснення, які допомагають покращити діапазон уваги моделі, щоб навчитися виділяти значущі характеристики щодо уражень інсульту різного розміру.

Базова 3D-модель використовує 3D-блоки подвійної згортки, які складаються з шару 3D-згортки, шару пакетної нормалізації, шару активації та тих самих 3 шарів, які повторюються для створення 3D-блоку подвійної згортки. Увага 3D використовує 3D-блоки подвійної згортки з додаванням блоку збудження стиснення, який складається з адаптивного 3D-середнього шару об'єднання, 3D-згорткового шару, шару активації, ще одного 3D-згорткового шару, за яким слідує сигмоподібний шар активації. Отже, вище описано кожен тривимірний блок подвійної згортки збудження стискання, який становить першу половину мережі 3D, яка має 4 блоки зменшення дискретизації, створюючи 3D-кодер, який модель використовує для вилучення 3D-об'єктів. Посередині 3D має 3D-блок подвійної згортки вузького місця, стискання, збудження, для запам'ятовування вилучених 3D-об'єктів, коли дані переходять до наступного етапу нашої моделі.

Для другої половини 3D також використовується 3D-блок подвійної згортки зі збудженням стискання, але на передній частині кожного блоку є 3D-транспонований шар згортки, який допомагає використати дискретизацію або декодувати витягнуті функції та має 4 блоки підвищення дискретизації (кожен є 3D-транспонований згортковий шар, а потім подвійні 3D-згорткові блоки

збудження), де з кожним підвищенням дискретизації 3D вчиться сегментувати 3D-витягнені особливості, що представляють інтерес, зрештою відстежуючи ураження інсульту. Крім того, частина 3D-декодера 3D під час трасування маски сегментації повертається до попередніх знань моделі про витягнуті функції з 3D-кодувальника за допомогою пропуску з'єднань, що є відображенням між 3D-кодувальником і 3D-декодером, щоб декодер міг отримати доступ різні масштаби вилучення ознак кодера, оскільки декодер сегментує більш чіткий об'єкт інтересу.

### 2.3 Алгоритм процесу підтримки якості програмного забезпечення діагностичного спрямування

Для впровадження розробленого методу підтримки якості програмного забезпечення діагностичного спрямування було запропоновано алгоритм підтримки якості програмного забезпечення діагностичного спрямування, що передбачає проходження ряду етапів від визначення проблеми до підтримки розробленої системи.

Першим кроком є визначення проблеми, опис її межі та предметної області. Також потрібно визначити цільову область діагностики (медична, технічна тощо). Далі визначаються типи даних, які будуть використовуватися для діагностики.

Наступним кроком має бути збір даних. Потрібно зібрати дані з визначених джерел. Це можуть бути медичні записи, технічні звіти тощо. Необхідно також переконатися, що дані містять достатню кількість інформації для навчання моделі.

Для коректної роботи усієї системи необхідно, щоб дані були актуальними та достовірними. Для цього потрібно здійснити їх очищення, тобто видалити або виправити неповні, помилкові, або некоректні записи.

Після того, як відбувається очищення та обробка необхідних наборів даних здійснюється нормалізація даних. Щоб здійснити нормалізацію даних потрібно привести дані до єдиного формату (масштабування числових значень, кодування категорійних змінних).

Дані, що вже сформовані розподіляють на вибірки: навчальну, валідаційну та тестову вибірки.

Також важливим кроком є здійснення вибору, а також налаштування моделі машинного навчання. Для цього обирається відповідний алгоритм, що найбільш підходить до даної ситуації. це може бути регресивний метод, класифікації тощо.

Модель налаштовується шляхом оптимізації гіперпараметрів алгоритму.

Важливим кроком також є навчання моделі на навчальних даних із використанням валідаційної вибірки для налаштування моделі. Навчальна модель - це набір даних, який використовується для навчання алгоритму машинного навчання. Він складається з вибірки вихідних даних і відповідних наборів вхідних даних, які впливають на вихід. Навчальна модель використовується для проходження вхідних даних через алгоритм, щоб співвіднести оброблений вихід із зразком. Результат цієї кореляції використовується для модифікації моделі.

Цей ітеративний процес називається «припасуванням моделі». Точність навчального набору даних або набору даних перевірки має вирішальне значення для точності моделі.

Навчання моделі машинною мовою – це процес подачі даних в алгоритм ML, щоб допомогти визначити та вивчити хороші значення для всіх задіяних атрибутів. Існує кілька типів моделей машинного навчання, з яких найпоширенішими є контрольоване та неконтрольоване навчання.

Контрольоване навчання можливе, коли навчальні дані містять як вхідні, так і вихідні значення. Кожен набір даних, який має вхідні та очікувані вихідні дані, називається контрольним сигналом. Навчання виконується на основі відхилення обробленого результату від задокументованого результату, коли вхідні дані вводяться в модель.

Навчання без нагляду передбачає визначення закономірностей у даних. Потім додаткові дані використовуються для підгонки шаблонів або кластерів. Це також ітераційний процес, який покращує точність на основі кореляції з очікуваними шаблонами або кластерами. У цьому методі немає довідкового набору вихідних даних.

Після проведення налаштування та навчання моделі здійснюється тестування моделі на тестових даних (обраних датасетах) та оцінюються результати за допомогою метрик загальна точність, f1-score, recall, precision.

Розробка графічного інтерфейсу для користувачів із наданням дозволу вводити дані для отримання діагнозу.

У процесі діагностики користувач вводить необхідні дані в систему, яка відповідно здійснює аналіз даних за допомогою навченої моделі. Вихідні результати виводяться у вигляді діагностики або рекомендацій.

До алгоритму також входять моніторинг та оновлення програмної системи, що є важливими для забезпечення безпеки, ефективності та сумісності ІТ-систем вашої організації. Однак вони також можуть створювати проблеми, такі як простоя, проблеми сумісності, опір користувачів і бюджетні обмеження, тому необхідно регулярно моніторити ефективність системи та оновлювати модель новими даними для поліпшення результатів.

Важливим також є зворотній зв'язок із користувачами для вдосконалення та оновлення програмної системи. Для цього проводяться постійні опитування, анкетування тощо.

Цей алгоритм може бути адаптований та деталізований залежно від специфіки застосування системи та її цілей.

На рисунку 2.2 зображено алгоритм підтримки якості програмного забезпечення діагностичного призначення.

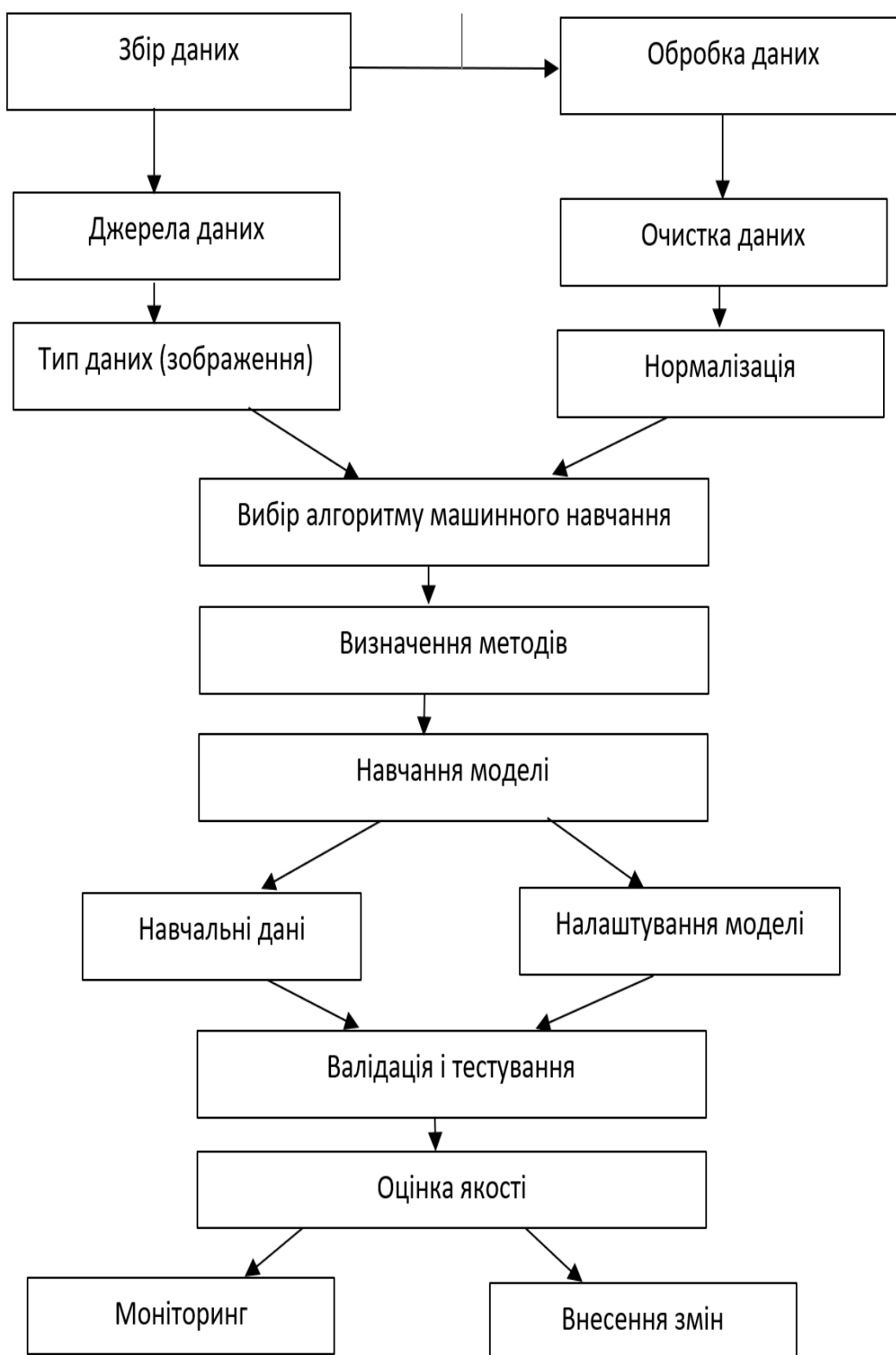


Рисунок 2.2 – Алгоритм підтримки якості програмного забезпечення діагностичного призначення

Отже, як бачимо вказаний алгоритм містить велику кількість кроків для досягнення кінцевої мети, а саме оцінки якості програмної системи діагностичного спрямування із використанням штучного інтелекту.

#### 2.4. Висновки до другого розділу

Отже, у другому розділі було здійснено концептуальне моделювання, результатом якого стала концептуальна модель, що містить в своїй основі складові для впровадження у програмне забезпечення призначене для діагностики, зокрема хвороб нервової системи.

Також було описано удосконалений метод, заснований на використанні машинного навчання, що полягає у введенні спеціалізованих датасетах, що містять зображення МРТ сегментації ураження мозку.

Для впровадження розробленого методу підтримки якості програмного забезпечення діагностичного спрямування було запропоновано алгоритм підтримки якості програмного забезпечення діагностичного спрямування, що передбачає проходження ряду етапів від визначення проблеми до підтримки розробленої системи.

### **3 АРХІТЕКТУРА СИСТЕМИ ПІДТРИМКИ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДІАГНОСТИЧНОГО ПРИЗНАЧЕННЯ**

3.1 Формування та аналіз вимог програмної реалізації системи підтримки якості програмного забезпечення діагностичного призначення

Якість зазвичай визначається як ступінь, до якого набір властивих характеристик відповідає вимогам. Вимоги - це потреби або очікування що зазначено, мається на увазі або є обов'язковим. Один із найслабших місць сучасного етапу програмного забезпечення стандартів якості продукції є відсутність вказівки щодо точного формулювання вимог до якості. Для неупередженої оцінки якості такі вимоги мають бути чіткими, повними, послідовними і вимірюваними.

Ключовим та важливим фактором в процесі проектування та розробки будь-якого програмного забезпечення будь-якої складності є формування вимог, зокрема функціональних та нефункціональних.

В рамках даної кваліфікаційної роботи визначено систему як організовану комбінацію взаємодіючих елементів для досягнення однієї чи кількох заявлених цілей, а також межі системи.

Межа системи залежить від точки зору. Що становить елементи системи, залежить від точки зору, і може бути кілька різних відповідних способів визначення елементів системи. Програмне забезпечення можна вважати одним із елементів системи.

Функція вимірювання представляє інтерпретацію властивості якості програмного забезпечення та цільового значення, а міра якості представляє вимогу до якості.

Аналогічно, фактичне значення вимірювання якості представляє спостережувану якість програмного забезпечення.

Вимоги до якості програмного забезпечення, а також усі інші вимоги не можна розглядати ізольовано, але повинні розглядатися в ширшому контексті.

Вимоги до якості програмного забезпечення мають особливий тісний зв'язок із функціональними вимогами. Функціональні вимоги відіграють важливу роль для визначення вимог до якості програмного забезпечення. Функціональність є однією з шести характеристик для внутрішньої та зовнішньої якості у відповідності до стандартів. Вимоги до функціональності не слід плутати з функціональними вимогами.

Функціональність - це здатність програмного забезпечення надавати функції, які відповідають його функціональним вимогам. Вимоги до функціональності уточнюються у відповідності до вимог програмного продукту, точні, сумісні, безпечні та сумісні за відповідними функціональними стандартами та правилами.

У деяких ситуаціях має сенс вказати яка вимога до якості програмного продукту, тоді як в іншій ситуації, до яких застосовуються лише вимоги до якості частини програмного продукту. Наприклад, деякі функції актуальні лише для конкретних користувачів і мають специфічні вимоги до якості, які відрізняються від вимоги до якості інших функцій, призначених для інших цілей та інших користувачів. Тому важливо визначити, якою частиною є програмний продукт відповідно до вимогам якості програмного забезпечення. Іншими словами, вимога до якості пов'язана з частиною програмного продукту (набору функцій).

Наприклад, деякі функції можуть бути призначені для звичайних кінцевих користувачів і, отже, можуть вимагати низьку похибку толерантності, тоді як інша група функцій може бути призначеними для спеціалістів і таким чином дозволити більшу толерантність до помилок. В обох випадках допускається механічна помилка і необхідний ступінь допуску до помилок повинен бути строго визначеним.

Якість у використанні визначається як ступінь, в якому продукт відповідає потребам певних користувачів, тобто визначені цілі з ефективністю, продуктивністю, безпекою і задоволення в певному контексті використання. Тому вимоги до якості у використанні тісно пов'язані з іншими системними вимогами, наприклад, такими як вимоги до обладнання, бізнес-вимоги та вимоги кінцевого користувача.

Виходячи із вище визначеного, програмна система повинна задовольняти вимогам, що висуваються до програмного забезпечення діагностичного призначення, а саме діагностики хвороб нервової системи. А саме визначати якісний ступінь проведеного діагностичного дослідження.

Передбачається, що це мають бути кілька модулів, тобто модуль збору даних, модуль висновків, базу даних для зберігання зібраних вимог і результатів процесу висновків, а також модуль презентації, який використовується для демонстрації можливого проекту діагностичної системи. Для розробки цієї системи використовувалася мова програмування (наприклад, Java чи Python) та середовище баз даних, наприклад MySQL чи інші. Процес виведення здійснюється на мові програмування R з використанням вибраних функцій із пакету. Кінцевим результатом використання цього програмного забезпечення є звіт у вигляді списку, який містить очікувані функціональні можливості діагностичної системи та відповідні їм підсистеми, що описують оптимальний проект діагностичної системи.

### 3.2 Розробка архітектури системи якості програмного забезпечення діагностичного спрямування

Для того, щоб система функціонувала необхідно здійснити проектування її архітектури. При розробці програмного забезпечення використовується різноманітні типи архітектурних рішень. В даному випадку доцільно використовувати багат шарову модель.

В архітектурі програмного забезпечення рівні діють як окремі процеси в інфраструктурі програми. Ці шари зазвичай утворюють шаблон, який також називають шаблоном n-рівневої архітектури. Це галузевий стандарт для багатьох програм, і це широко відома концепція багатьом дизайнерам, інженерам, архітекторам і розробникам. У шаблоні багаторівневої архітектури кожен рівень виконує певну відповідальність і роль у програмі. Деякі зосереджуються на логіці

інтерфейсу користувача, а інші займаються виконанням бізнес-правил. Ці рівні доповнюють унікальні цілі один одного, але вони явно не залежать один від одного для виконання власних завдань.

По суті, шари складають структуру шаблону багаторівневої архітектури. Хоча цей шаблон не вказує чітку кількість або тип шарів, більшість багат шарових архітектур складаються з чотирьох-п'яти стандартних шарів. Однак розмір програми може визначити, скільки шарів вона має. Наприклад, менші програми можуть мати лише три рівні. Порівняно, більші та складніші програми можуть містити більше п'яти або більше рівнів. Існує розділення завдань між різними рівнями, тобто кожен рівень має справу лише з логікою, яка стосується його власного рівня.

Передбачається, що це буде програмний продукт, який має такі складові як датасети (спеціалізований набір даних).

Датасети, а саме здійснення збору та підготовки даних, що включають медичні записи, рентгенографічні зображення, МРТ, ЕЕГ та інші діагностичні дані.

Далі, відповідно до розробленого алгоритму необхідна безпосередня підготовка даних, до якої входить очищення даних, нормалізація та анотація даних для покращення якості моделі. Після цього відбувається вибір необхідної моделі, що включає використання нейронних мереж, таких як CNN (Convolutional Neural Networks) для аналізу зображень, або RNN (Recurrent Neural Networks) для аналізу темпоральних даних.

Для того аби досягти поставленого завдання потрібно здійснити навчання моделі. Тобто необхідна підготовка моделі на основі вибраних алгоритмів машинного навчання, використовуючи методи, такі як перетворення зображень, аугментація даних та оптимізація градієнтів.

Після того як відбулось навчання моделі потрібно здійснити її перевірку. Для цього використовуються відокремлені дані для перевірки точності моделі та виявляються можливі перевантаження.

Фактично заключним етапом є впровадження програмного продукту, тобто інтеграція моделі в медичні системи для діагностики хвороб нервової системи, забезпечення інтерфейсу для лікарів та пацієнтів.

Загальну структуру можливої програмної системи підтримки якості програмного забезпечення діагностичного спрямування показано на рисунку 3.1.

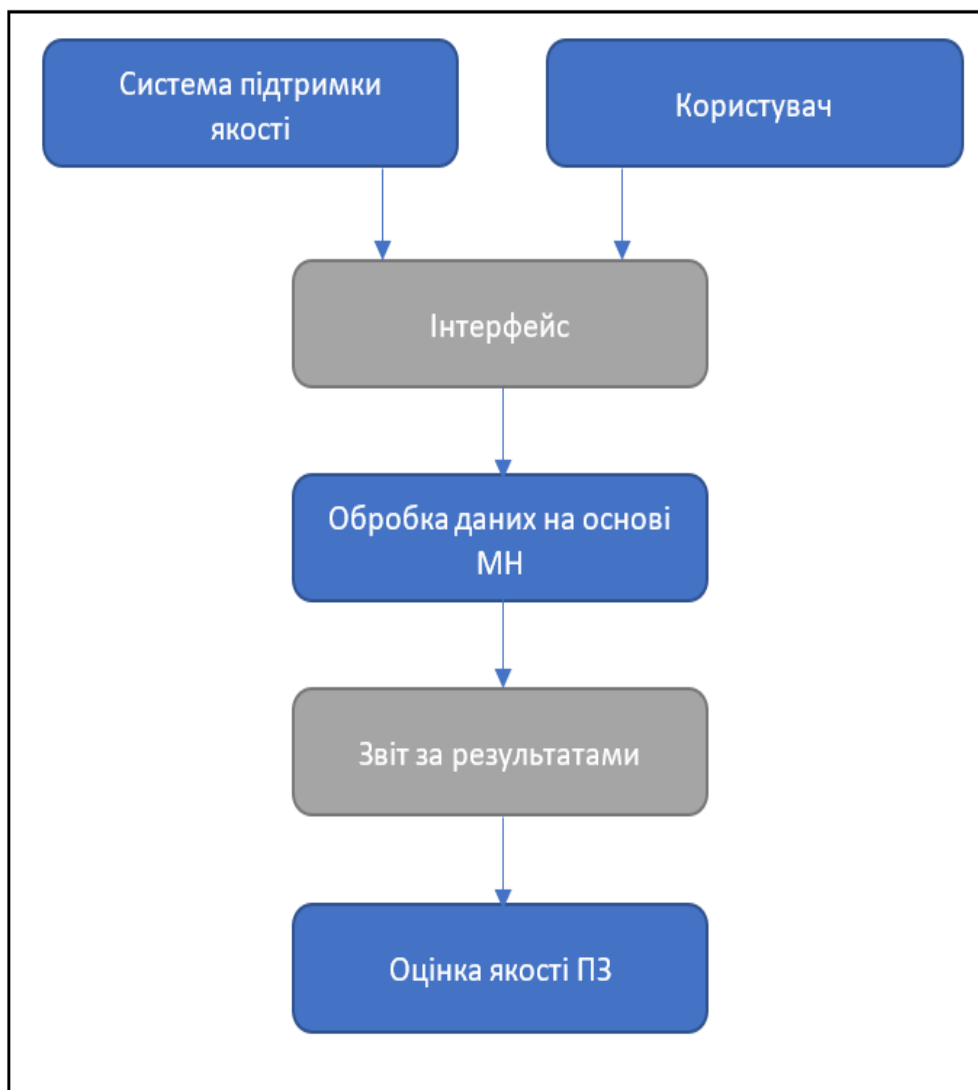


Рисунок 3.1 – Архітектура системи підтримки якості програмного забезпечення діагностичного спрямування

Отже, архітектурне рішення складається із таких блоків (рівнів): інтерфейс користувача (презентаційний рівень), прикладного рівня, бізнес-рівня, рівень доступу до даних, рівень бази даних.

Рівень презентації, який також називають рівнем інтерфейсу користувача, обробляє взаємодію користувачів із програмним забезпеченням. Це найпомітніший та найдоступніший рівень, який визначає загальний вигляд і представлення програми для кінцевих користувачів. Це рівень, який будь-хто може використовувати зі свого клієнтського пристрою.

Прикладний рівень обробляє основні програми архітектури. Він містить визначення коду та більшість основних функцій розробленої програми. Це рівень, на якому програмісти проводять більшу частину свого часу, працюючи над програмним забезпеченням. Ви можете використовувати цей рівень, щоб реалізувати певну логіку координації, яка не узгоджується повністю ні з презентаційним, ні з бізнес-рівнем.

Бізнес-рівень (доменний рівень) - робота бізнес-логіки програмної системи. Бізнес-логіка - це набір правил, які вказують системі, як запускати програму на основі вказівок організації. Цей рівень, по суті, визначає поведінку всієї програми. Після завершення однієї дії програма повідомляє, що робити далі.

Рівень доступу до даних, діє як захисний шар. Він містить код, необхідний для доступу до рівня бази даних. Цей рівень також містить набір кодів, які дозволяють маніпулювати різними аспектами бази даних, такими як деталі підключення та оператори SQL.

Рівень бази даних – це місце, де система зберігає всі дані. Це найнижчий рівень в архітектурі програмного забезпечення, який містить не лише дані, а й індекси та таблиці. Тут часто відбуваються операції пошуку, вставки, оновлення та видалення. Дані програми можуть зберігатися на файловому сервері або сервері бази даних, відкриваючи важливі дані, але зберігаючи процедури зберігання та пошуку даних прихованими.

### 3.3. Висновки до 3-го розділу

У третьому розділі здійснено аналіз вимог до програмного забезпечення діагностичного спрямування. Встановлено, що ключовим та важливим фактором в процесі проектування та розробки будь-якого програмного забезпечення будь-якої складності є формування вимог, зокрема функціональних та нефункціональних.

В рамках даної кваліфікаційної роботи визначено систему як організовану комбінацію взаємодіючих елементів для досягнення однієї чи кількох заявлених цілей, а також межі системи.

## 4 ОЦІНКА СИСТЕМИ ПІДТРИМКИ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДІАГНОСТИЧНОГО ПРИЗНАЧЕННЯ

4.1 Методологія оцінювання програмних систем якості програмного забезпечення діагностичного призначення

Оцінка якості програмного забезпечення є одним із найважливіших процесів для надання якісних програмних продуктів. Оцінка якості програмного забезпечення має значний вплив на весь проект. Однак, для оцінки якості програмного забезпечення доступна низка методів. Більшість моделей базуються на характеристиках програмного забезпечення та відповідних оцінках. Але розгляд і оцінка програмного забезпечення за всіма якісними характеристиками є трудомісткими і складними, що підходить не для всіх видів проектів.

Так як причини несправностей програмних систем є різноманітні, попит на розвиток і еволюцію нових рішень, алгоритми, моделі та методи для виробництва надійної програмної системи збільшення. Щоб оцінити надійність програмного забезпечення систем, недостатньо розробити методи, потрібно також впроваджувати використання результатів, отриманих під час тестування. Важливими є фактори, що впливають на надійність програмних систем і впорядковуються вони відповідно до рейтингу на основі емпіричних експериментів, а саме: документації, тестування, часу, складності програмного забезпечення, навичок програміста тощо.

Мінімізація помилки в програмному коді є однією з найбільш актуальних питань забезпечення надійності. Для реалізації потрібен комплексний підхід, тобто мають розглядатись такі питання:

- запобігання несправностям;
- виявлення та усунення несправностей;
- забезпечення відмовостійкості;
- розробка передбачення несправностей і моделі підвищення надійності;

До першої групи належать принципи та методи спрямовані на мінімізацію або повне усунення несправності. До цієї групи належать принципи та методи метою якого є запобігання помилок у готовому програмний код. Їх можна поділити на такі категорії:

Залежно від мети проекту, команда, яка працює над програмним забезпеченням, має бути вибрано правильно, тому що основним джерелом в недоліки - людський фактор.

Мінімізація складності програмного коду, оскільки інтенсифікація в ускладненні призводить до збільшення кількості невиявлених помилок.

Запобігання помилкам є найбільш оптимальним рішенням для розробки надійного програмного забезпечення. Однак, незважаючи на ці заходи, неможливо розробити безвідмовне програмне забезпечення. Джерелом збоїв у програмному забезпеченні є професіонали, які над цим працюють. Іншими словами, це конкретні люди з індивідуальними особливостями, професіоналізм, талант і досвід.

До другої групи відносяться розробки нових підходів та методів виявлення помилок під час тестування програмного забезпечення. Виявляються помилки і усуваються під час тестування програмних систем.

Збільшення складності програмних систем збільшив попит на розвиток сучасних методи тестування. Тестування програмного забезпечення – це ефективний технічний засіб, що забезпечує якість програмні продукти шляхом виявлення помилок.

Розуміння причини та типу помилок робить легшим керування ними, зменшує витрати на тестування та допомагає зменшити повторення дефектів шляхом перенавчання розробники програмного забезпечення. Таким чином, класифікація помилки програмного забезпечення є центральним процесом. Крім того, збільшення кількості програмних систем і збільшення складності системи, кількість дефектів зросла до такої міри, що є їх неможливо точно і швидко класифікувати традиційними методами. Розвиток машинного навчання сприяло автоматизації процесу тестування.

У процесі тестування задіяно інтеграцію штучного інтелекту у процеси тестування програмного забезпечення із використанням нових методів та моделей для ефективної обробки великих даних.

Застосування штучного інтелекту технології в тестуванні програмних систем забезпечує більш точні результати, ніж традиційні методи тестування та зменшує час, витрачений на тестування, скорочення часу розробки ПЗ.

До третьої групи відносяться розробки нових методів автоматичного відновлення ПЗ коли виникають певні помилки. Стійкість – це здатність програмного забезпечення системи продовжувати працювати за будь-яких умов та обставини.

Методи стабільності полягають у запобіганні неактивним помилкам, перетворюючись на активні помилки. Резервні версії програмних систем створюються для забезпечення їх стійкості. Мета методів з цієї групи є підтримка роботи програмного забезпечення системи, незважаючи на наявність помилок. Коли помилка виникає в основній програмі, контроль переноситься в програму резервного копіювання.

Недоліком цих методів є вартість програмної системи, яка звісно збільшується, її важко виявити помилки в процесі тестування. Цей спосіб використовується в особливо критичні системи, зокрема й медичного та діагностичного призначення.

До четвертої групи відносяться розробки методів та алгоритмів прогнозування помилок в програмні модулі та оцінка надійності. Програмне забезпечення моделювання надійності є одним з найбільш активних напрямків розробки програмного забезпечення.

4.2 Оцінка системи якості програмного забезпечення діагностичного спрямування

У даному розділі необхідно оцінити методологію та проаналізувати показники продуктивності для моделей CV і NLP, які використовуються в конвеєрі глибокого навчання для сегментації видалення черепа, сегментації уражень інсульту та субтитрів МРТ у системі діагностики інсульту. Спочатку, щоб побачити, наскільки добре ці моделі сегментації вчать виконувати завдання сегментації, потрібно оцінити їхнє навчання, використовуючи фокусні втрати, і втрати категорійної крос-ентропії.

Як ранній крок до оцінки моделі сегментації черепа можна оцінити її за допомогою точності пікселів. Для оцінки продуктивності моделей сегментації зачистки черепа та моделей сегментації уражень інсульту можна оцінювати їх на основі оцінки втрати кубиків, оцінки якості, які обидва обчислюють, наскільки добре основна правда порівнюється з прогнозованою маскою сегментації для перетину над об'єднанням. Для використовуваної моделі субтитрів до медичних зображень оцінювання здійснюється на основі показника, який використовується, щоб допомогти побачити, наскільки згенеровані дані були близькі до правдивих даних.

Під час навчання в мережу надходять дані 3D-сканування МРТ мозку та 3D-анотованих масок мозку, потім мережа розділяє всі 3D-сканування МРТ для кожного пацієнта на зрізи 2D-МРТ, а потім усі ці зрізи 2D-МРТ розширюються в загальний список, у якому кожен зріз 2D МРТ має 2D анотовану маску сегментації мозку. Таким чином, мережа тренується на кожному 2D-зрізі МРТ із пов'язаною 2D-анотованою мозковою маскою без черепа. Після того, здійснено навчання мережі і виявлено її показники продуктивності, відбувається подальше оцінювання її на тестовому наборі даних 3D МРТ мозку. Коли досягаються показники продуктивності тестування, які задовольняють вимогам, відбувається тестування мережі 2D на зовнішньому наборі даних.

Мережа автоматизує видалення черепа з 3D МРТ-сканування мозку. Набір даних 2.0, тому попередня обробка максимально точна, подібно до роботи спеціаліста з відстеження мозку. Втрати тренувального фокусу зменшуються до менше 0,1 після 12 епох, тоді як втрати перевірки зменшуються трохи нижче 0,2.

Втрати в цілому говорять, наскільки добре двовимірна модель вчиться виконувати сегментацію черепа; низькі втрати в нашій моделі означають, що вона добре вчиться. Фокусна втрата є сумою SSE, а оцінка втрат кубиків, була обрана, оскільки вона настійно рекомендована для роздачі.

Для сегментації зачистки черепа на МРТ навчалась та перевірялась модель на МРТ інсульту із масками сегментації зачистки черепа. Було оцінено базову 3D-модель за допомогою середніх показників, що можна продемонструвати за допомогою результуючих графіків (рисунок 4.1). Для середньої втрати видно, що модель досягла втрат на навчання та перевірку нижче 0,025. На рисунку позначено LossT – тренування, Loss V – валідація (перевірка). Для середнього показника валідації на рисунку 4.2 видно, що модель досягла близько 0,95, а для оцінки валідації модель отримала вище 0,97.

Для сегментації черепа за допомогою МРТ також маємо таблицю показників навчання та перевірки результатів моделі на основі МРТ інсульту з черепом.

### LossT і Loss V

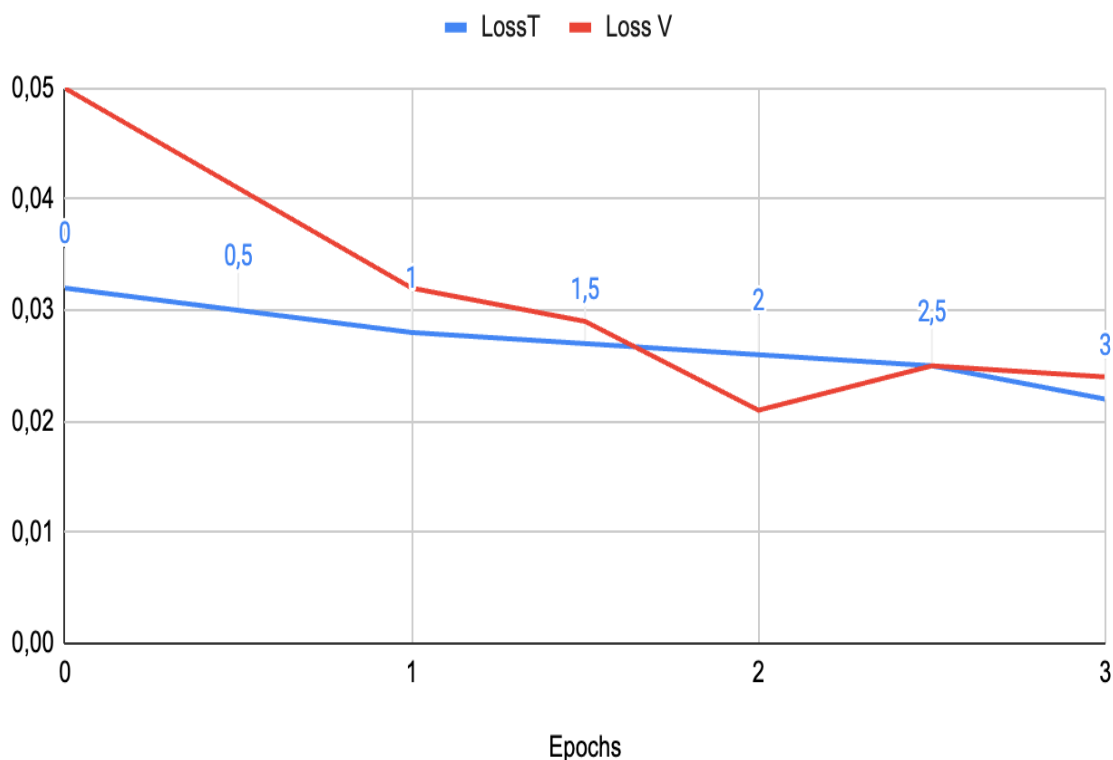


Рисунок 4.1 – Результати втрат

## Valid\_Av i Valid\_R

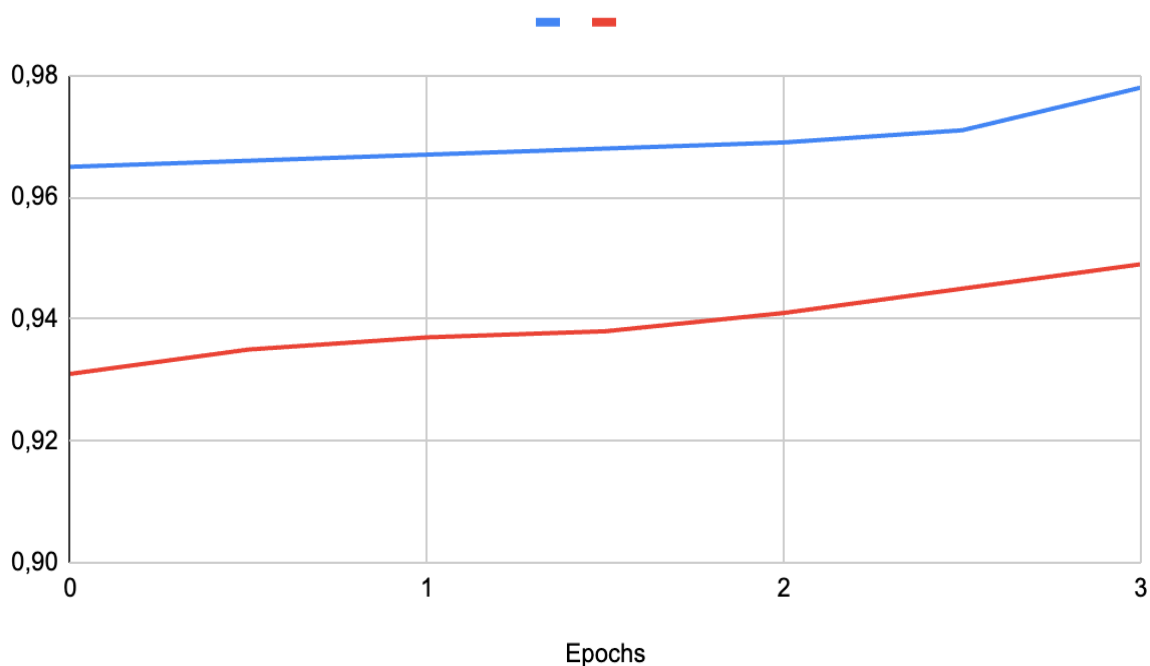


Рисунок 4.2 – Результати перевірки втрат кубиків

Отже, як видно із описаних вище даних модель отримала середню оцінку кубиків перевірки 0,95, оцінку валідації 0,97 і втрату 0,02 за епоху 3.

На рисунку 4.2 позначено valid\_av – середній показник валідації, valid\_r – оцінка валідації.

Можна зробити висновки про те, що перевірка на якість програмного забезпечення показує зростання до 97 % проти 95 %, що на 2% більше, що більш наочно та схематично показано на рисунку 4.3. Для використання програмного забезпечення діагностичного спрямування це досить велика цифра, оскільки навіть незначні покращення діагностики стану здоров'я, зокрема неврологічних захворювань можуть покращити якість діагностики в цілому, що суттєво впливатиме на життя та здоров'я пацієнта медичного діагностичного закладу.

## Оцінка якості , %

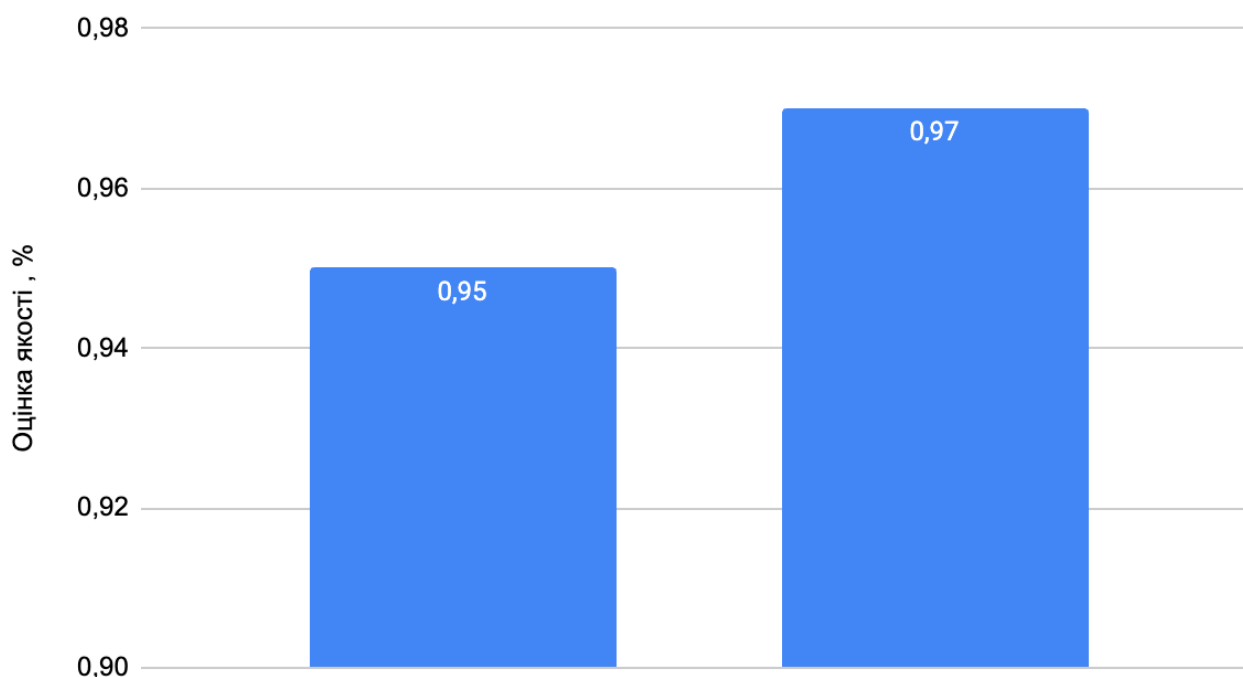


Рисунок 4.3 – Результат оцінки якості програмної системи

### 4.3 Висновки до 4-го розділу

У розділі було здійснено виокремлення теоретичних засад оцінки якості програмного забезпечення діагностичного спрямування. Визначено, що оцінка якості програмного забезпечення відіграє надзвичайно важливу роль у процесі розробки програмного забезпечення, оскільки напряду впливає на надання якісних програмних продуктів. У випадку із програмним забезпеченням діагностичного спрямування якість може відігравати ще й критичну роль, оскільки від якісної діагностики залежить не тільки здоров'я, а подекуди й людське життя, особливо коли це стосується хвороб нервової системи.

Оцінка якості програмного забезпечення має значний вплив на весь проект. Разом з тим, для оцінки якості програмного забезпечення доступна низка методів. Більшість моделей базуються на характеристиках програмного забезпечення та відповідних оцінках. Але розгляд і оцінка програмного забезпечення за всіма

якісними характеристиками є трудомісткими і складними, що підходить не для всіх видів проектів.

У випадку використання машинного навчання для оцінки точності навчання обраної моделі здійснюється вимірювання середніх показників, що продемонстровано за допомогою результуючих графіків. Загалом для середньої визначено, що модель досягла втрат на навчання та перевірку нижче 0,025. Для середнього показника валідації, що модель досягла близько 0,95, а для оцінки валідації модель отримала вище 0,97.

## ВИСНОВКИ

Отже, в результаті проведеного дослідження можна зробити такі висновки.

Якість є важливою складовою розробки програмного забезпечення. Медичні заклади використовують складні програмні засоби для керування великою кількістю медичних даних та їх обробки, щоб поставити ранній і точний діагноз і таким чином забезпечити прийняття важливих клінічних рішень. Через важливість цих програмних продуктів розробники програмного забезпечення для охорони здоров'я повинні здійснити перевірку та оцінку свого програмного забезпечення, щоб гарантувати його ремонтпридатність, надійність і характеристики безпеки, тобто його якість. Помилки у роботі програмного забезпечення, такі як неправильне програмування та помилкові розрахунки, небажані в будь-якій проблемній області, але в медичному середовищі вони можуть призвести до прийняття неправильних клінічних рішень і, як наслідок, спричинити смерть. Іншими словами, програмне забезпечення для охорони здоров'я відрізняється від інших типів програмного забезпечення, оскільки воно може створювати загрози та завдати шкоди людському життю.

У першому розділі було здійснено аналіз предметної області. Результатом цього аналітичного матеріалу стали сталі уявлення про те, що таке машинне навчання та як його можна використовувати під час розробки програмного забезпечення діагностичного призначення.

Також здійснено аналіз методів та засобів машинного навчання та їх використання відповідно до предметної області. В даному дослідженні предметною областю є медична галузь, а саме діагностування хвороб нервової системи.

Було визначено мету даного дослідження, що полягає в удосконаленні методики використання машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

Відповідно до мети були встановлені такі задачі:

1. Здійняти аналіз предметної області, а саме програмне забезпечення діагностичного призначення.

2. Провести аналіз тих методів, що вже існують в реальних умовах.

3. Здійснення дослідження методів машинного навчання для досягнення та підтримки якості програмного забезпечення діагностичного призначення.

4. Удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

В переважній більшості випадків використовуються такі алгоритми машинного навчання:

- нейронні мережі;
- лінійна регресія;
- логістична регресія;
- кластеризація;
- дерева рішень;
- випадкові ліси.

Кожен із цих методів має свої переваги та недоліки і може використовуватись відповідно до тих задач, що потребують виконання та розв'язку.

У другому розділі було здійснено концептуальне моделювання, результатом якого стала концептуальна модель, що містить в своїй основі складові для впровадження у програмне забезпечення призначене для діагностики, зокрема хвороб нервової системи.

Також було описано удосконалений метод, заснований на використанні машинного навчання, що полягає у введенні спеціалізованих датасетах, що містять зображення МРТ сегментації ураження мозку

Для впровадження розробленого методу підтримки якості програмного забезпечення діагностичного спрямування було запропоновано алгоритм підтримки якості програмного забезпечення діагностичного спрямування, що передбачає проходження ряду етапів від визначення проблеми до підтримки розробленої системи.

У третьому розділі здійснено аналіз вимог до програмного забезпечення діагностичного спрямування. Встановлено, що ключовим та важливим фактором в процесі проектування та розробки будь-якого програмного забезпечення будь-якої складності є формування вимог, зокрема функціональних та нефункціональних.

В рамках даної кваліфікаційної роботи визначено систему як організовану комбінацію взаємодіючих елементів для досягнення однієї чи кількох заявлених цілей, а також межі системи.

У четвертому розділі У розділі було здійснено виокремлення теоретичних засад оцінки якості програмного забезпечення діагностичного спрямування. Визначено, що оцінка якості програмного забезпечення відіграє надзвичайно важливу роль у процесі розробки програмного забезпечення, оскільки напряму впливає на надання якісних програмних продуктів. У випадку із програмним забезпеченням діагностичного спрямування якість може відігравати ще й критичну роль, оскільки від якісної діагностики залежить не тільки здоров'я, а подекуди й людське життя, особливо коли це стосується хвороб нервової системи.

Оцінка якості програмного забезпечення має значний вплив на весь проект. Разом з тим, для оцінки якості програмного забезпечення доступна низка методів. Більшість моделей базуються на характеристиках програмного забезпечення та відповідних оцінках. Але розгляд і оцінка програмного забезпечення за всіма якісними характеристиками є трудомісткими і складними, що підходить не для всіх видів проектів.

У випадку використання машинного навчання для оцінки точності навчання обраної моделі здійснюється вимірювання середніх показників, що продемонстровано за допомогою результируючих графіків. Загалом для середньої визначено, що модель досягла втрат на навчання та перевірку нижче 0,025. Для середнього показника валідації, що модель досягла близько 0,95, а для оцінки валідації модель отримала вище 0,97.

Практичне значення отриманих результатів. У цій кваліфікаційній роботі магістра показано удосконалення методу машинного навчання для підвищення якості програмного забезпечення діагностичного призначення. Результатом даного

дослідження став алгоритм роботи методу підтримки якості програмного забезпечення на основі машинного навчання. Це в свою чергу дозволить програмістам використовувати даний метод та алгоритм під час розробки програмного забезпечення в медичній галузі для продуктів діагностичного спрямування. Потрібність і актуальність подібних досліджень продемонстровано у перших двох розділах.

В ході проведення даного дослідження використано методи аналізу та синтезу, абстракції, порівняльні критерії.

Відповідно до теми кваліфікаційної роботи опубліковані тези «Метод підтримки якості програмного забезпечення на основі машинного навчання» на конференції «Актуальні проблеми комп'ютерних наук (АПКН-2024)».

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Barbareschi M., Barone S., Carbone R. et al. Scrum for safety: an agile methodology for safetycritical software systems. *Software Qual J.* 2022. №30, pp. 1067–1088 <https://doi.org/10.1007/s11219-022-09593-2>.
2. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. URL: [Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions | SN Computer Science \(springer.com\)](https://www.springer.com) (дата звернення 30.08.2024).
3. Improving Code Quality Through Predictive Analytics with KINE: A New Frontier. URL: <https://www.linkedin.com/pulse/improving-code-quality-through-predictive-analytics-kine-new-ttgsc> (дата звернення 31.08.2024).
4. IBM. URL: [https://www-ibm-com./](https://www.ibm-com/) (дата звернення 31.08.2024).
5. The Legal and Ethical Implications of Using AI in Hiring. URL: [https://hbr.tradepub.com/free/w\\_defa6904/prgm.cgi?a=1](https://hbr.tradepub.com/free/w_defa6904/prgm.cgi?a=1) (дата звернення 31.08.2024).
6. Deep Learning. URL: <https://www-ibm-com./> (дата звернення 01.09.2024).
7. ISO/IEC 9126-1:2001. Software engineering – Software product quality – Part 1: Quality model.
8. URL: [https://www-sciencedirect-com.translate.google.com/science/article/abs/pii/S2214785322014936?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=uk&\\_x\\_tr\\_hl=uk&\\_x\\_tr\\_pto=sc](https://www-sciencedirect-com.translate.google.com/science/article/abs/pii/S2214785322014936?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=uk&_x_tr_pto=sc). (дата звернення 02.09.2024).
9. Академія Хана. «Що таке інсульт?». Khan Academy - Здоров'я та медицина. UR: <https://www.khanacademy.org/science/health-and-medicine/circulatory-system-diseases/stroke/a/what-is-a-stroke> (дата звернення 12.11.2024).
10. Software Quality Measurement Analysis based on Techniques, Criteria, Metrics, Models, and Datasets. <https://ieeexplore.ieee.org/document/10031990/metrics#metrics>. (дата звернення 02.09.2024).
11. Л. Лі та ін. Глибоке навчання для виявлення та сегментації геморагічних уражень на КТ-зображеннях мозку, журнал *IEEE Journal of Biomedical and Health Informatics*, vol. 25, вип. 5, стор. 1646-1659, 2021, doi: 10.1109/JBHI.2020.3028243.

12. The Impact of Machine Learning on Programming. (дата звернення 03.09.2024).
13. J. Hofmeister, J. Siegmund, and D. V. Holt. Shorter identifier names take longer to comprehend. In 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), pages 217–227, 2019.
14. К.-Ф. Лю та ін. «Виявлення на основі глибокого навчання та сегментація аномалій дифузії при гострому ішемічному інсульті», Communications Medicine, vol. 1, № 1, 2021, doi: 10.1038/s43856-021-00062-8
15. С. Флемінг. Це 10 головних світових причин смерті, але дві хвороби зменшуються. Всесвітній економічний форум. URL: <https://www.weforum.org/agenda/2021/02/top-10-global-causes-death/> (дата звернення 12.11.2024).
16. Stafford, I.; Kellermann, M.; Mossotto, E.; Beattie, R.; MacArthur, B.; Ennis, S. A systematic review of the applications of artificial intelligence and machine learning in autoimmune diseases. NPJ Digit. Med. 2020, 3, 1-11.
17. Tamilla A. Bayramova, Nazakat C. Malikova. Developing a conceptual model for improving the software system reliability Problems of Information Society, 2024, vol.15, no.1, 42-56.
18. What Are Nervous System (Neurological) Diseases? <https://www.healthline.com/health/nervous-system-diseases> (дата звернення 12.11.2024).
19. Kazman, R., Bianco, P., Ivers, J., & Klein, J. (2020). Integrability. CMU/SEI-2020-TR-001. Software Engineering Institute, Carnegie Mellon University. URL: <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=637375> (дата звернення 23.10.2024).
20. Харченко В. О. Основи машинного навчання : навч. посіб. / В. О. Харченко. – Суми : Сумський державний університет, 2023. – 264 с.
21. Zhu Yan. Supervised machine learning approaches: a survey. April 2015. International Journal of Soft Computing. 5(3):946-952.
22. L. Song, C. He, L. Xu і B. Zheng, «CapGAN: медичні субтитри до зображень. Клінічний стиль і збереження патології за допомогою умовних генеративних

змагальних мереж», 2022 р. 5-та Міжнародна конференція з розпізнавання образів і штучного інтелекту. (PRAI), 2022.

23. К.-Ф. Лю та ін. «Атлас артеріальних територій цифрового 3D МРТ мозку», Scientific Data, том. 10, № 1, 2023.

24. С. Yin та ін. «Автоматичне створення звіту про діагностику медичного зображення з ієрархічною рекурентною нейронною мережею», Міжнародна конференція IEEE з інтелектуального аналізу даних (ICDM), 2019 р.

25. J. Ramos та ін. «Отримання зображень на основі вмісту за допомогою метричних досліджень із радіологічних звітів: застосування до інтерстиціальних захворювань легенів», IEEE Journal of Biomedical and Health Informatics, vol. 20, № 1, стор. 281-292, січень 2016.

**ДОДАТОК А**  
(обов'язковий)

**КОПІЇ НАУКОВИХ ПУБЛІКАЦІЙ**

УДК 004.9

Ямборко Д.А., Кустовський Р.С., Яшина О.М.

*Хмельницький національний університет*

## **МЕТОД ПІДТРИМКИ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МАШИННОГО НАВЧАННЯ**

*Розглянуто задачу підтримки якості програмного забезпечення на основі машинного навчання, що дозволяють здійснювати правильну розробку програмного забезпечення.*

*The article considers the task of maintaining software quality based on machine learning, which allows for correct software development.*

Машинне навчання (МН) є галуззю використання штучного інтелекту, яка зосереджена на використанні даних і алгоритмів, щоб дозволити ШІ імітувати спосіб навчання людей, поступово підвищуючи його точність.

Машинне навчання розбиває алгоритм на три основні частини: прийняття рішень, прогнозування моделі (виявлення помилок), оптимізація моделі.

1. Процес прийняття рішення. Загалом алгоритми машинного навчання використовуються для прогнозування чи класифікації. На основі деяких вхідних даних, які можуть бути позначеними чи не позначеними, ваш алгоритм вироблятиме оцінку шаблону в даних.

2. Функція помилок: функція помилок оцінює прогноз моделі. Якщо є відомі приклади, функція помилок може зробити порівняння, щоб оцінити точність моделі.

3. Процес оптимізації моделі: якщо модель може краще відповідати точкам даних у навчальному наборі, тоді вагові коефіцієнти коригуються, щоб зменшити розбіжність між відомим прикладом і оцінкою моделі. Алгоритм повторюватиме цей ітеративний процес «оцінки та оптимізації», автономно оновлюючи рівновагу, доки не буде досягнуто порогу точності.

Інструменти для перегляду коду на основі штучного інтелекту видозмінюють методи та засоби розробки програмного забезпечення, пропонуючи розробникам спосіб підвищити якість та ефективність коду, а разом із тим і якість програмного забезпечення загалом. Ці інструменти використовують алгоритми машинного навчання для аналізу коду, виявлення проблем і надання пропозицій щодо покращення в режимі реального часу. Автоматизувавши процес перегляду коду, розробники можуть заощадити час, зменшити кількість помилок і підтримувати єдині стандарти кодування у своїх проектах.

Однією з ключових переваг таких перевірок коду є їхня здатність виявляти помилки кодування, вразливості безпеки та проблеми з ефективністю, які можуть залишитися непоміченими під час ручних перевірок. Ці інструменти не лише виявляють проблеми, але й пропонують пропозиції щодо оптимізації структури коду та впровадження найкращих практик. Різноманітні інструменти на основі штучного інтелекту надають пропозиції в режимі реального часу, щоб підвищити якість коду та оптимізувати робочі процеси розробки.

Інструменти для перевірки коду зі штучним інтелектом працюють, аналізуючи вихідний код програмного проєкту за допомогою алгоритмів машинного навчання та великих бібліотек коду. Вони надають миттєвий зворотний зв'язок і рекомендації, допомагаючи розробникам поліпшити цілісність коду і дотримуватися встановлених норм кодування. Щоб інтегрувати ШІ в робочий процес, розробники можуть вибрати відповідний інструмент, ознайомитися з його можливостями і поступово впроваджувати його в свої повсякденні завдання. Це передбачає інтеграцію інструменту в середовище розробки відповідно до специфічних вимог проєкту, а також регулярний моніторинг його пропозицій на відповідність нормам кодування.

Хоча інструменти для перегляду коду ШІ пропонують значні переваги, вони мають також недоліки та обмеження. Такі проблеми, як розуміння контексту і намірів, а також усунення помилкових спрацювань, вимагають збалансованого підходу, що поєднує ШІ з людським досвідом. Використовуючи сильні сторони як ШІ, так і людський потенціал, розробники можуть максимізувати ефективність і якість коду, одночасно долаючи технічну несправність та помилки.

Отже, використання машинного навчання для підтримки якості програмного коду є актуальним питанням у галузі штучного інтелекту.

#### **Перелік посилань**

1. ISO/IEC 9126-1:2001. Software engineering – Software product quality – Part 1: Quality model.
2. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. URL: [Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions | SN Computer Science \(springer.com\)](#) (дата звернення 30.08.2024).
3. Improving Code Quality Through Predictive Analytics with KINE: A New Frontier. URL: <https://www.linkedin.com/pulse/improving-code-quality-through-predictive-analytics-kine-new-ttgsc> (дата звернення 12.09.2024).

**ДОДАТОК Б**  
(обов'язковий)

**ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ**

# Метод підтримки якості програмного забезпечення для діагностики захворювань нервової системи на основі машинного навчання

Автор роботи:

студент групи ІПЗм-23-1 Ямборко Д.

Керівник роботи:

к.т.н., доцент Яшина О.М.

Рисунок 1 - Слайд 1. Тема роботи

## Актуальність

Актуальність даного дослідження полягає у тому, підтримка якості програмного забезпечення відіграє важливу роль у повсякденному житті людей. Але в деяких випадках може бути критичним фактором, особливо, якщо це стосується життя та здоров'я. Від якісної діагностики залежить подальше лікування і відповідно пов'язані із цим наслідки. Щоб забезпечити якість програмного забезпечення широко використовують штучний інтелект. У даній кваліфікаційній роботі поєднано підтримку якості програмного забезпечення діагностичного спрямування із використанням методів та засобів штучного інтелекту, зокрема машинного навчання.

Загалом багато науковців займаються дослідженням питань пов'язаних із якістю програмного забезпечення, що використовує штучний інтелект, зокрема такий його аспект як машинне навчання.

Рисунок 2 – Слайд 2. Актуальність дослідження

**Аналіз останніх досліджень, що проводились у даному аспекті показують такі підходи:**

- **Лаврішєва К.М.**, яка займалась питаннями програмної інженерії та якістю програмного забезпечення зокрема;
- **Коцовський В.М.** розглядав супровід програмних систем в контексті якості.
- **Лю** та інші пропонують автоматизований підхід, заснований на методі глибокого навчання для налагодження назви на основі узгодженості між назвою методу та його реалізацією.
- **Manjural Ahsan, Shahana Akter Luna, Zahed Siddique** вивчали системи, методи діагностики та моніторингу розладів нервової системи пацієнта з використанням систем на основі штучного інтелекту

Рисунок 3 – Слайд 3. Аналіз останніх досліджень

**Аналіз останніх досліджень, що проводились у даному аспекті показують такі підходи:**

- **Cinzia Volonte** та інші розглядали машинне навчання як новий інструмент у профілактиці, діагностиці та лікуванні неврологічних захворювань.
- **Ana Fatima** та інші вивчають підходи машинного навчання для прогнозування неврологічних захворювань.
- **Radha Raman Chandan, Jagendra Singh** та інші зробили огляд впливу машинного навчання на діагностику та прогнозування захворювань з погляду якості програмних застосунків.

Рисунок 4 - Слайд 4. Аналіз останніх досліджень

**Мета:** удосконалення методики використання машинного навчання для підтримки якості програмного забезпечення діагностичного спрямування.

**Завдання дослідження:**

1. Здійснення аналізу предметної області досліджуваної проблеми.
2. Здійснення дослідження методів машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.
3. Удосконалення методу підтримки якості програмного забезпечення на основі машинного навчання.
4. Розроблення алгоритму роботи системи підтримки якості програмного забезпечення на основі машинного навчання.

Рисунок 5 - Слайд 5. Мета та завдання дослідження

**Об'єкт дослідження:** процес використання методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.

**Предмет дослідження:** методи машинного навчання, що використовуються для підтримки якості програмного забезпечення діагностичного призначення.

Рисунок 6 – Слайд 6. Об'єкт та предмет дослідження

## Наукова новизна

1. Удосконалення методу машинного навчання для підтримки якості програмного забезпечення діагностичного призначення.
2. Розроблено алгоритм роботи методу підтримки якості програмного забезпечення на основі машинного навчання.

Рисунок 7 – Слайд 7. Наукова новизна

## Практична значимість

У цій кваліфікаційній роботі показано удосконалення методу машинного навчання для якості програмного забезпечення діагностичного призначення. Результатом даного дослідження став алгоритм роботи методу підтримки якості програмного забезпечення на основі машинного навчання. Це в свою чергу дозволить програмістам використовувати даний метод та алгоритм під час розробки програмного забезпечення в медичній галузі для продуктів діагностичного спрямування. Потрібність і актуальність подібних досліджень продемонстровано у перших двох розділах.

Рисунок 8 – Слайд 8. Практична значимість

## Концептуальна модель



Рисунок 9 – Слайд 9. Концептуальна модель

## Метод підтримки якості програмного забезпечення на основі машинного навчання

Полягає у тому, що введено датасети, що містять зображення МРТ сегментації ураження мозку. На основі цього було здійснено розробку алгоритму підтримки якості програмного забезпечення на основі машинного навчання.

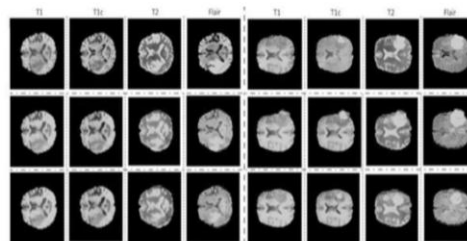


Рисунок 10 - Слайд 10. Удосконалений метод

## Алгоритм підтримки якості

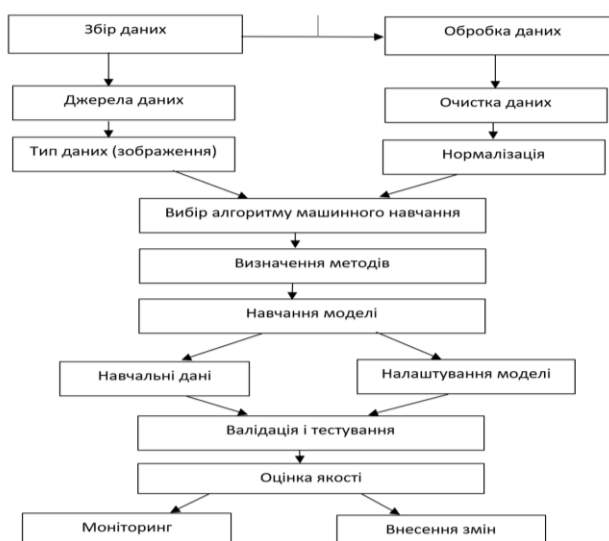


Рисунок 11 – Слайд 11. Алгоритм підтримки якості

## Архітектура програмної системи



Рисунок 12 – Слайд 12. Архітектура програмної системи

## Результати

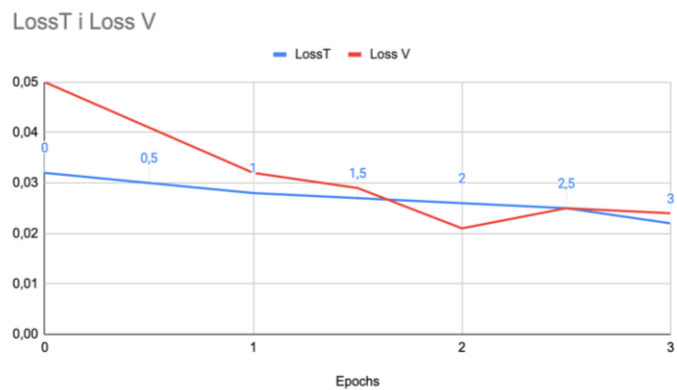


Рисунок 1

Рисунок 13 – Слайд 13. Результаты

## Результати

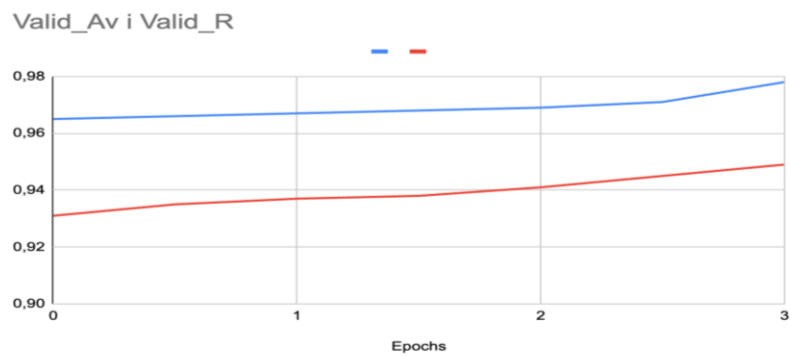


Рисунок 2

Рисунок 14 - Слайд 14. Результаты

## Результати

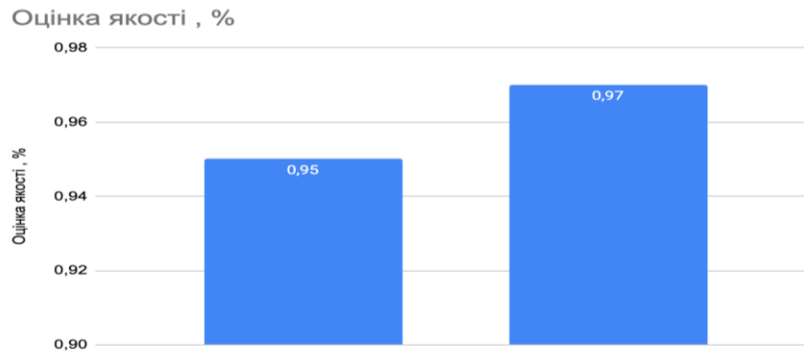


Рисунок 3

Рисунок 15 – Слайд 15. Результати

## Публікації за темою роботи

- За темою дипломної роботи опубліковано тези доповіді конференції АПКН-2024 Хмельницького національного університету «Метод підтримки якості програмного забезпечення на основі машинного навчання».

Рисунок 16 – Слайд 16. Публікації за темою роботи

## Висновки:

1. Здійснено якісний аналіз предметної області.
2. Проаналізовано існуючі методи підтримки якості програмного програмного забезпечення діагностичного призначення на основі машинного навчання
3. Проведено удосконалення методу підтримки якості програмного забезпечення діагностичного призначення із використанням штучного інтелекту.
4. Розроблено алгоритм роботи системи підтримки якості програмного забезпечення діагностичного призначення із використанням машинного навчання.

Рисунок 17 – Слайд 17. Висновки

Завідувачу кафедри  
інженерії програмного забезпечення  
проф. Леоніду БЕДРАТЮКУ  
студента групи ІПЗм-23-1

\_\_\_\_\_  
Ім'я, ПРІЗВИЩЕ

### ЗАЯВА

Прошу закріпити за мною тему кваліфікаційної роботи освітнього ступеня «магістр» за спеціальністю 121 «Інженерія програмного забезпечення»:

Метод підтримки логіки програмного забезпечення  
для діагностики заборговань керувані системи  
на основі машинного навчання

(керівник кваліфікаційної роботи – Оксана Липина)  
Ім'я, ПРІЗВИЩЕ

02.09.2024  
Дата

  
Підпис здобувача

Завідувачу кафедри інженерії програмного  
забезпечення проф. Леоніду БЕДРАТЮКУ  
здобувача вищої освіти  
**Дмитра ЯМБОРКА**  
факультет ІТ, 2 курс, група ІПЗм-23-1

### ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності в Хмельницькому національному університеті, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

02.09.2024  
дата

  
підпис

**ДЕКЛАРАЦІЯ УЧАСНИКА ОСВІТНЬОГО ПРОЦЕСУ**

**щодо дотримання академічної доброчесності**

Цією декларацією я, Ямборко Дмитро Анатолійович,

студент II курсу спеціальності 121 – Інженерія програмного забезпечення,  
група ІІЗм-23-1

здобувач вищої освіти (шифр та назва спец-ті, курс, академічна група)

підтверджую, що ознайомився з Положенням про систему забезпечення академічної доброчесності у Хмельницькому національному університеті та Кодексом академічної доброчесності і **зобов'язуюсь** дотримуватися їх вимог під час освітнього процесу, проведення наукової діяльності, виконання організаційно-адміністративних функцій тощо.

**Усвідомлюю**, що у разі порушення мною принципів академічної доброчесності нестиму відповідальність перед академічною спільнотою ХНУ згідно з нормами, визначеними Положенням про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, законодавства України.

1 вересня 2023 р.

  
Підпис

# Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 6.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилоч в документах: 6%

ID: 154466 Назва: МКР_Метод підтримки якості програмного забезпечення для діагностики захворювань нервової системи на основі машинного навчання Додано в БД: 2024-12-04 Автора: Дмитро ЯМБОРКО Керівники: к. техн. наук, доцент Оксана ЯШИНА Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	86348	1211	6425 (7%)	86 (7%)

## Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

## Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Дмитро ЯМБОРКО

**Співавтор:**

**Назва:** Метод підтримки якості програмного забезпечення для діагностики захворювань нервової системи на основі машинного навчання

**Науковий керівник:** к. техн. наук, доцент Оксана ЯШИНА

**Підрозділ:** Кафедра інженерії програмного забезпечення

**Коефіцієнт подібності 1:** 10.5%

**Коефіцієнт подібності 2:** 5%

**Мікропробіли:** 0

**Заміна букв:** 0

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2024-12-05 10:03:38.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

05.12.2024

Дата

  
експерт

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів подібності щодо роботи, продукуваними програмно-технічним засобом(ами) перевірки текстів на плагіат.

Назва: «Метод підтримки якості програмного забезпечення для діагностики захворювань нервової системи на основі машинного навчання»

Автор: Ямборко Дмитро Анатолійович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Яшина Оксана Миколаївна кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<b>відповідає</b>
2	Виявлені запозичення не є плагіатом, розміщені у розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за два дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені у розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені у роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системою перевірки на плагіат StrikePlagiarism виявлено схожість з деякими документами у частині загальноновживаних обов'язкових словосполучень у стандартних бланках (титулка, бланк завдання), у структурі змісту, у назвах розділів/підрозділів, у назвах публікацій переліку джерел посилання тощо;

2) в якості запозичень системою StrikePlagiarism було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) запозичення, виявлені в тексті роботи, є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Максимальний обсяг запозичень, визначений системою Anti-Plagiarism, складає 6%. За системою StrikePlagiarism коефіцієнт подібності 1 складає 105%, коефіцієнт подібності 2 складає 5%.

Дата \_\_\_\_\_

Завідувач кафедри ІПЗ

Гарант освітньої програми

Керівник кваліфікаційної роботи

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Леонід БЕДРАТЮК

Оксана ЯШИНА

Леонід БЕДРАТЮК

**РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**  
**освітнього ступеня «Магістр»**

Дипломник Ямборко Дмитро Анатолійович

Тема Метод підтримки якості програмного забезпечення для діагностики захворювань нервової системи на основі машинного навчання

Спеціальність 121 – Інженерія програмного забезпечення

**Обсяг кваліфікаційної роботи:**

Кількість листів креслень \_\_\_\_\_; кількість сторінок записки 83

1. Короткий зміст пояснювальної записки та прийнятих рішень. У кваліфікаційній роботі було досліджено предметну область та проведено аналіз існуючих актуальних рішень в сфері діагностики захворювань нервової системи. Визначено недосліджені частини у сфері, проаналізовано як покращити існуючі рішення. У рамках роботи було сформовано технічне завдання, удосконалено метод машинного навчання для підтримки якості програмного забезпечення діагностичного призначення, розроблено алгоритм роботи методу підтримки якості програмного забезпечення на основі машинного навчання.

2. Висновок про відповідність проекту поставленому завданню. Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу проекту, ступінь використання останніх досягнень науки і техніки та передових методів роботи. У вступі доведено актуальність теми, визначено мету та завдання кваліфікаційної роботи. У першому розділі проведено аналіз предметної області, розглянуто існуючі рішення та визначені функціональні і не функціональні вимоги до розроблюваного програмного забезпечення. У другому розділі проведено аналіз сучасних концепцій методів та рішень, розглянуто їх переваги і недоліки. У третьому розділі було детально описано проектування. У четвертому розділі підготовлено всі залежності для написання коду та виконано практичну розробку програмних модулів і описано їх особливості, було виконано тестування системи та проведено її у відповідності до функціональних вимог, в результаті чого було підтверджено доцільність використання методу.

4. Позитивні сторони проекту. Тематика дипломного проекту є актуальною, що було підтверджено великою кількістю сучасних досліджень в даній сфері. Під час проектування було досліджено актуальні дослідження та наведено приклад реалізації.

5. Негативні сторони проекту:

незрозумілим є процес формування датасетів. У роботі також не описано як проходило навчання моделі.

---

---

6. Оцінка графічного оформлення та пояснювальної записки проекту Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Викладення матеріалу пояснювальної записки є структурованим, послідовним та чітким, що дозволяє зрозуміти викладений матеріал у рамках тематики кваліфікаційного проекту. Графічний матеріал надає можливість наочно побачити деталі проектування методу. Однак, робота містить певні неточності та недоліки.

---

---

---

8. Інші зауваження \_\_\_\_\_

---

---

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі, однак містить певні недоліки та неточності та заслуговує на оцінку «задовільно».

---

---

РЕЦЕНЗЕНТ к.т.н., доцент каф. КІІС ХНУ, Капустян М.В.

" 06 "

судити

2024 р.

М.В.

(підпис)