


## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі

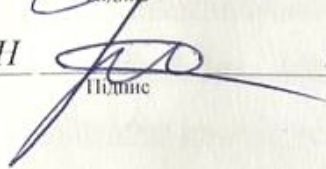
Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань


Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності

Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконала: студент групи КН-20-1  Арсен ГАМБАРЯН  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Керівник: док.філ., ст. викл. каф. КН  Павло РАДЮК  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:  
зав. кафедри КН, д.т.н., професор  Олександр БАРМАК  
Підпис Ім'я, ПРІЗВИЩЕ

24 серпня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор Олександр БАРМАК

« 16 » 02 2024 року

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі»

2. Завдання видано студенту Арсену ГАМБАРЯНУ  
(Ім'я, прізвище)

3. Керівник роботи старший викладач кафедри КН Павло РАДЮК  
(посада, ім'я, прізвище)

4. Затверджено наказом університету від « 16 » 02 2024 р. № 8


5. Дата видачі завдання студенту: « 16 » 02 2024 р.


6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета кваліфікаційної роботи бакалавра – покращення якості ідентифікації патологій серця за сигналом електрокардіограми. Метою роботи досягнуто внаслідок розв'язання таких завдань: проведено аналіз нейромережесевих методів для оброблення сигналів електрокардіограм; спроектовано спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі; спроектовано архітектуру настільного застосунок для програмної реалізації способу ідентифікації; розроблено настільний застосунок за спроектованим способом ідентифікації патологій серця; проведено експериментальне тестування створеного настільного застосунок за еталонними наборами даних.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2024	виконано
2	Формулювання мети та завдань дослідження, визначення об'єкта та предмета дослідження	січень 2024	виконано
3	Робота над розділом 1 – Аналіз нейромережесевих методів для ідентифікації патологій серця за його сигналами електрокардіограм	лютий 2024	виконано
4	Робота над розділом 2 – Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі	березень 2024	виконано
5	Робота над розділом 3 – Програмна реалізація способу ідентифікації патологій серця за сигналом електрокардіограми	квітень 2024	виконано
6	Складання пояснювальної записки, урахування коментарів керівника, оформлення роботи, відповідно до вимог	травень 2024	виконано
7	Створення презентації та попередній захист кваліфікаційної роботи	травень 2024	виконано
8	Захист кваліфікаційної роботи бакалавра	червень 2024	виконано

Виконавець: студента групи КН-20-1  Арсен ГАМБАРЯН  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Керівник: док.філ., ст. викл. каф. КН  Павло РАДІУК  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

## Анотація

Тема кваліфікаційної роботи бакалавра: «Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-20-1 Арсен ГАМБАРЯН

Керівник кваліфікаційної роботи бакалавра: доктор філософії, старший викладач кафедри КН Павло РАДЮК

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
67	25	11	30	2

Мета кваліфікаційної роботи бакалавра полягала у покращенні якості ідентифікації патологій серця за сигналом ЕКГ через проектування способу ідентифікації патологій серця за сигналом ЕКГ з використанням автокодувальної нейромережі та створення на його основі настільного застосунку. Для розроблення застосунку було використано мову програмування Python та технології Anaconda Navigator та Jupyter Environment.

Настільний застосунок дає змогу виявляти та аналізувати інформаційні аспекти ЕКГ-сигналів на основі автокодувальної нейромережі, що дає можливість лікарям краще розуміти результати аналізу та приймати інформовані рішення. Розроблений застосунок має потенціал поліпшити процес ранньої діагностики ССЗ.

Ключові слова: сигнал електрокардіограми, ідентифікація ЕКГ, автокодувальна нейромережа, настільний застосунок.

Виконавець: студент групи КН-20-1  
Група виконавця

  
Підпис

Арсен ГАМБАРЯН  
Ім'я, ПРІЗВИЩЕ

## Зміст

Перелік скорочень .....	4
Вступ.....	5
Розділ 1 Аналіз нейромережових методів для ідентифікації патологій серця за його сигналами електрокардіограм .....	7
1.1 Огляд теоретичних підходів до розв’язку подібних задач, що пов’язані з обробленням ЕКГ-сигналів .....	7
1.2 Огляд нейромережових методів та підходів до аналізу ЕКГ-сигналів.....	12
1.3 Аналіз програмних рішень для ідентифікації патологій серця за сигналами ЕКГ .....	16
1.4 Мета та задачі кваліфікаційної роботи .....	18
Розділ 2 Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі .....	20
2.1 Схема способу ідентифікації патологій серця за сигналом електрокардіограми.....	20
2.2 Проектування архітектури автокодувальної нейромережі для окреслення сигналів електрокардіограм .....	24
2.3 Проектування структури програмної реалізації способу ідентифікації патологій серця за сигналом електрокардіограми.....	31
2.4 Проектування бази даних програмної реалізації .....	36
2.5 Висновки до розділу 2 .....	39
Розділ 3 Програмна реалізація способу ідентифікації патологій серця за сигналом електрокардіограми.....	40
3.1 Структура та функціональне призначення складових програмної реалізації способу ідентифікації патологій серця .....	40
3.2 Вибір засобів розроблення для програмної реалізації способу ідентифікації у вигляді настільного застосунку .....	45
3.3 Особливості реалізації програмних складових настільного застосунку .....	46
3.4 Тестування програмної реалізації та основні вимоги до розгортання розробленого настільного застосунку.....	50
3.5 Функціональний опис створеного настільного застосунку .....	56
3.6 Висновки до розділу 3 .....	61

Загальні висновки.....	62
Перелік посилань.....	64
Додатки	

**Перелік скорочень**

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
АКН	Автокодуювальні нейромережі
БД	База даних
ЕКГ	Електрокардіограма
ССЗ	Серцево-судинні захворювання
ШІ	Штучний інтелект
(PTB-db)	PTB Diagnostic ECG Database
DenseNet	Densely Connected Convolutional Networks
ReLU	Rectified Linear Unit

## Вступ

Кваліфікаційна робота бакалавра присвячена покращенню якості ідентифікації патологій серця за сигналом електрокардіограми через проектування способу ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі та програмної реалізації на його основі у вигляді настільного застосунку.

**Актуальність.** Серцево-судинні захворювання (ССЗ) наразі є глобальною загрозою для здоров'я населення, будучи провідною причиною смертності та інвалідності у світовому масштабі. За статистикою Всесвітньої організації охорони здоров'я, у 2019 році ССЗ спричинили 17,9 мільйона смертей, що складає майже третину від загальної кількості летальних випадків. Ситуація в Україні є ще більш критичною, де на частку ССЗ припадає приблизно 60% усіх смертей. Вирішення цієї проблеми вимагає комплексного підходу, де ключову роль відіграють раннє діагностування та своєчасне лікування.

Електрокардіографія (ЕКГ) залишається одним з найпоширеніших та найефективніших методів діагностування серцево-судинних патологій, даючи змогу виявити широкий спектр захворювань, включаючи ішемічну хворобу серця, інфаркт міокарда та різноманітні аритмії. Традиційний аналіз ЕКГ, який виконується лікарями-кардіологами, є трудомістким та часозатратним процесом. У зв'язку з цим, розроблення та впровадження підходів до автоматизованої ідентифікації патологій серця на основі сигналу електрокардіограми з використанням автокодувальних нейронних мереж є вкрай актуальним та перспективним. Подібні підходи та способи можуть прискорити процес ідентифікації патологій серця, підвищити її точність та, як наслідок, покращити лікування ССЗ, що насамкінець сприятиме зниженню смертності від ССЗ.

**Об'єкт дослідження** – процес ідентифікації патологій серця за сигналом електрокардіограми.

**Предмет дослідження** – нейромережеві методи ідентифікації патологій серця.

**Мета кваліфікаційної роботи бакалавра** – покращення якості ідентифікації патологій серця за сигналом електрокардіограми.

**Завдання кваліфікаційної роботи бакалавра.** Провести аналіз нейромережових методів для оброблення сигналів електрокардіограм. Спроекувати спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі. Спроекувати архітектуру настільного застосунку для програмної реалізації способу ідентифікації патологій серця за сигналом електрокардіограми. Розробити настільний застосунок за спроектованим способом ідентифікації патологій серця. Провести експериментальне тестування створеного настільного застосунку за еталонними наборами даних. Продемонструвати покращення якості ідентифікації патологій серця за сигналом електрокардіограми через впровадження створеного настільного застосунку.

## **Розділ 1 Аналіз нейромережевих методів для ідентифікації патологій серця за його сигналами електрокардіограм**

### **1.1 Огляд теоретичних підходів до розв'язку подібних задач, що пов'язані з обробленням ЕКГ-сигналів**

Одним із найбільш поширених і безболісних методів визначення електричних сигналів, які виробляє серце, є електрокардіограма, або коротко ЕКГ. Електрокардіограма оцінює велику кількість захворювань або аномальну діяльність серця, яка передуює захворюванню [1]. Діаграми, зміни різниці потенціалів серця, які виникають під час збудження міокарда, можна використовувати для представлення вихідних результатів електрокардіограми. В середньому ці зміни зберігаються протягом 0,26–0,36 секунди.

Вихідні дані, також відомі як зміни потенціалів, можуть бути записані одна за одною або по кілька змін одночасно, в залежно від результатів електрокардіографу. Перш за все, ЕКГ проводять часто, щоб діагностувати або лікувати певні симптоми. У цьому випадку вік пацієнта не є визначальним для проведення тесту; однак за допомогою аналізу електрокардіограми вік пацієнта можна передбачити та визначити.

ЕКГ використовується для визначення наступного: аномалій у ритмі серця; звуження або закупорювання артерії серця, тобто порушення в серцевих м'язах; проблеми з метаболізмом; проблеми з провідністю; або збільшення відділів серця (шлуночки та передсердя) [2].

Крім того, зверніть увагу на будь-які незвичайні симптоми, такі як діабет, безпричинний пульс, больові відчуття в грудях, проблеми з диханням, скачки тиску та задишку під час спокою.

Загалом електрокардіограма займає лише 7 хвилин і складається з декількох етапів:

- а) оголити груди, передпліччя, зап'ястя та гомілку, лягти на кушетку;

б) для того, щоб електроди краще прилягали до шкіри, необхідно обробити місце прилягання електродів спиртом і спеціальним гелем;

в) закріплення манжети та електродів;

г) запуск електрокардіографу.

Ейнтховен записав першу ЕКГ у світі в 1903 році. Оскільки на той час не було доступних клейких електродів або посилок для системи, єдиним способом контакту з тілом було помістити кінцівки у відро з розчином солі. Ейнтховен зміг створити достатній контактний опір до тіла, щоб зробити першу ЕКГ видимою за допомогою струнного гальванометра. Ейнтховен міг виміряти напругу між правою і лівою рукою (відведення I), правою рукою та лівою ногою (відведення II), а також лівою рукою та лівою ногою (відведення III) [3].

Сьогодні дуже поширені електрокардіограми з одним відведенням. Ці системи можуть використовуватися для запису або дуже тривалого запису до 14 днів, або короткого запису) до кількох секунд. ЕКГ з одним відведенням не має медичного значення. Найпоширенішим методом запису 24-годинного показання є електрокардіограма з трьома підведеннями. Цілодобове читання, яке компенсується як довгострокове читання, є популярним методом діагностики серцевих проблем [4].

ЕМГ з дванадцятьма підведеннями поєднує системи відведення від Ейнтховена, Голдбергера та Вілсона, надаючи інформацію про горизонтальні та вертикальні осі. Стандартом для діагностики електрокардіограма у дванадцяти є електрокардіограма у спокої та під напругою.

Відведення, на базі яких формується система з 12:

- кінцівки: I, II, III, aVR, aVL, aVF;
- грудна клітина: V1, V2, V3, V4, V5, V6;
- виміряні відведення: I, II, III, V1, V2, V3, V4, V5, V6;
- розраховані відведення: aVR, aVL, aVF.

Графічне відображення електрокардіограми демонструє деякі скупчення зубців і відстані між ними. Ці зубці називають P, Q, R, S і T, на рисунку 1.1 зображено ознаки сигналу ЕКГ.

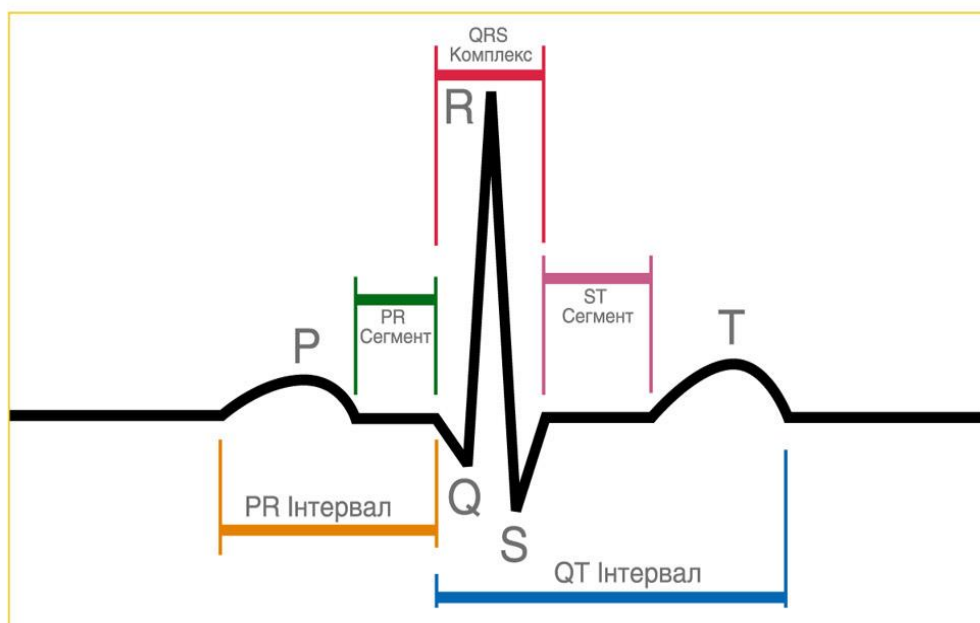


Рисунок 1.1 – Ознаки сигналу ЕКГ [5]

Відстань між піками електрокардіограми, які показують швидке переміщення реполяризації або депольоризації від однієї частини серця до іншої, називається інтервалами

Сегменти – це відстань на електрокардіографі, накладений на ізолінію, що показує відсутність різниці потенціалів [6]. Дані позначок показують імпульси, які генерує серце. Крім того, необхідно визначити, які елементи складають провідну систему серця. На рисунку 1.2 показана будова провідної системи серця.

1. Синоатріальний вузол, також відомий як СА-вузол, регулює ритм серця, створюючи синусовий ритм. Він виробляє імпульси з частотою 60–80 за хвилину. Цей відрізок буде знаходитися на ізолінії, оскільки збудження СА-вузла на електрокардіограмі не відображається.

2. Міжвузлові передсердні шляхи – наступний елемент провідної системи є між вузлові передсердні шляхи, які складаються з трьох шляхів: переднього Тореля, середнього Венкебаха та заднього Бахмана. Двома вузлами з'єднують ці шляхи: СА-вузол і АВ-вузол, також відомий як атріовентрикулярний вузол.

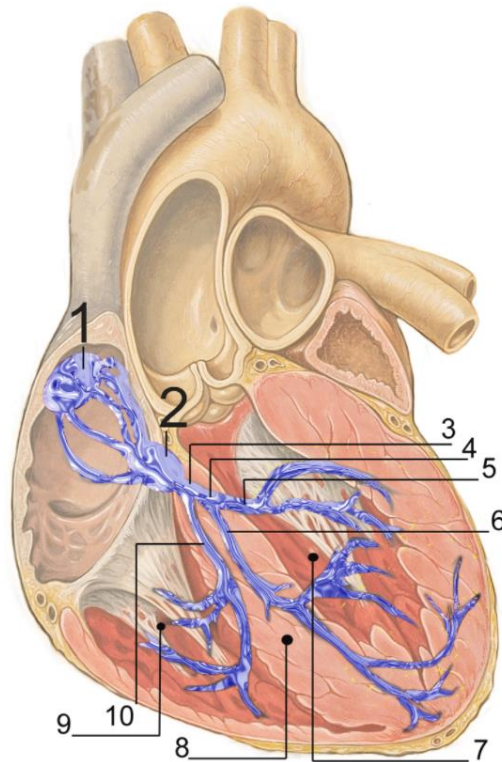


Рисунок 1.2 – будова провідної системи серця, де 1 – СА-вузол; 2 – АВ-вузол; 3 – пучок Гіса; 4 – ліва ніжка пучка Гіса; 5 – задня гілка лівої ніжки пучка Гіса; 6 – передня гілка лівої ніжки пучка Гіса; 7 – лівий шлуночок; 8 – міжпередсердна перегородка; 9 – правий шлуночок; 10 – права ніжка пучка Гіса [7]

3. Атріовентрикулярний вузол, також відомий як АВ-вузол, затримує імпульс збудження, який прямує до шлункових артерій. Він виробляє імпульси з частотою від сорока до шести десяти хвилин на хвилину. На електрокардіографі збудження, яке проходить через АВ-вузол, виглядає як сегмент (а не інтервал) PQ, розташований на ізолінії.

4. Пучок Гіса – це пучок, який знаходиться в між шлунковою перегородкою. Він виробляє імпульси з частотою від двадцяти п'яти до сорока хвилин за хвилину. Цей компонент має дві ніжки: праву та ліву.

5. Права ніжка пучка Гіса – діє на правий шлуночок, де вона збуджує міокард.

6. Ліва ніжка пучка Гіса – йде в лівий шлуночок і просвітлюється на дві вітки: передню і задню вітки лівої ніжки пучка Гіса.

7. Волокна Пуркіне – останній компонент системи серця

8. Імпульс, який проходить через ніжки пучка Гіса та збудження міокарда шлуночків, відображається на електрокардіограмі як QRS комплекс.

Реполіаризація передсердь відбувається одночасно з деполіаризацією шлуночків. Однак на електрокардіограмі ми бачимо лише збудження шлуночків, оскільки розслаблення передсердь накладається на QRS комплекс. На електрокардіограмі ми бачимо сегмент ST, коли шлуночки повністю деполіаризовані, а передсердя повністю реполіаризовані. Після цього відбувається реполіаризація шлуночків, і ми бачимо зубець T на електрокардіограмі.

Зубець P представляє деполіаризації міокарда передсердь. Такий зубець передує QRS комплексу і його амплітуда, тривалість, позитивність і негативність оцінюються. Зазвичай амплітуда цього зубця не перевищує 0.25 мВ, а вихід, який перевищує норму, вказує на гіпертрофію.

Інтервал PQ триває від 0,12 до 0,2 секунди з початку зубця P до початку зубця Q. Якщо він не існує, розшифровка цього інтервалу не відбувається на EKG.

QRS комплекс триває від піка зубця Q до піка зубця S і становить 0.11 секунд до 0.12 секунд можуть свідчити про інфаркт міокарда, гіпертрофію шлуночків, неповну блокаду та тривалість до 0.13 секунд можуть вказувати на блокаду лівої ніжки пучка Гіса.

Автокодуювальні нейромережі (АКН) є типом штучного інтелекту, який використовується для навчання на даних. АКН складаються з двох основних компонентів: кодера та декодера. Кодер приймає вхідні дані та перетворює їх у стиснене уявлення, яке називається кодом, тоді як декодер приймає цей код і відтворює з нього оригінальні вхідні дані [8].

Навчання АКН відбувається через процес, відомий як "неконтрольоване навчання". Це означає, що їм не надаються мітки даних, а вони самостійно знаходять закономірності в даних.

АКН можуть бути ефективно застосовані для аналізу ЕКГ завдяки їх здатності виявляти закономірності в даних. Це робить їх перспективним

інструментом для діагностики ССЗ. АКН можуть навчитися класифікувати ЕКГ на нормальні та патологічні, виявляти аномалії, які можуть бути ознаками ССЗ, та візуалізувати дані ЕКГ в новому форматі, що може допомогти лікарям у діагностиці захворювань.

Використання автокодувальних нейромереж для аналізу електрокардіограм відкриває нові можливості для медичної діагностики завдяки їх здатності автоматично знаходити та інтерпретувати складні патерни в сигналах. Це знижує залежність від ручної обробки та дає змогу отримувати точніші результати.

АКН можуть ефективно виконувати класифікацію ЕКГ. Наприклад, нейромережі можуть бути навчені розрізняти нормальні ЕКГ-сигнали від патологічних, таких як сигнали, що вказують на ішемію або інфаркт міокарда. Це досягається шляхом навчання на великому наборі даних, де кожен сигнал має відповідну анотацію.

Візуалізація даних ЕКГ з використанням АКН також надає важливі переваги. АКН можуть представляти дані в нових форматах, які спрощують інтерпретацію та аналіз [9, 10]. Це може включати візуалізацію закономірностей або аномалій, що робить процес діагностування більш прозорим і зрозумілим для лікарів.

## **1.2 Огляд нейромережевих методів та підходів до аналізу ЕКГ-сигналів**

Аналіз сигналів електрокардіограми є важливим етапом в медичній діагностиці серцевих захворювань. Різні методи та підходи використовуються для отримання інформації з ЕКГ-сигналів [11]. Їхні переваги та недоліки наведено в таблиці 1.1.

АКН можуть зберігати інформацію, необхідну для відновлення, одночасно зменшуючи розмір і шум вхідних даних. Автокодувальні мережі складаються з декодера та кодера. Кодер зменшує вхідні дані за допомогою

складного шару нейронів. Після цього декодер отримує код і використовує його, щоб відновити вихідні дані.

Таблиця 1.1 – Порівняння нейронних методів [12]

Назва методу	Переваги	Недоліки	Впровадження
Автокодуювальна нейронна мережа	Швидкість Спостереження за динамікою Автоматизація аналізу	Потрібен обширний обсяг даних	Стандартизація та регулювання Інтеграція
Рекурентна нейронна мережа	Швидкість	Не специфічність	Стандартизація та регулювання Інтеграція
Пряма нейронна мережа	Спостереження за динамікою	Не специфічність	Стандартизація Інтеграція з іншими системами
Машинне навчання	Автоматизація аналізу	Потрібен обширний обсяг даних	Стандартизація Інтеграція з іншими системами

Процес навчання автокодуювальної мережі використовує вхідні дані. Мета мережі полягає в тому, щоб визначити найефективніший метод стиснення та відновлення даних у декодері. У процесі навчання часто використовуються методи, спрямовані на зменшення розміру вихідних даних для зміни їх форми та зменшення шуму [13].

На рисунку 1.3 зображено архітектуру АКН.

Рекурентні нейронні мережі мають вузли. Кожен рекурентний вузол містить власну пам'ять.

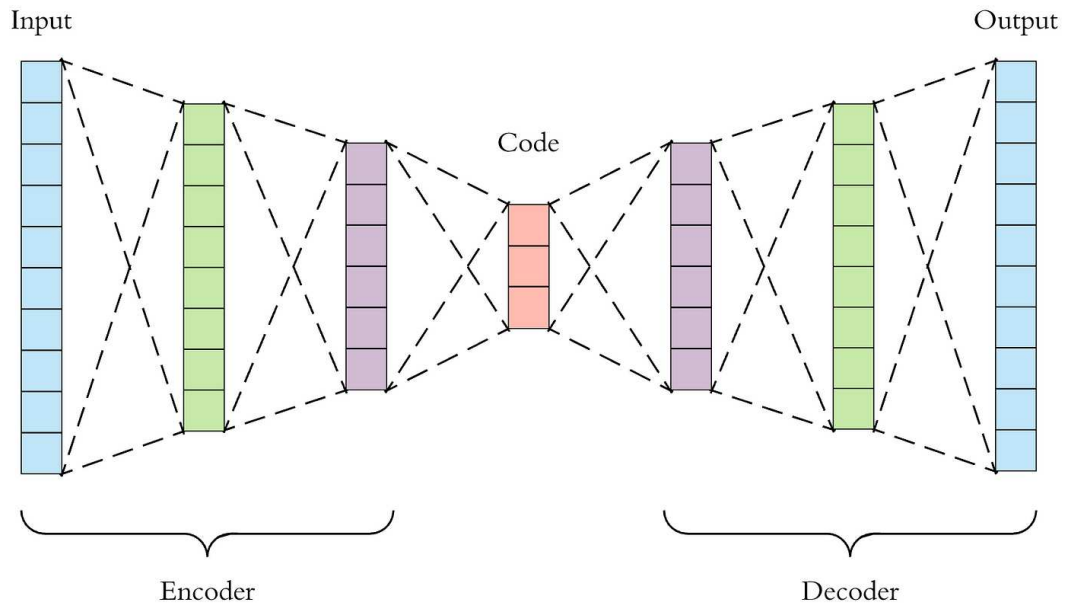


Рисунок 1.3 – Архітектура автокодувальної нейронної мережі [14]

Це дає змогу мережам зберігати дані з попередніх вхідних даних і використовувати їх для обробки поточного вхідного сигналу [15]. У зв'язку між конкурента вузлами мережа може використовувати внутрішню пам'ять для обробки послідовних вхідних даних.

Прямі нейронні мережі дозволяють передавати дані від вхідних вузлів до вихідних [16]. Такий тип мережі складається з шарів вузлів, де кожен вузол пов'язаний з кожним вузлом на попередніх і наступних шарах. Кожен вузол має зв'язки з вузлами попереднього шару. Вони мають здатність передавати сигнали з одного вузла до наступного.

У таблиці 1.2 наведено порівняння нейронних архітектур.

Densely Connected Convolutional Networks (DenseNet) відзначається можливістю навчання на немічених даних, здатністю добре генерувати дані, аналогічні тренувальним, та використовується для виявлення аномалій. Проте вона може бути чутливою до шуму в даних і не завжди забезпечує зрозумілі пояснення для своїх прогнозів.

U-Net, з іншого боку, має переваги у можливості моделювання залежностей в часових рядах і ефективності у прогнозуванні послідовностей, але може бути складним у навчанні та схильним до проблеми зникнення градієнта.

ResNet виділяється простотою, швидкістю та ефективністю, але не може моделювати залежності в часових рядах і часто не забезпечує задовільне генерування даних.

Таблиця 1.2 – Порівняння нейронних архітектур

Архітектура	Опис	Переваги	Недоліки
DenseNet [17]	Нейронна мережа, яка складається з двох частин: кодувальника та декодера. Кодувальник стискає вхідні дані в латентне представлення, а декодер відновлює вхідні дані з латентного представлення.	Можуть навчатися на немічених даних Добре генерують дані, що подібні до тренувальних Можуть використовуватися для виявлення аномалій	Можуть бути чутливі до шуму в даних Не завжди пояснюють свої прогнози
U-Net [18]	Нейронна мережа, в якій з'єднання між нейронами в одному або декількох шарах утворюють петлі.	Можуть моделювати залежності в даних з часовим рядом Добре підходять для прогнозування послідовностей	Можуть бути складними для навчання Схильні до зникнення градієнта
ResNet [19]	Найпростіший тип нейронної мережі, який складається з одного або декількох шарів нейронів, з'єднаних вперед.	Прості Швидкі Ефективні	Не можуть моделювати залежності в даних з часовим рядом Не завжди добре генерують дані

Проте після проведеного аналізу методів та архітектури, вибір ліг на автокодувальні нейронні мережі тобто DenseNet, які надзвичайно ефективні у розв'язанні задач типу реконструкції вхідних даних. Ці фактори включають ефективність моделі, здатність вивчати корисні представлення вхідних даних і можливість автоматично витягувати важливі ознаки. Їхня здатність це ефективно вивчати корисні представлення та неявні залежності в наборах даних, робити їх коректними для завдань, де важлива внутрішня представленість даних. Автокодувальні нейронні мережі, які мають здатність вивчати представлення, мають здатність автоматично витягувати важливі ознаки з вхідних даних шляхом навчання без нагляду. Це дає змогу їм створювати компактні та інформативні представлення даних.

### **1.3 Аналіз програмних рішень для ідентифікації патологій серця за сигналами ЕКГ**

На цей час існує кілька програмних рішень, які використовують автокодувальні нейромережі для ідентифікації патологій серця за сигналами електрокардіограми [20]. Ці рішення можуть варіюватися за функціональністю, точністю та можливостями інтеграції. Деякі з них можуть бути призначені для клінічного використання, тоді як інші можуть бути представлені у вигляді дослідницьких прототипів. Нижче наведено загальний огляд теперішніх програмних рішень у цій області.

CardioScape [21] – це платформа для обробки та аналізу ЕКГ-сигналів за допомогою нейромережі (рисунок 1.6). CardioScape використовує автокодувальні архітектури для екстракції важливих функцій з сигналів. Виявлення аномалій та патологій на основі автоматичного вивчення характеристик ЕКГ. Можливість інтеграції з іншими медичними інформаційними системами.

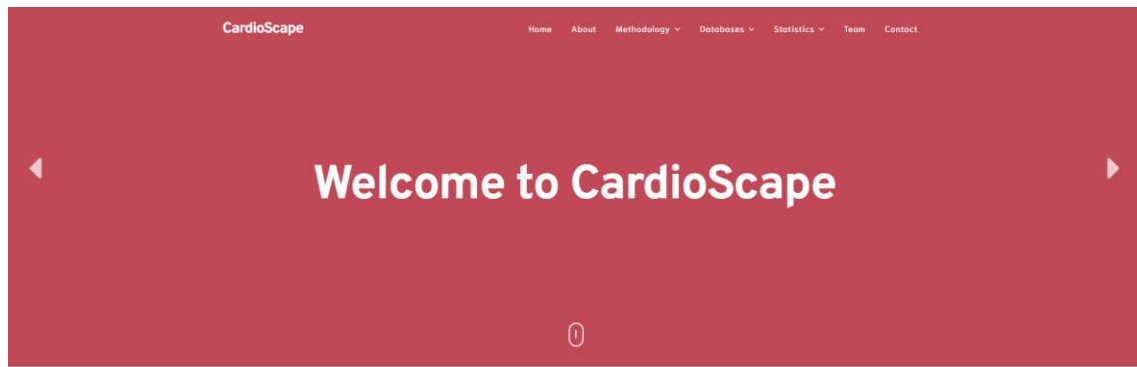


Рисунок 1.6 – Головна сторінка сайту CardioScape [21]

CardioNet [22] – це система для моніторингу ССР, яка використовує нейромережі для ідентифікації аномалій на основі даних ЕКГ (рисунок 1.7). Забезпечення високочутливого виявлення патологій та генерація рекомендацій для лікарів. Можливість інтеграції з мобільними пристроями для дистанційного моніторингу.

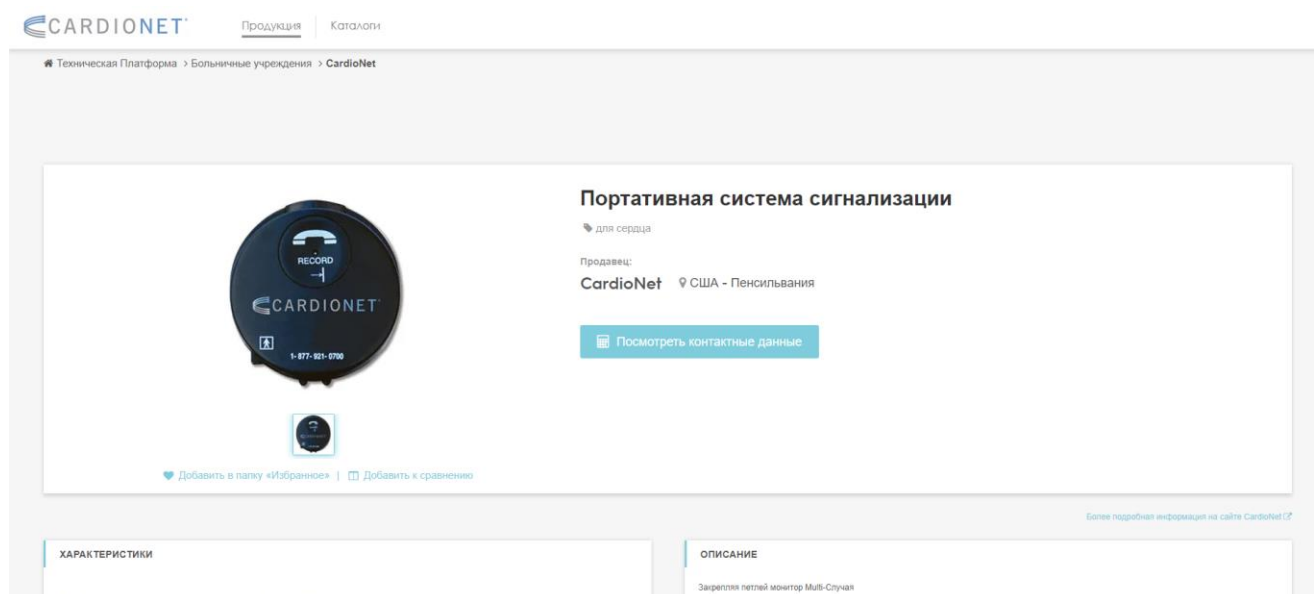


Рисунок 1.7 – Портативна система сигналізації CardioNet [22]

ECGNet [23] – дослідницький проект, що використовує глибокі нейромережі для автоматичного аналізу сигналів ЕКГ (рисунок 1.8). Виявлення

патологій та аномалій на різних етапах розвитку захворювань серця. Прототип з можливістю доповнення новими модулями для розширення функціональності.

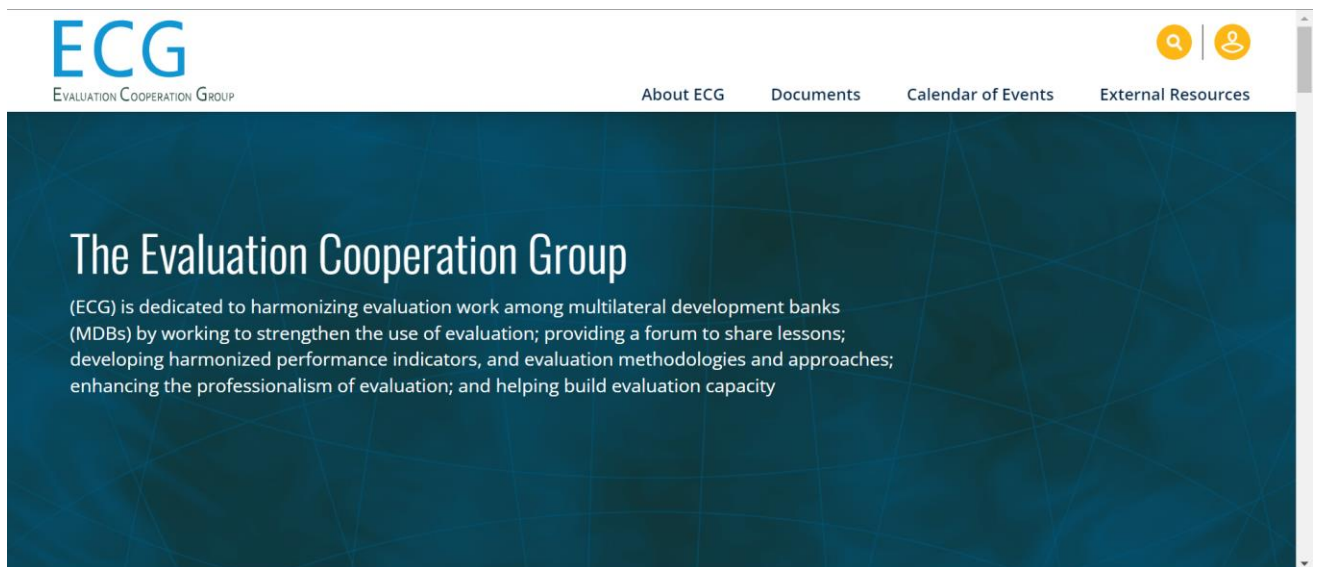


Рисунок 1.8 – Головна сторінка ECGNet [24]

Ці ресурси є гарним прикладом веб-сайтів які використовують штучний інтелект (ШІ) для глибокого автоматичного аналізу сигналів ЕКГ [25, 26]. Програмні рішення можуть мало інтегруватися з чинними медичними ІТ-системами, що ускладнює обмін інформацією та взаємодію з іншими медичними процесами.

З огляду на проведений аналіз методів та засобів ідентифікації запалення серця за ЕКГ-сигналами, встановлено, що низька точність ідентифікації, брак різноманітних даних для навчання систем штучного інтелекту та складна інтеграція з наявними медичними системами залишаються актуальними проблемами галузі. Тому, вирішення цих проблем сприятиме поліпшенню ефективності застосування таких рішень в медичній практиці.

#### **1.4 Мета та задачі кваліфікаційної роботи**

З огляду на проведений аналіз предметної області, прийшли до висновку, що ССЗ залишаються головною причиною смертності у світі, що підкреслює

критичну важливість раннього виявлення та точного діагностування цих патологій. Аналіз сигналів ЕКГ відіграє ключову роль у цьому процесі, проте традиційні методи часто виявляються недостатньо точними через необхідність залучення висококваліфікованих фахівців та значні часові затрати. Це створює суттєві обмеження для застосування та доступності якісної діагностики, особливо в умовах зростаючого навантаження на системи охорони здоров'я.

Впровадження сучасних технологій, зокрема АКН, для автоматичного аналізу ЕКГ сигналів відкриває нові перспективи у боротьбі з ССЗ. Ці передові підходи дають можливість підвищити точність діагностики та скоротити час обробки даних. Інтеграція таких технологій у медичні системи може значно покращити якість медичної допомоги, зменшити кількість помилкових діагнозів та підвищити загальну ефективність лікування ССЗ. З огляду на це, розроблення та вдосконалення способів ідентифікації патологій серця на основі аналізу сигналів електрокардіограми з використанням автокодувальних нейронних мереж є актуальним та перспективним напрямом досліджень, який може значно сприяти успішному діагностуванню пацієнтів із ССЗ.

Отже, на основі проведеного дослідження сформульовано мету кваліфікаційної роботи бакалавра, яка полягає у покращенні якості ідентифікації патологій серця за сигналом ЕКГ.

Досягнення мети роботи передбачає виконання наступних завдань:

1. Спроекувати спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі.
2. Спроекувати архітектуру настільного застосунку для програмної реалізації способу ідентифікації патологій серця за сигналом електрокардіограми.
3. Розробити настільний застосунок за спроектованим способом ідентифікації патологій серця.
4. Провести експериментальне тестування створеного настільного застосунку за еталонними наборами даних.

## Розділ 2 Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі

### 2.1 Схема способу ідентифікації патологій серця за сигналом електрокардіограми

Спосіб формалізації сигналу електрокардіограм складаються з чотирьох основних етапів. На рисунку 2.1 вказані основні етапи формалізації сигналу електрокардіограм.

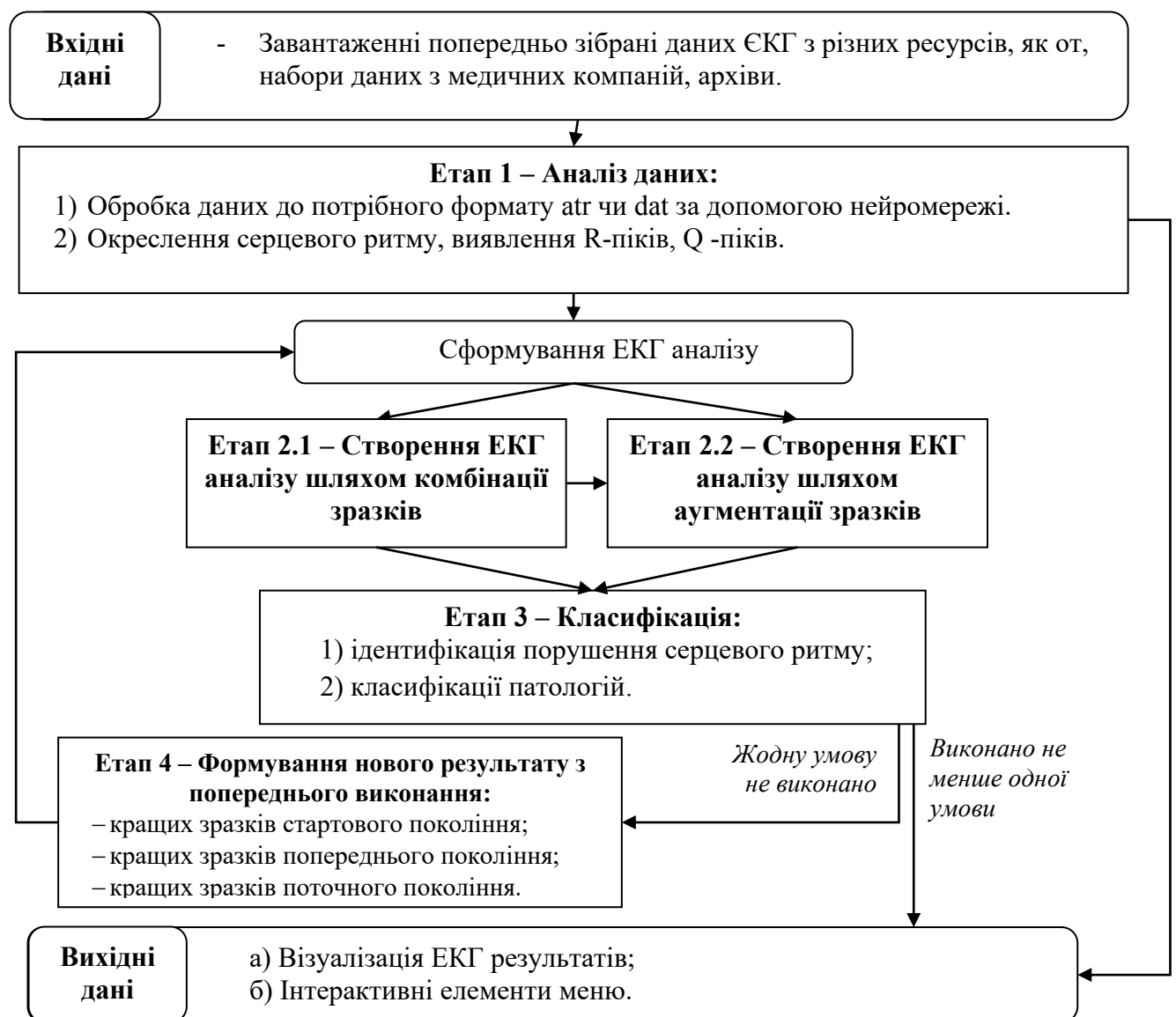


Рисунок 2.1 – Схема способу ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі

Для виконання цієї схеми основних складових настільного застосунку будуть зібрані дані ЕКГ з різних джерел, включаючи набори даних медичних компаній та архіви. Ці дані представляють собою записи електрокардіограм, що містять інформацію про серцевий ритм та можливі патології.

Етап перший, цей етап включає попередню обробку даних ЕКГ та їх підготовку для подальшого аналізу нейромережею. Спочатку зібрані дані ЕКГ необхідно перетворити у формат, придатний для аналізу в даному випадку це формат `atr` чи `dat`. Це може включати нормалізацію сигналів, фільтрацію шуму та інтерполяцію пропущених даних. Після попередньої обробки дані проходять через нейромережу, яка визначає серцевий ритм. Це включає виявлення ключових компонентів сигналу ЕКГ. Нейромережа ідентифікує R-піки (максимальні точки в QRS-комплексі) та Q-піки (мінімальні точки перед R-піками). Це є важливим кроком для подальшої класифікації та аналізу.

Етап два крок один починається з початкового вибору зразків для комбінації, де вибираються початкові зразки ЕКГ-аналітики з попередньої епохи або базового набору даних. Далі здійснюється комбінація зразків, яка включає використання комбінації різних частин обраних зразків для створення нових зразків. Це може включати змішування сегментів сигналів ЕКГ з різних зразків. Алгоритм комбінації передбачає об'єднання різних сегментів сигналу, наприклад, поєднання початку одного зразка з кінцем іншого. Після цього відбувається генерація нових зразків, що призводить до отримання нових зразків ЕКГ шляхом комбінування різних сегментів від обраних зразків. Потім проводиться перевірка якості зразків, де оцінюється якість нових зразків за допомогою попередньо встановлених критеріїв, а низькоякісні зразки, які не відповідають заданим вимогам, відсіюються. На завершення формується поточна популяція, де нові зразки інтегруються до поточної популяції для подальшої еволюції та аналізу.

Етап два крок два починається з початкового вибору зразків для аугментації, де вибираються початкові зразки ЕКГ-аналітики з попередньої епохи або базового набору даних. Потім здійснюється аугментація зразків, яка

включає використання методів аугментації даних для створення нових зразків. Це може включати різні перетворення, такі як зміна масштабу, додавання шуму, розтягування або стиснення сигналу. Алгоритм аугментації передбачає застосування різноманітних перетворень до існуючих зразків для створення нових варіацій. Після цього відбувається генерація нових зразків, що призводить до отримання нових зразків ЕКГ з використанням методів аугментації даних. Далі проводиться перевірка якості зразків, де оцінюється якість нових зразків за допомогою попередньо встановлених критеріїв, а низькоякісні зразки, які не відповідають заданим вимогам, відсіюються. На завершення формується поточна популяція, де нові зразки інтегруються до поточної популяції для подальшої еволюції та аналізу.

Етап третій класифікація, на цьому етапі дані, оброблені на першому етапі, використовуються для класифікації патологій та порушень серцевого ритму. Використовуючи оброблені дані, нейромережа аналізує серцевий ритм і виявляє можливі порушення, такі як аритмічні ритми, тахікардічні ритми. Далі нейромережа класифікує виявлені патології, відносячи їх до певних категорій, таких як інфаркт міокарда, гіпертрофія лівого шлуночка, фібриляція передсердь тощо. Це дає змогу лікарям точніше діагностувати стан пацієнта.

Після аналізу та класифікації результати представляються у зрозумілому форматі. Отримані результати візуалізуються у вигляді графіків ЕКГ, на яких відображені ключові точки, такі як R-піки та Q-піки, а також виділені ділянки з патологіями. Це допомагає лікарям швидко та ефективно оцінити стан серця пацієнта. Для зручності користувачів система надає інтерактивні елементи меню, що дозволяють лікарям взаємодіяти з результатами аналізу, деталізувати окремі ділянки ЕКГ, переглядати додаткову інформацію про виявлені патології та отримувати рекомендації щодо подальших дій

Щодо етапів підготовки способу:

1) збір даних: включає збір сигналів ЕКГ, отримання текстових описів медичних станів пацієнтів та збір інших медичних даних;

2) підготовка даних: включає масштабування, фільтрацію та корекцію сигналів ЕКГ, а також перетворення текстових даних для використання в машинному навчанні;

3) інтеграція з медичною системою: включає інтеграцію з додатком та надання лікарям доступу до результатів аналізу ЕКГ на їхніх робочих місцях.

Інтеграція аналізу сигналів ЕКГ з додатком значно полегшує лікарям отримання інформації про стан серцево-судинної системи пацієнтів. Це покращує їхні можливості зробити обґрунтований вибір під час діагностики та лікування, сприяє більшій прозорості та довірі до медичних рішень, а також покращує досвід медичного обслуговування. Така інтеграція допомагає сприяти розвитку більш ефективної та надійної системи охорони здоров'я.

У цьому розділі ми детально розглянули схему настільного застосунку для аналізу сигналів ЕКГ, що базується на інтелектуальному аналізі даних. Початково ми описали процес збору даних, який включає імпорт ЕКГ сигналів з різних джерел, таких як медичні бази даних та архіви. Ці дані є важливими, оскільки містять інформацію про серцевий ритм і можливі патології, які можуть бути виявлені під час аналізу.

Далі ми перейшли до першого етапу обробки даних, де розглянули попередню обробку сигналів ЕКГ. Цей процес включає нормалізацію сигналів, фільтрацію шуму та інтерполяцію пропущених даних. Після цього сигнали готуються для подальшого аналізу нейромережею. На цьому етапі нейромережа виконує визначення ключових компонентів сигналу ЕКГ, таких як R-піки та Q-піки, що є важливим для наступної класифікації і аналізу.

Другий етап ми поділили на дві підгрупи – перший включає комбінацію зразків для створення нових зразків ЕКГ, а другий – аугментацію існуючих зразків за допомогою різних методів. Обидва підходи спрямовані на збільшення різноманітності даних для навчання нейромережі та підвищення її загальної ефективності.

Третій етап – класифікація, включає в себе використання оброблених даних для виявлення та класифікації патологій серцевого ритму, таких як аритмії

та інші аномалії. Нейромережа аналізує відповідність виявлених патологій до певних медичних категорій, що допомагає лікарям швидше і точніше робити діагнози.

Усі результати аналізу та класифікації візуалізуються в зрозумілому форматі для лікарів. Це дає змогу швидко оцінювати стан серця пацієнта і приймати необхідні медичні рішення. Система також надає інтерактивні елементи, що спрощують взаємодію з результатами аналізу та забезпечують доступ до деталізованої інформації про виявлені патології.

Впровадження цієї настільного застосунку може значно поліпшити якість медичної діагностики і лікування, сприяючи більш прозорому та обґрунтованому підходу до медичних висновків. Інтеграція з медичними системами дає змогу миттєвий доступ до результатів аналізу для медичного персоналу, що підвищує ефективність та надійність медичного обслуговування пацієнтів.

## **2.2 Проєктування архітектури автокодувальної нейромережі для окреслення сигналів електрокардіограм**

У цій кваліфікаційній роботі розглядається використання нейромережі DenseNet для аналізу сигналів електрокардіограм. Використання DenseNet обґрунтовується її перевагами, що можуть призвести до поліпшення результатів у виявленні патологій серця.

DenseNet має кілька важливих переваг. По-перше, ефективність навчання моделі значно підвищується завдяки щільним зв'язкам між шарами, що сприяють полегшенню потоку градієнтів і покращують навчання. Це особливо корисно для задач аналізу ЕКГ, де дані можуть бути шумними та містити велику кількість інформації.

По-друге, DenseNet дає змогу створювати моделі значно більшої глибини, що сприяє навчанню складніших нелінійних залежностей у даних. Це важливо для виявлення тонких нюансів в сигналах ЕКГ, які можуть бути пов'язані з серцевими патологіями. По-третє, короткі зв'язки між шарами допомагають

зменшити проблему зникнення градієнта, що часто виникає в глибоких нейронних мережах, роблячи DenseNet більш стійким до шуму в даних та покращуючи його здатність до узагальнення. І нарешті, щільні зв'язки між шарами роблять DenseNet більш економічним з точки зору обчислювальних ресурсів, що важливо при використанні моделі на мобільних пристроях або в інших обмежених середовищах.

При Проектуванні архітектури DenseNet для окреслення сигналів ЕКГ слід врахувати кілька ключових факторів. Кількість блоків у DenseNet визначає глибину моделі: глибші моделі здатні навчатися складнішим залежностям, але можуть бути більш схильними до перенавчання. Розмір блоків визначає кількість з'єднань між нейронами в блоці: більші блоки дозволяють моделі навчатися складнішим локальним залежностям, але вони також можуть бути більш обчислювально дорогими. Кількість шарів у блоці визначає глибину кожного блоку: більша кількість шарів сприяє навчанню складніших залежностей, але збільшує ризик перенавчання.

Рисунок 2.2 демонструє структуру DenseNet, завдяки чому зрозуміло взаємодію між різними компонентами DenseNet під час окреслення сигналів ЕКГ, включаючи підготовку даних, навчання автокодера та аналіз реконструйованих сигналів.

DenseNet є типом глибокої згорткової нейронної мережі, яка відома своїми щільними зв'язками між шарами. Ця архітектура була розроблена для подолання проблем зникаючих градієнтів і поліпшення потоку градієнтів у глибоких нейронних мережах, що часто виникають під час тренування.

Основним компонентом DenseNet є блоки Dense, які є основними будівельними одиницями мережі. Кожен блок Dense складається з згорткового шару, за яким слідує шар активації (наприклад, ReLU) і шар групування функцій (наприклад, баченням по максимуму або середньому). Унікальність DenseNet полягає в тому, що вихід кожного шару Dense з'єднується з входами всіх наступних шарів Dense. Це створює щільні зв'язки, які дозволяють глибшим

шарам мережі отримувати доступ до градієнтів з більш раних шарів, підвищуючи швидкість збігання і покращуючи точність моделі.

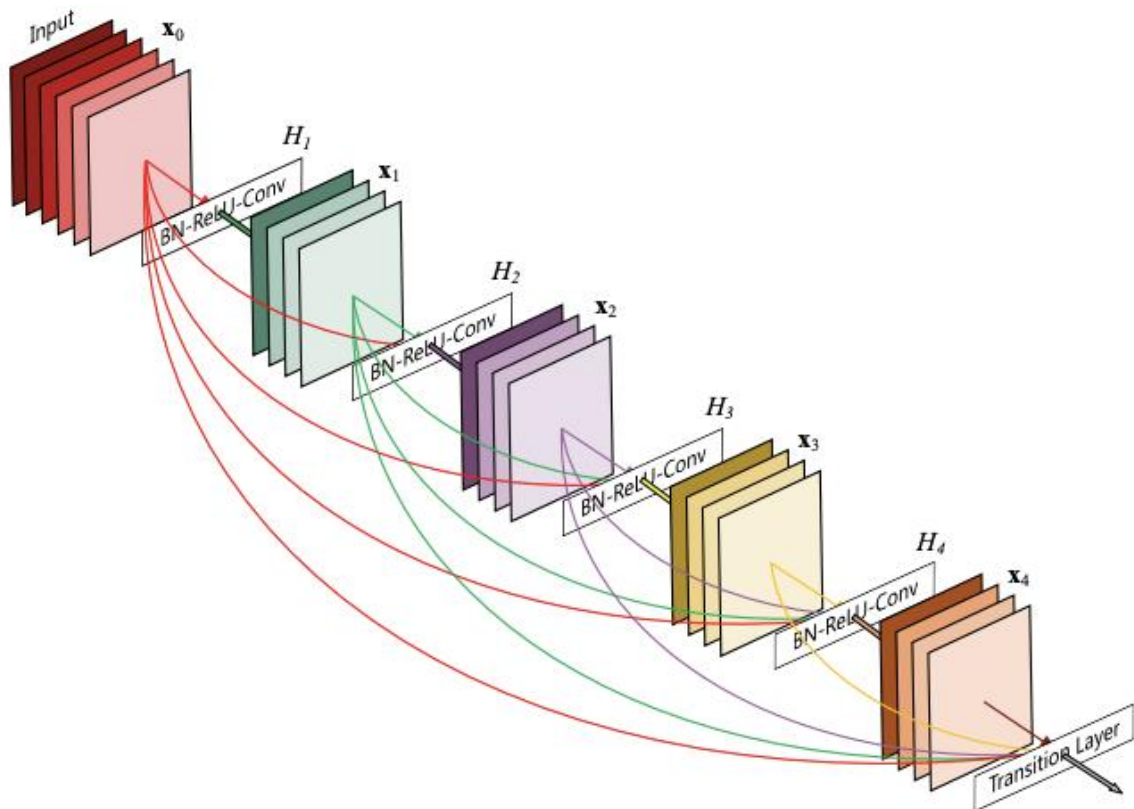


Рисунок 2.2 – Структура використаної архітектури АКН типу DenseNet [27]

Ще однією важливою складовою DenseNet є перехідні шари, які використовуються для зменшення розмірності карт активації між блоками Dense. Це зменшує обчислювальні витрати і сприяє збільшенню швидкості обробки сигналів у мережі. На завершення структури DenseNet включає класифікаційний шар, який використовується для остаточної класифікації вхідних даних після того, як вони пройшли через всі блоки Dense і перехідні шари.

В контексті застосування DenseNet для аналізу ЕКГ сигналів, процес може включати підготовку даних шляхом завантаження і передпроцесування ЕКГ сигналів, їх нормалізацію та сегментацію на окремі комплекси QRS. Ці сегменти перетворюються на зображення, які подаються на вхід DenseNet. Під час навчання DenseNet використовується як частина автокодера для

реконструкції вхідних зображень. Градієнти функції втрат використовуються для оновлення ваг мережі, що дає змогу досягти оптимальної точності реконструкції. Після навчання реконструйовані зображення QRS аналізуються для виявлення патологій і відхилень, що дає змогу ефективно класифікувати ЕКГ сигнали та забезпечувати точні медичні діагнози.

Переваги використання DenseNet для аналізу ЕКГ сигналів включають кращий потік градієнтів, що сприяє ефективній навчанні моделі, зменшення проблеми зникаючих градієнтів і покращення загальної точності класифікації. DenseNet виявив себе дуже ефективним для завдань обробки зображень і аналізу медичних даних, роблячи його важливим інструментом у сучасних дослідженнях та медичній діагностиці.

Діаграма станів відображає різні стани, в яких може перебувати DenseNet під час окреслення сигналів ЕКГ, що допомагає зрозуміти, як система переходить від одного стану до іншого протягом всього процесу. На рисунку 2.3 показана діаграма станів DenseNet.

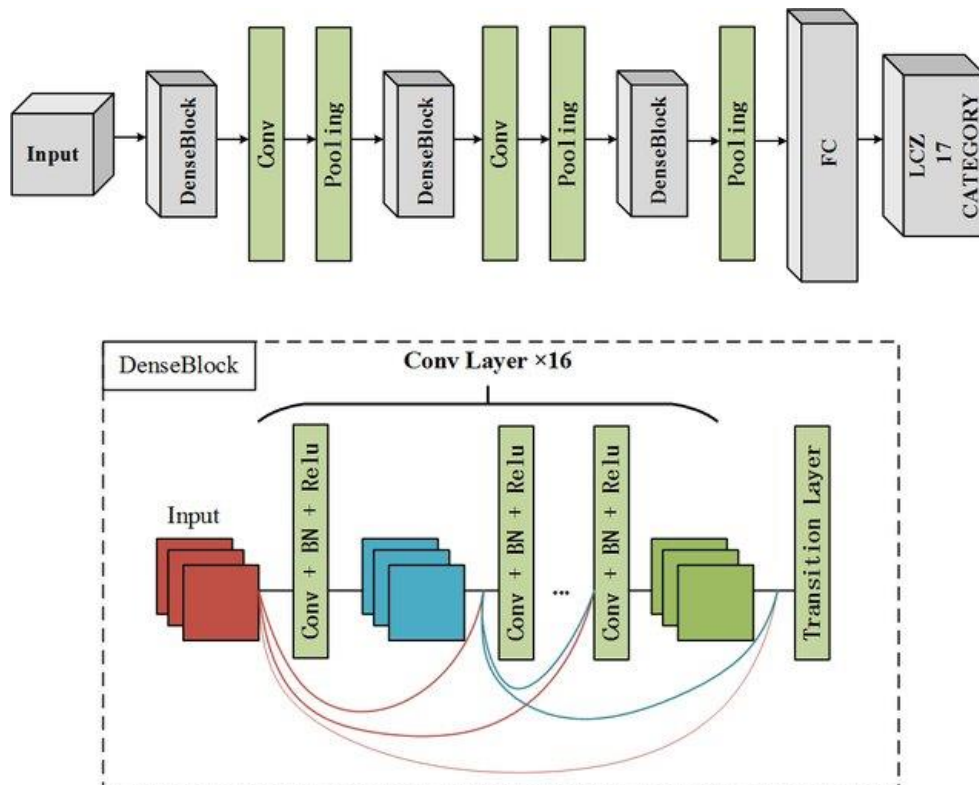


Рисунок 2.3 – Діаграма станів DenseNet [28]

На рисунку 2.3 зображено діаграму станів автокодувальної нейромережі для ідентифікації патологій серця за сигналами ЕКГ. В ній використовує щільні блоки (Dense Blocks) та перехідні шари (Transition Layers) для ефективного навчання та класифікації.

На самому початку подається вхідний сигнал ЕКГ, який далі проходить через серію обробних шарів. Щільний блок (Dense Block) Щільні блоки складаються з кількох згорткових шарів, де кожен шар отримує на вхід всі попередні шари в межах блока. Це дає змогу максимально використовувати інформацію і підвищувати ефективність навчання. Conv Layer x16: Це згорткові шари, які включають

Conv: Згортковий шар, який виділяє ознаки з сигналу ЕКГ. BN: Нормалізація пакетів (Batch Normalization), що стабілізує навчальний процес. ReLU: Активаційна функція ReLU, яка додає нелінійність і дає змогу моделі вивчати складні залежності.

Перехідний шар (Transition Layer) Між щільними блоками розташовані перехідні шари, які включають згортку 1x1, нормалізацію пакетів, активацію ReLU і пулінг для зменшення розмірності сигналу. Процес обробки DenseBlock: На першому етапі дані проходять через щільний блок, що включає кілька згорткових шарів з нормалізацією і активацією ReLU. Conv + Pooling: Після щільного блоку дані передаються через звичайний згортковий шар з наступним пулінгом (зазвичай максимальним пулінгом), який зменшує просторові розміри.

Повторення DenseBlock і Pooling: Цей процес повторюється кілька разів, де щільні блоки чергуються із згортковими і пулінг-шарами. FC (Fully Connected): На завершальному етапі дані проходять через повнозв'язний шар, де відбувається обробка для остаточної класифікації. Softmax/Category: Останнім шаром є Softmax або інша функція активації, яка генерує ймовірності для кожного класу патологій серця і здійснює остаточною класифікацію. Детальний опис щільного блоку (Dense Block) У збільшеному вигляді показано структуру щільного блоку (Dense Block):

Input: Вхідний сигнал подається на перший згортковий шар. Conv + BN + ReLU: Кожен згортковий шар складається зі згортки, нормалізації пакетів і активації ReLU. Concatenation: Результати кожного шару об'єднуються з усіма попередніми шарами. Transition Layer: Після кількох згорткових шарів йде перехідний шар, який зменшує розмірність і кількість параметрів.

Система автокодування DenseNet підтримуватиме різні сфери:

- аналіз ЕКГ для діагностики кардіоміопатій;
- моніторинг ЕКГ у режимі реального часу та дослідження ЕКГ.

У процесі діагностики кардіоміопатій DenseNet використовуватиметься для окреслення сигналів ЕКГ і виявлення ознак цього захворювання. Це включає аналіз різних показників, таких як форма та амплітуда QRS-комплексу, що може свідчити про наявність кардіоміопатії. Моніторинг у режимі реального часу передбачає використання DenseNet для безперервного аналізу сигналів ЕКГ, що дає змогу швидко виявляти ознаки аритмії та інших серцевих порушень. Це особливо важливо для пацієнтів з високим ризиком, оскільки своєчасне виявлення аритмій може запобігти серйозним ускладненням. Дослідження ЕКГ включає використання DenseNet для глибокого вивчення характеристик сигналів, що сприяє розробці нових методів діагностики й лікування ССЗ. Це може включати аналіз частоти серцевих скорочень, варіабельності серцевого ритму та інших параметрів.

Кожна з цих сфер включає кілька ключових функцій. Спочатку відбувається завантаження та візуалізація сигналів ЕКГ. Цей етап дає змогу користувачам бачити дані у зрозумілому форматі, що полегшує подальший аналіз. Потім налаштовуються параметри автокодера, включаючи архітектуру нейронної мережі, функції активації та оптимізатор. Це важливо для забезпечення ефективної роботи системи. Далі запускається процес окреслення сигналів ЕКГ, де DenseNet аналізує дані та кодує їх у компактні представлення, які потім декодуються для відновлення сигналів. Після цього система аналізує результати окреслення, порівнюючи оригінальні та реконструйовані сигнали для виявлення патологій. Отримані результати візуалізують для зручності

користувача, зберігаються для подальшого використання та генеруються звіти, які можуть бути використані для документування або подальшого аналізу.

Автокодуювальна нейромережа, що використовується в цій інформаційній системі, складається з двох основних компонентів: енкодер та декодер.

Енкодер відповідає за кодування вхідних даних, тобто сигналів ЕКГ, у вектор меншої розмірності. Він складається з декількох шарів нейронів, кожен з яких застосовує нелінійну функцію активації до своїх входів, що дає змогу ефективно зменшити розмірність даних, зберігаючи при цьому основну інформацію.

Декодер відповідає за декодування отриманого від енкодера вектор меншої розмірності назад у вихідні дані, тобто реконструйовані сигнали ЕКГ. Як і енкодер, декодер складається з декількох шарів нейронів, кожен з яких також застосовує нелінійну функцію активації до своїх входів, щоб відновити сигнал з максимальною точністю.

Вхідні дані, тобто сигнали ЕКГ, представлені у вигляді векторів, де кожен елемент вектора відповідає значенню сигналу в певний момент часу. Розмірність цих векторів варіюється залежно від частоти дискретизації та тривалості запису ЕКГ. Вихідні дані, тобто реконструйовані сигнали ЕКГ, також представлені у вигляді векторів, де кожен елемент вектора відповідає значенню сигналу в певний момент часу. Розмірність цих векторів відповідає розмірності вхідних даних, що дає змогу забезпечити точне відновлення сигналів після їх обробки нейромережею.

Система автокодування DenseNet буде використовуватись у різних сферах. У клінічній практиці вона стане незамінною для діагностики та моніторингу стану пацієнтів з кардіологічними захворюваннями, надаючи точні та оперативні дані, необхідні для прийняття медичних рішень. Наприклад, кардіологи зможуть використовувати DenseNet для швидкої оцінки стану серця пацієнта і виявлення потенційних проблем, що дозволить призначити ефективне лікування. У наукових дослідженнях DenseNet буде застосовуватись для вивчення характеристик сигналів ЕКГ, що сприятиме розробці нових методів

діагностики та лікування. Це дозволить дослідникам отримувати точніші дані і краще розуміти механізми серцевих захворювань. У телемедицині DenseNet стане важливим інструментом для дистанційного діагностування і моніторингу стану пацієнтів, що особливо важливо для людей, які живуть у віддалених районах або не мають доступу до спеціалізованих медичних закладів. Це дозволить лікарям надавати якісну медичну допомогу на відстані, використовуючи дані, зібрані системою. У фітнес-індустрії DenseNet буде використовуватися для аналізу ЕКГ спортсменів, допомагаючи тренерам і лікарям оцінювати серцево-судинну систему спортсменів і оптимізувати їхні тренувальні програми. Це дозволить спортсменам тренуватися більш безпечно та ефективно.

### **2.3 Проєктування структури програмної реалізації способу ідентифікації патологій серця за сигналом електрокардіограми**

Система автокодування DenseNet складається з кількох функціональних груп, кожна з яких виконує певні завдання для забезпечення ефективної роботи всієї системи.

На першому етапі відбувається підготовка даних. Користувач завантажує сигнали ЕКГ у систему, яка відповідає за їхнє форматування. Це передбачає перетворення сигналів у відповідний формат для подальшої обробки. Потім система проводить аналіз сигналів, яка включає видалення шуму та нормалізацію. Цей процес важливий для підвищення якості даних і забезпечення точності подальшого аналізу. Після цього система сегментує сигнали на фрагменти, що відповідають комплексам QRS. Сегментація дає змогу виділити важливі частини сигналу для більш детального аналізу.

Наступним етапом є навчання автокодера. Система за нашою вказівкою обирає параметри автокодера, такі як архітектура мережі, функції активації та оптимізатор. Ці параметри визначають структуру і поведінку автокодера під час навчання. Після вибору параметрів система за нашим готовим набором даних

навчає автокодер ЕКГ. Навчання передбачає оптимізацію моделі для точного відтворення вхідних сигналів на виході. Після завершення навчання навчена модель автокодера зберігається для подальшого використання.

На етапі окреслення сигналів ЕКГ користувач завантажує нові сигнали, які він хоче проаналізувати. Система знову виконує сегментацію сигналів на фрагменти, що відповідають комплексам QRS. Після цього система кодує ці фрагменти за допомогою навченого автокодера, тобто перетворює їх у компактні представлення. Потім ці компактні коди декодуються назад у реконструйовані сигнали ЕКГ. Це дає змогу відновити початкові сигнали з компактних представлень.

Система аналізує реконструйовані сигнали для виявлення патологій. Аналіз може включати порівняння оригінальних та реконструйованих сигналів для виявлення відхилень та аномалій, які можуть свідчити про наявність патологій. Нарешті, результати аналізу візуалізують для користувача, що дає змогу легко інтерпретувати отримані дані та прийняти відповідні рішення щодо подальшого медичного обстеження або лікування.

Набір даних РТВ Diagnostic ECG Database (РТВ-db) є одним з найбільших та найвідоміших наборів даних ЕКГ, що використовується для досліджень та розроблення методів аналізу ЕКГ. Цей набір містить записи ЕКГ від понад 200 пацієнтів з різними серцевими захворюваннями, а також від здорових людей. Записи оцифровані з частотою дискретизації 250 Гц і містять 12 відведень.

Використання набору даних РТВ-db в інформаційній системі передбачає навчання та тестування автокодувальної нейромережі. Для цього з набору вибираємо підмножину записів ЕКГ для навчання, а інші записи використовувати для тестування. Це дає змогу розробляти та вдосконалювати методи аналізу ЕКГ, забезпечуючи високу точність і надійність результатів.

Великий розмір та різноманітність набору даних РТВ-db надають можливість навчати автокодувальну нейромережу на широкому спектрі ЕКГ, що сприяє поліпшенню її ефективності та здатності розпізнавати різні патології. Крім того, РТВ-db добре анотований, що дає змогу використовувати його для

точного оцінювання роботи нейромережі. Оскільки цей набір даних широко використовується дослідниками, це полегшує порівняння результатів роботи настільного застосунку з результатами інших досліджень, сприяючи підвищенню рівня довіри до отриманих висновків.

Проте використання набору даних РТВ-db має деякі недоліки. Записи ЕКГ були зроблені в клінічних умовах, що може призвести до того, що нейромережа навчатиметься на сигналах, які мають шум або інші артефакти. Це може ускладнити процес навчання та потребує додаткової обробки даних для видалення небажаних перешкод. Крім того, набір даних РТВ-db не містить інформації про клінічний стан пацієнтів, що ускладнює інтерпретацію результатів аналізу ЕКГ, проведеного нейромережею. Відсутність цієї інформації може створювати труднощі при визначенні контексту і точного діагнозу для кожного конкретного випадку.

Інтерфейс програмного забезпечення для обробки електрокардіографічних сигналів повинен бути максимально зручним для користувача та простим у використанні. Інтерфейс має на меті надати користувачам доступ до всіх можливостей програми та забезпечити максимальну точність інтерпретації електрокардіографічних сигналів.

Інтерфейс програми для обробки електрокардіографічних сигналів має включати декілька ключових елементів для зручного та ефективного використання:

- головне меню – це основне керівне меню програми, яке надає доступ до основних функцій, таких як відкриття та збереження файлів, налаштування програми та інші;

- панель інструментів містить інструменти для обробки та аналізу даних електрокардіографів, що спрощує доступ до потрібних операцій;

- вікно відображення даних показує електрокардіограму сигналу, дає змогу масштабувати та переглядати різні ділянки сигналу в згорнутому, розгорнутому або інтервальному режимі;

– вікно додаткової інформації відображає результати обробки даних електрокардіограми після аналізу, що допомагає користувачеві зрозуміти та інтерпретувати отримані результати.

Під час запуску програми користувач має можливість вибрати файл з електрокардіограмою для обробки. Сигнал ЕКГ відображається на графічному інтерфейсі для зручного перегляду та аналізу.

Програма має функції для обробки електрокардіографічних сигналів, такі як фільтрація, аналіз аритмій, визначення параметрів, які будуть використовуватись за допомогою кнопок інтерфейсу.

Крім того, програма дає змогу користувачеві зберігати оброблені електрокардіограми сигнал та результати аналізу у файлі або надрукувати їх. Усі взаємодії з програмним забезпеченням пов'язані з обробкою даних ЕКГ і мають бути простими та логічними для користувача.

Створення програми для обробки електрокардіографічних сигналів потребує ретельного Проектування інтерфейсу та використання сучасних технологій для створення зручного та інтерактивного інтерфейсу. Це дозволить користувачеві ефективно працювати з програмою та отримувати необхідні результати аналізу.

Програмне забезпечення, яке реалізує обробку ЕКГ, повинно мати досить широкий функціонал, на рисунку 2.4 зображено основні функції програми.

Ця інформаційна система для аналізу сигналів ЕКГ розроблена для глибокого і комплексного оброблення медичних даних, що надходять з різних джерел. Вона починає свою роботу з першої функціональної частини – імпорту ЕКГ сигналів. Цей модуль відповідає за завантаження і введення даних з різних джерел, таких як цифрові записи, архівні дані чи прямі вимірювання зі спеціалізованих медичних пристроїв. Важливою складовою цього процесу є конвертація і стандартизація даних до формату, що використовується системою для подальшої обробки.

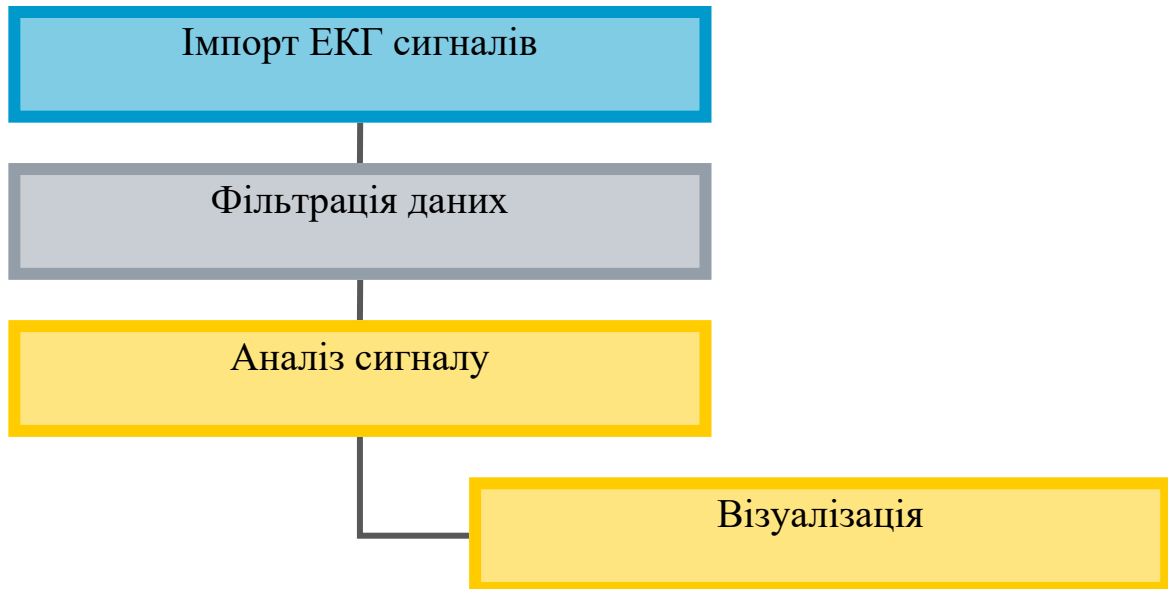


Рисунок 2.4 – Основні функції програми

Після завантаження даних настає фаза фільтрації. Ця функція використовується для очищення сигналів ЕКГ від шумів та артефактів, що можуть виникнути в процесі їх збору чи передачі. Застосування цифрових фільтрів дає змогу виділити корисні сигнали і підготувати їх для подальшого аналізу, зберігаючи при цьому високу точність і достовірність даних.

Після фільтрації даних вступає в дію модуль аналізу сигналу. Цей етап включає в себе автоматизоване або напівавтоматичне виділення ключових компонентів сигналу ЕКГ, таких як комплекси QRS, які відображають скорочення шлуночків серця, і визначення інших параметрів, що характеризують функцію серця. Він включає аналіз інтервалів PQ, QRS, QT і інших параметрів, які використовуються для оцінки структурних і функціональних аспектів серцевої діяльності. Крім того, модуль аналізу виявляє різні аномалії в ритмі серця, такі як аритмії, екстрасистолії та інші відхилення, що можуть вказувати на серйозні медичні стани.

Останнім етапом є візуалізація результатів. Ця функція надає користувачам можливість переглядати оброблені дані у вигляді інтерактивних графіків ЕКГ і виводити ключові параметри, що були визначені в процесі аналізу. Вона дає змогу лікарям та іншим медичним фахівцям оцінювати стан

серця пацієнтів з точністю і зрозумілістю, що є критично важливим для прийняття інформованих медичних рішень.

Ця інтегрована платформа забезпечує високу ефективність у обробці та аналізі ЕКГ-даних, підвищуючи точність діагностики серцевих захворювань і сприяючи поліпшенню підходів до лікування пацієнтів.

## **2.4 Проектування бази даних програмної реалізації**

У цьому розділі буде розглянуто аспекти Проектування програмного продукту для автоматизованої системи ідентифікації патологій серця за ЕКГ сигналами з використанням автокодувальної нейромережі.

Схема роботи програми включають узагальнену схему програмного комплексу, яка покаже загальну структуру системи, її основні компоненти та їх взаємозв'язки. Буде представлена схема користувачів та груп користувачів системи, яка опише типи користувачів, які матимуть доступ до системи, та їхні ролі. Окрім цього, буде детально описано функціонал, доступний кожній групі користувачів. Схема архітектури системи покаже компоненти системи, їх зв'язки та потоки даних.

Опис компонентів системи розпочнеться з бази даних (БД), яка міститиме всі дані, необхідні для роботи системи. Це включає ЕКГ сигнали пацієнтів, результати аналізу ЕКГ сигналів автокодувальною нейромережею, інформацію про пацієнтів, налаштування системи. Склад БД буде представлений у вигляді переліку таблиць/сутностей з описом їх полів та зв'язків між ними.

Інтерфейси користувачів включають графічний інтерфейс користувача для лікарів, який дозволить завантажувати ЕКГ сигнали пацієнтів, запускати аналіз нейромережею та переглядати результати, а також переглядати інформацію про пацієнтів та їх медичну історію. На рисунку 2.5 показана схема настільного застосунку.

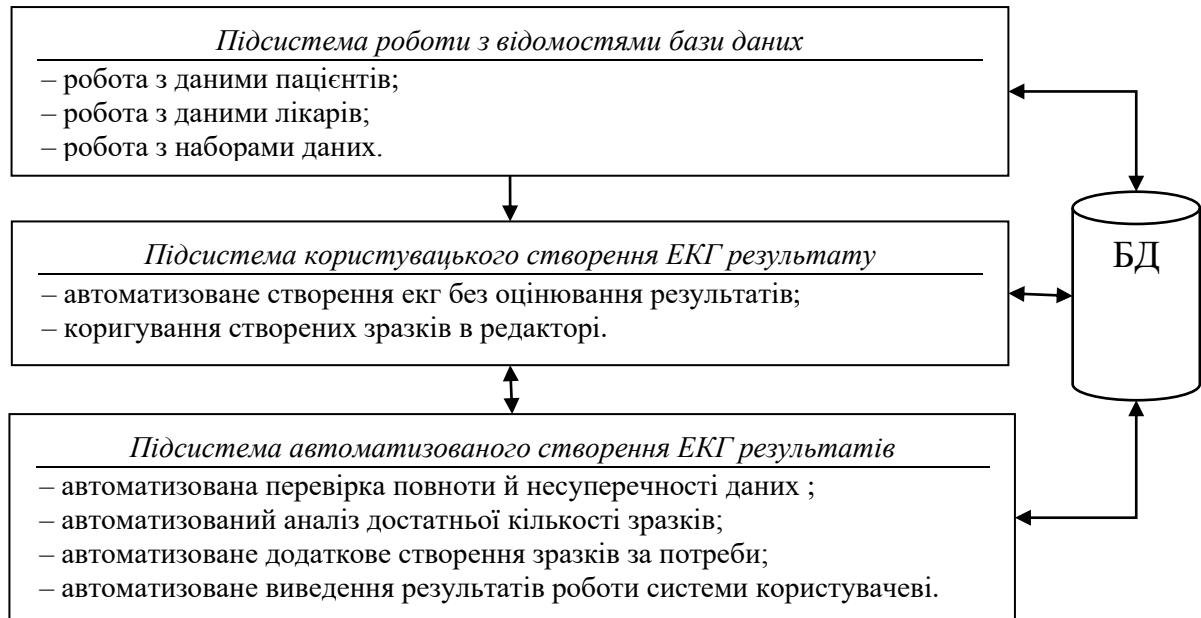


Рисунок 2.5 – Схема БД настільного застосунку

Підсистеми включають підсистему завантаження ЕКГ сигналів, яка відповідає за завантаження сигналів з різних джерел, перевіряє їх цілісність та формат. Підсистема передпроцесингу ЕКГ сигналів виконує фільтрацію шуму, нормалізацію даних, сегментацію сигналів. Підсистема аналізу ЕКГ сигналів автокодувальною нейромережею запускає аналіз, інтерпретує результати та формує висновок про стан серця пацієнта. Підсистема візуалізації результатів візуалізує ЕКГ сигнали та результати аналізу, дає змогу лікарям порівнювати їх з нормами. Підсистема звітності формує звіти про результати аналізу, дає змогу експортувати їх в різних форматах.

Зв'язки між компонентами системи описують взаємодію користувачів з системою через інтерфейси користувачів, які передають дані та команди підсистемам. Підсистеми обмінюються даними між собою, а БД зберігає всі дані, які використовуються системою. Підсистема аналізу використовує модель нейронної мережі, яка зберігається на диску.

Проектування БД для автоматизованої системи ідентифікації патологій серця за ЕКГ сигналами включає кілька основних підсистем, які зберігають дані пацієнтів, лікарів, результати ЕКГ аналізу, метадані та інші допоміжні дані. Вони містять інформацію про пацієнтів, зокрема унікальний ідентифікатор, ім'я,

прізвище, дату народження, стать, контактну інформацію та медичну історію. Також включають дані про лікарів, такі як унікальний ідентифікатор, ім'я, прізвище, спеціалізація, контактна інформація та ідентифікатор користувача та зберігає результати ЕКГ аналізу, включаючи унікальний ідентифікатор результату, ідентифікатори пацієнта та лікаря, дані ЕКГ, дату та час проведення аналізу, результат аналізу та коментарі лікаря. Також містить інформацію про набори даних, такі як унікальний ідентифікатор, назва, опис та дата створення, включає дані про користувачів системи, зокрема унікальний ідентифікатор, ім'я користувача.

Підсистема автоматизованого створення ЕКГ результатів включає автоматизовану перевірку повноти й несуперечності даних, аналіз достатньої кількості зразків, додаткове створення зразків за потреби та виведення результатів роботи системи користувачеві. В ній зберігаються дані про автоматизовані аналізи, зокрема унікальний ідентифікатор аналізу, ідентифікатор результату ЕКГ, статус аналізу, повідомлення про помилку (якщо є) та дату і час проведення аналізу.

Підсистема роботи з відомостями БД включає роботу з даними пацієнтів, лікарів та датасетами. Підсистема користувацького створення ЕКГ результату забезпечує автоматизоване створення ЕКГ без оцінювання результатів та коригування створених зразків у редакторі, вона містить дані про створені користувачем результати, такі як унікальний ідентифікатор створеного результату, ідентифікатор користувача, дані ЕКГ, дату та час створення результату та коментарі користувача.

Ці таблиці забезпечують зберігання всіх необхідних даних для функціонування автоматизованої системи ідентифікації патологій серця за ЕКГ сигналами, включаючи збереження інформації про пацієнтів, лікарів, результати аналізів та інші супутні дані.

## 2.5 Висновки до розділу 2

У цьому розділі наведено опис спроектованого способу ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі. Цей спосіб складається з чотирьох основних етапів: збір даних, обробка даних, навчання нейромережі та візуалізація. Описано в деталях архітектуру автокодувальної нейромережі DenseNet, яка використовується в межах способу.

На основі спроектованого було спроектовано архітектуру настільного застосунку зі зручним інтерфейсом користувача, який включає головне меню, панель інструментів, вікно відображення даних та панель інтерфейсу. Користувач може завантажувати сигнали ЕКГ, обробляти їх за допомогою різних функцій та зберігати результати.

Використання набору даних PTB Diagnostic ECG Database (PTB-db) дало змогу навчати та тестувати автокодувальну нейромережу, що сприяє поліпшенню її точності та здатності розпізнавати різні патології. Набір даних PTB-db є одним з найбільш повних і надійних джерел для навчання моделей машинного навчання, що дає змогу забезпечити високу точність і надійність виявлення патологій серця.

Отже, спроектований спосіб ідентифікації патологій серця надає переваги для медичної діагностики, дозволяючи лікарям швидко і точно виявляти патології серця, знижуючи водночас навантаження на медичний персонал і покращуючи результати лікування пацієнтів.

## Розділ 3 Програмна реалізація способу ідентифікації патологій серця за сигналом електрокардіограми

### 3.1 Структура та функціональне призначення складових програмної реалізації способу ідентифікації патологій серця

У цьому розділі наведено програмну реалізацію способу ідентифікації патологій серця за сигналом електрокардіограми. Схематично програмна реалізація подана у вигляді схеми класів на рисунку 3.1.

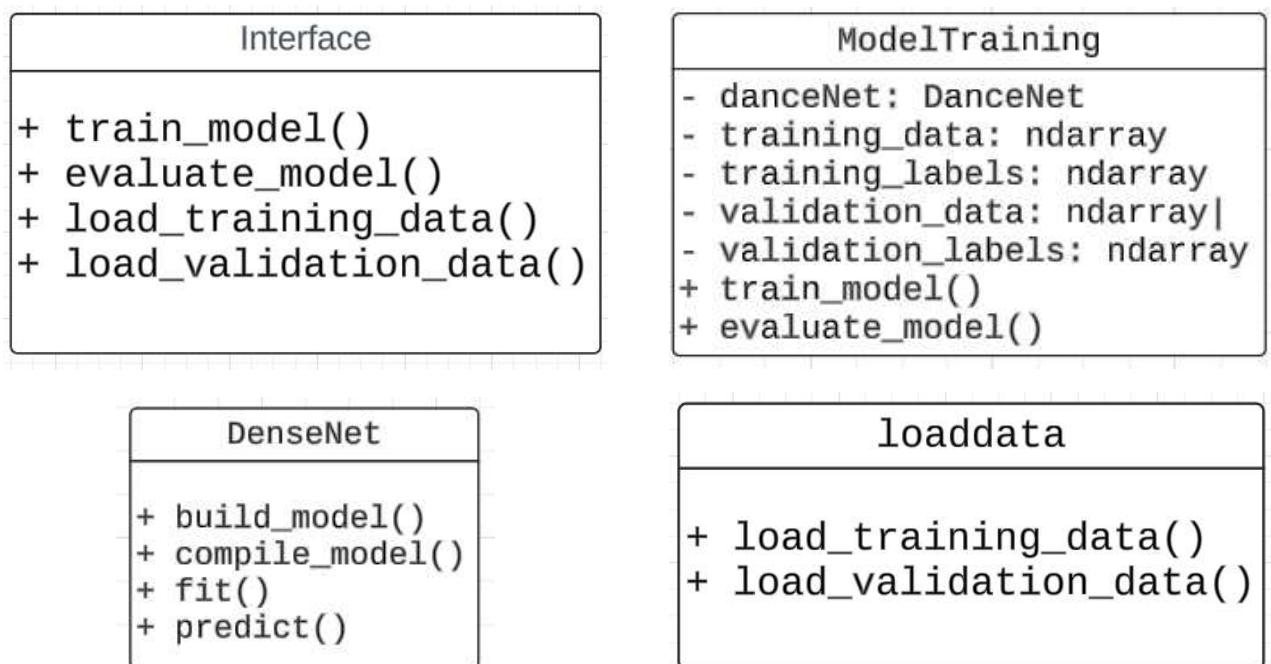


Рисунок 3.1 – Схема класів створеного настільного застосунку

Схема на рисунку 3.1 відображає структуру та взаємозв'язки компонентів системи для ідентифікації патологій серця за сигналами електрокардіограми. Основною метою даної системи є забезпечення автоматичного аналізу ЕКГ сигналів з використанням сучасних методів машинного навчання, зокрема автокодувальної нейромережі DenseNet.

Клас Interface визначає базові методи, які повинні бути реалізовані іншими класами. Це свого роду шаблон, який гарантує, що всі класи, які його наслідують, будуть мати однаковий набір методів (таблиця 3.1).

Таблиця 3.1 – Функції класу «Interface»

Метод	Опис
train_model()	Запускає процес навчання моделі.
evaluate_model()	Оцінює точність моделі на валідаційних даних.
load_training_data()	Завантажує навчальні дані.
load_validation_data()	Завантажує валідаційні дані.

Клас ModelTraining реалізує інтерфейс Interface і відповідає за процес навчання та оцінки моделі. Він взаємодіє з класом DenseNet для роботи з нейромережею та класом loaddata для завантаження даних (таблиця 3.2).

Таблиця 3.2 – Функції класу «ModelTraining»

Метод	Опис
init()	Ініціалізує об'єкт ModelTraining, створюючи об'єкт DenseNet і встановлюючи початкові значення для атрибутів даних.
train_model()	Викликає методи класу DenseNet для компіляції та тренування моделі на навчальних даних.
evaluate_model()	Викликає методи класу DenseNet для оцінки моделі на валідаційних даних.
load_training_data()	Викликає методи класу loaddata для завантаження навчальних даних і міток.
load_validation_data()	Викликає методи класу loaddata для завантаження валідаційних даних і міток.

Клас DenseNet відповідає за побудову, компіляцію, навчання та передбачення моделі нейронної мережі. Він використовує бібліотеку Keras для створення автокодувальної нейромережі (таблиця 3.3).

Таблиця 3.3 – Функції класу «DenseNet»

Метод	Опис
init()	Ініціалізує об'єкт DenseNet, встановлюючи початкове значення для атрибута model.
build_model()	Будує архітектуру моделі нейронної мережі, включаючи шари для кодування та декодування.
compile_model()	Компілює модель з оптимізатором та функцією втрат.
fit()	Навчає модель на навчальних даних з використанням вказаних параметрів, таких як кількість епох, розмір пакету і т.д.
evaluate()	Оцінює продуктивність моделі на валідаційних даних.
predict()	Виконує передбачення на нових даних, використовуючи навчену модель.

Клас `loaddata` містить методи для завантаження даних, необхідних для навчання та валідації моделі. Ці методи зчитують дані з файлів та повертають їх у вигляді `numpy` масивів (таблиця 3.4).

Таблиця 3.4 – Функції класу «LoadData»

Метод	Опис
load_training_data()	Завантажує навчальні дані та мітки з файлів і повертає їх як <code>numpy</code> масиви.
load_validation_data()	Завантажує валідаційні дані та мітки з файлів і повертає їх як <code>numpy</code> масиви.

Якщо підсумовувати ці класи, вони спільно формують систему для ідентифікації патологій серця за сигналами електрокардіограми з використанням автокодувальної нейромережі DenseNet. Клас `ModelTraining` керує загальним процесом навчання та оцінки моделі, використовуючи методи з класів DenseNet і

loaddata. Клас DenseNet реалізує архітектуру нейронної мережі та методи для її тренування та оцінки, а клас loaddata забезпечує завантаження необхідних даних для цього процесу. Інтерфейс Interface гарантує, що всі необхідні методи реалізовані у відповідних класах.

Загалом структура системи забезпечує відповідні функціональні можливості для завантаження, обробки та класифікації електрокардіографічних сигналів. Використання таблиць з функціями та діаграм класів полегшує розуміння структури системи та взаємозв'язків між її компонентами.

Ці класи спільно формують систему для ідентифікації патологій серця за сигналами електрокардіограми з використанням автокодувальної нейромережі DenseNet. Клас ModelTraining керує загальним процесом навчання та оцінки моделі, використовуючи методи з класів DenseNet і loaddata. Клас DenseNet реалізує архітектуру нейронної мережі та методи для її тренування та оцінки, а клас loaddata забезпечує завантаження необхідних даних для цього процесу. Інтерфейс Interface гарантує, що всі необхідні методи реалізовані у відповідних класах.

Методи класу Interface включають: `train_model`, який запускає процес навчання моделі; `evaluate_model`, що оцінює точність моделі на валідаційних даних; `load_training_data`, який завантажує навчальні дані; та `load_validation_data`, що завантажує валідаційні дані. Клас ModelTraining реалізує інтерфейс Interface і відповідає за процес навчання та оцінки моделі. Він взаємодіє з класом DenseNet для роботи з нейромережею та класом LoadData для завантаження даних.

Методи класу ModelTraining включають: `init`, який ініціалізує об'єкт ModelTraining, створюючи об'єкт DenseNet і встановлюючи початкові значення для атрибутів даних; `train_model`, який викликає методи класу DenseNet для компіляції та тренування моделі на навчальних даних; `evaluate_model`, що викликає методи класу DenseNet для оцінки моделі на валідаційних даних; `load_training_data`, який викликає методи класу LoadData для завантаження навчальних даних і міток; та `load_validation_data`, що викликає методи класу LoadData для завантаження валідаційних даних і міток. Клас DenseNet відповідає

за побудову, компіляцію, навчання та передбачення моделі нейронної мережі. Він використовує бібліотеку Keras для створення автокодувальної нейромережі.

Методи класу DenseNet включають: `init`, який ініціалізує об'єкт DenseNet, встановлюючи початкове значення для атрибута `model`; `build_model`, що будує архітектуру моделі нейронної мережі, включаючи шари для кодування та декодування; `compile_model`, який компілює модель з оптимізатором та функцією втрат; `fit`, що навчає модель на навчальних даних з використанням вказаних параметрів, таких як кількість епох, розмір пакету і т.д.; `evaluate`, який оцінює продуктивність моделі на валідаційних даних; та `predict`, що виконує передбачення на нових даних, використовуючи навчену модель. Клас `LoadData` містить методи для завантаження даних, необхідних для навчання та валідації моделі. Ці методи зчитують дані з файлів та повертають їх у вигляді numpy масивів.

Методи класу `LoadData` включають: `load_training_data`, який завантажує навчальні дані та мітки з файлів і повертає їх як numpy масиви; та `load_validation_data`, що завантажує валідаційні дані та мітки з файлів і повертає їх як numpy масиви.

Загалом, ці класи спільно формують систему для ідентифікації патологій серця за сигналами електрокардіограми з використанням автокодувальної нейромережі DenseNet. Клас `ModelTraining` керує загальним процесом навчання та оцінки моделі, використовуючи методи з класів DenseNet і LoadData. Клас DenseNet реалізує архітектуру нейронної мережі та методи для її тренування та оцінки, а клас LoadData забезпечує завантаження необхідних даних для цього процесу. Інтерфейс `Interface` гарантує, що всі необхідні методи реалізовані у відповідних класах.

Взаємодія між класами відбувається таким чином, що `ModelTraining` ініціює процес навчання та оцінки, використовуючи методи DenseNet для роботи з нейромережею та LoadData для завантаження даних. DenseNet будує, компілює та тренує модель, а також оцінює її продуктивність. LoadData забезпечує надання необхідних даних для цього процесу. Інтерфейс `Interface` гарантує

стандартизацію методів, що реалізуються у відповідних класах. Це забезпечує узгодженість і взаємозв'язок між різними компонентами системи.

### **3.2 Вибір засобів розроблення для програмної реалізації способу ідентифікації у вигляді настільного застосунку**

Для розроблення настільного застосунку були обрані наступні програмні засоби: мова програмування Python, інтегроване середовище розроблення Jupyter Notebook та платформа розроблення Anaconda Navigator [29].

Python [30] є універсальною мовою програмування, яка широко використовується в галузі штучного інтелекту та машинного навчання завдяки своєму чіткому синтаксису, великій бібліотеці стандартних модулів та активній спільноті розробників.

Jupyter Notebook, як інтегроване середовище розроблення, є веб-заснованим інструментом, який поєднує код Python, текст, візуалізації та інші елементи в одному документі. Це робить його зручним для прототипування та розроблення машинних навчальних моделей.

Anaconda Navigator є графічним інтерфейсом користувача для Anaconda Distribution, яка є популярною платформою для наукових обчислень на Python. Ця платформа полегшує встановлення та керування пакетами Python, а також запуск програми Jupyter Notebook.

Вибір цих програмних засобів був обумовлений їхніми потужними можливостями та універсальністю:

- Python є ідеальною мовою для розроблення в галузі штучного інтелекту та машинного навчання завдяки своєму чіткому синтаксису, великій бібліотеці модулів та активній спільноті розробників;

- Jupyter Notebook, зі своєю здатністю поєднувати код, текст та візуалізації в одному документі, є ідеальним інструментом для прототипування та розроблення машинних навчальних моделей;

– Anaconda Navigator, як графічний інтерфейс для Anaconda Distribution, значно полегшує встановлення та керування пакетами Python, а також забезпечує зручний запуск Jupyter Notebook.

Використання цих програмних засобів дозволило створити інформаційну систему, яка є зручною у використанні, ефективною та масштабованою.

### **3.3 Особливості реалізації програмних складових настільного застосунку**

Реалізація настільного застосунку для ідентифікації серцевих патологій за допомогою сигналів електрокардіограм із застосуванням автокодувальної нейромережі включає кілька основних етапів, які разом сприяють досягненню цілей системи. Першим етапом є розробка користувацького інтерфейсу, який забезпечує зручну та інтуїтивно зрозумілу взаємодію з системою. Інтерфейс включає елементи для введення даних, кнопки та панелі для навігації і відображення результатів аналізу ЕКГ.

Наступним кроком є створення модуля завантаження та обробки даних, що відповідає за отримання вхідних даних і їх підготовку для подальшого аналізу. Цей модуль містить функції для завантаження сигналів ЕКГ, їх попередньої обробки, такої як фільтрація шумів, нормалізація та сегментація сигналів на окремі комплекси QRS.

Потім відбувається етап навчання моделі, де дані використовуються для тренування автокодувальної нейромережі, спеціалізованої на виявленні серцевих патологій. Під час цього етапу здійснюється оптимізація гіперпараметрів, навчання та оцінка продуктивності моделі на тренувальних і тестових наборах даних.

Останнім етапом є інтеграція навченої моделі в систему для її практичного застосування. Цей процес включає створення модулів для кодування нових сигналів ЕКГ, їх декодування у реконструйовані версії та аналіз результатів для виявлення можливих патологій. Результати аналізу візуалізують

та зберігаються для подальшого використання, що дає змогу користувачам зберігати або друкувати результати аналізу.

Модуль Interface відповідає за взаємодію з користувачем. Створення зручного та легко зрозумілого інтерфейсу є важливою частиною реалізації цього модуля, за допомогою Anaconda було реалізовано інтерфейс додатку, в якому користувач може взаємодіяти з даними ЕКГ. На рисунку 3.2 зображено інтерфейс застосунку.

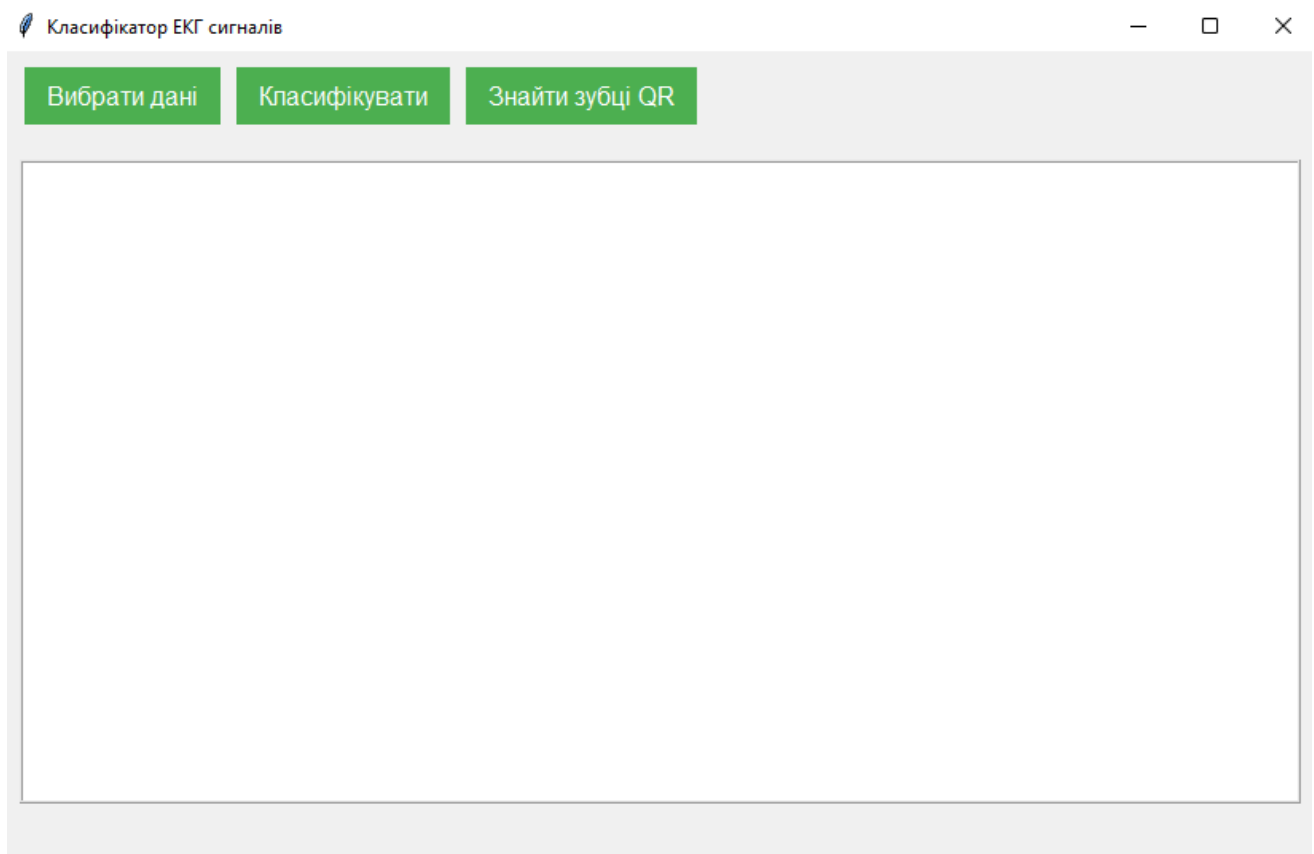


Рисунок 3.2 – Інтерфейс настільного застосунку

Модуль LoadData відповідає за завантаження та попередню обробку даних. Метод `load_validation_data` модуля LoadData виконує не лише нормалізацію сигналів ЕКГ, та аугментацію, що допомагає покращити роботу моделі DenseNet. Нормалізація включає масштабування та перетворення сигналів у формат, зрозумілий для моделі. Аугментація даних робить тренувальний набір різноманітнішим за рахунок застосування різних трансформацій, таких як зменшення шуму, зміна амплітуди та зміщення фаз. Це

покращує якість навчання та здатність моделі розпізнавати патології на сигналах ЕКГ з різними характеристиками. На рисунку 3.3 зображено роботу даного модуля.

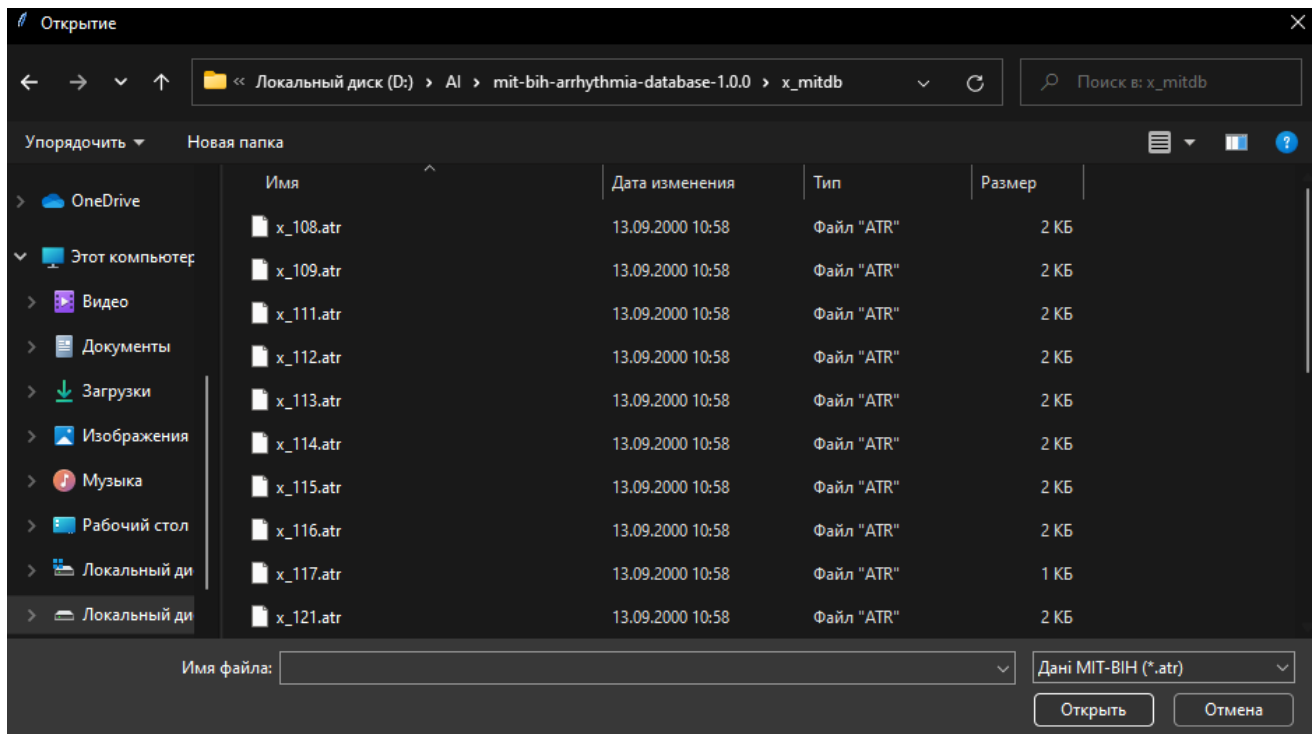


Рисунок 3.3 – Процес завантаження даних в розробленому застосунку

Модуль DenseNet відповідає за побудову, компіляцію, навчання та передбачення моделі нейронної мережі. Цей модуль забезпечує структуру та функціонал нейронної мережі, що використовується для ідентифікації патологій серця за сигналами електрокардіограм.

На рисунку 3.4 зображено роботу модуля DenseNet. Процес роботи модуля включає кілька ключових етапів.

1. Побудова моделі. Модуль DenseNet включає в себе логіку для побудови архітектури нейронної мережі. Це може бути здійснено за допомогою різних типів шарів, таких як звичайні пов'язані шари, згорткові шари, шари пулінгу тощо, які дозволяють моделі вивчати кореляції між вхідними даними.

2. Компіляція моделі. Після побудови архітектури моделі необхідно її скомпілювати, щоб задати параметри оптимізації, функції втрат та метрики для оцінки її продуктивності під час навчання.



Рисунок 3.4 – Робота модуля DenseNet

3. Навчання моделі. Після компіляції модель може бути навчена на навчальних даних. Цей етап включає передачу навчальних даних через мережу, визначення градієнтів та корекцію параметрів моделі для мінімізації значень функції втрат.

4. Передбачення. Після навчання модель може бути використана для передбачення класів або значень для нових вхідних даних. Цей етап дає змогу використовувати модель для ідентифікації патологій серця за сигналами електрокардіограм.

Модуль DenseNet грає ключову роль у процесі розроблення та використання нейронних мереж для розв'язання задачі ідентифікації патологій серця за сигналами ЕКГ.

### 3.4 Тестування програмної реалізації та основні вимоги до розгортання розробленого настільного застосунку

Тестування настільного застосунку є важливим етапом розроблення програмного забезпечення, яке дає змогу перевірити його функціональність, ефективність і надійність перед впровадженням. З метою виявлення потенційних проблем і помилок цей процес передбачає проведення спеціальних експериментів, у яких систему піддають різним сценаріям і навантаженням.

У процесі експериментального тестування інформаційних систем тест-кейси є важливою частиною процесу. Якщо дані не були обрані до виконання певних операцій, наприклад, натискання кнопок «Вибрати дані», «Класифікувати» або «Знайти пікові точки» без вибраних даних, буде проведено тестування на помилки. Вікно, що повідомляє про помилку, зображено на рисунку 3.5.

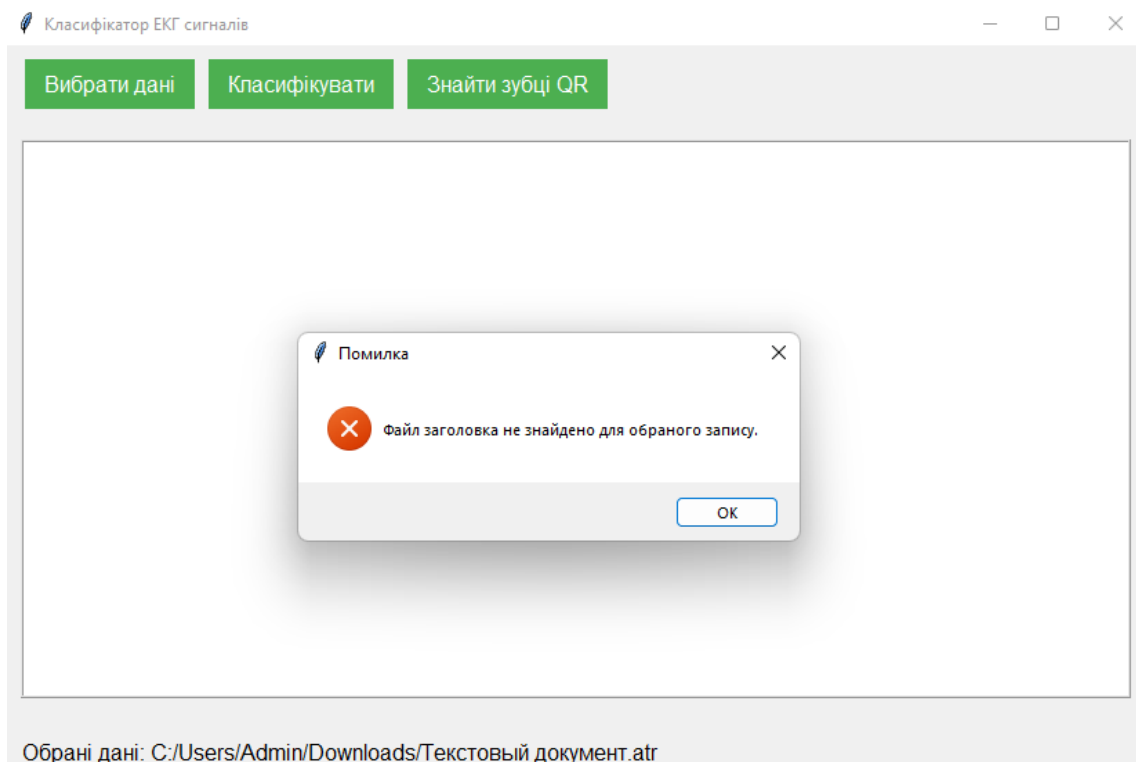


Рисунок 3.5 – Демонстрація сповіщення помилки

Опис тест-кейсу TC0001 подано в таблиці 3.5.

Таблиця 3.5 – Тест-кейс TC0001

<b>Тест-кейс ID:</b> TC0001	<b>Пріоритет:</b> 2	<b>Створено:</b> 21.05.24, Гамбарян А.А
<b>Назва:</b> Тестування на помилку, якщо дані не були обрані до виконання певних операцій, наприклад, натискання кнопок “Вибрати дані” <b>Натискання на кнопку «Вибрати дані» та вибір файла</b>		
<b>Кроки</b>		<b>Очікуваний результат</b>
Передумова: користувач вибрав дані, які не містять необхідних числових або символічних елементів. 1. Запустити програму. 2. Натиснути кнопку “Вибрати дані”. 3. Вибрати файл формату .atr.		З’являється вікно, яке сповіщає користувача про те, що «не вибрано жодних даних».
<b>Результат виконання тест-кейсу:</b> перевірку пройдено успішно.		

Опис тест-кейсу TC0002 подано в таблиці 3.6.

Таблиця 3.6 – Тест-кейс TC0002

<b>Тест-кейс ID:</b> TC0002	<b>Пріоритет:</b> 2	<b>Створено:</b> 21.05.24, Гамбарян А.А
<b>Назва:</b> Тестування на помилку в ситуаціях, коли дані не були обрані для виконання певних операцій (наприклад, натискання кнопки «Класифікувати»). <b>Натискання на кнопку «Класифікувати»</b>		
<b>Кроки</b>		<b>Очікуваний результат</b>
Передумова: користувач вибрав дані, які не містять необхідних числових або символічних елементів. 1. Запустити програму. 2. Натиснути кнопку «Класифікувати».		З’являється вікно, яке сповіщає користувача про те, що «не вибрано жодних даних».
<b>Результат виконання тест-кейсу:</b> перевірку пройдено успішно.		

Коли користувач натискає кнопку «Класифікувати» без завантаження даних, програма повідомить йому, що файл з відповідними даними не встановлений. В результаті програма буде сповіщати про помилку.

Демонстрація сповіщення помилки зображено на рисунку 3.6.

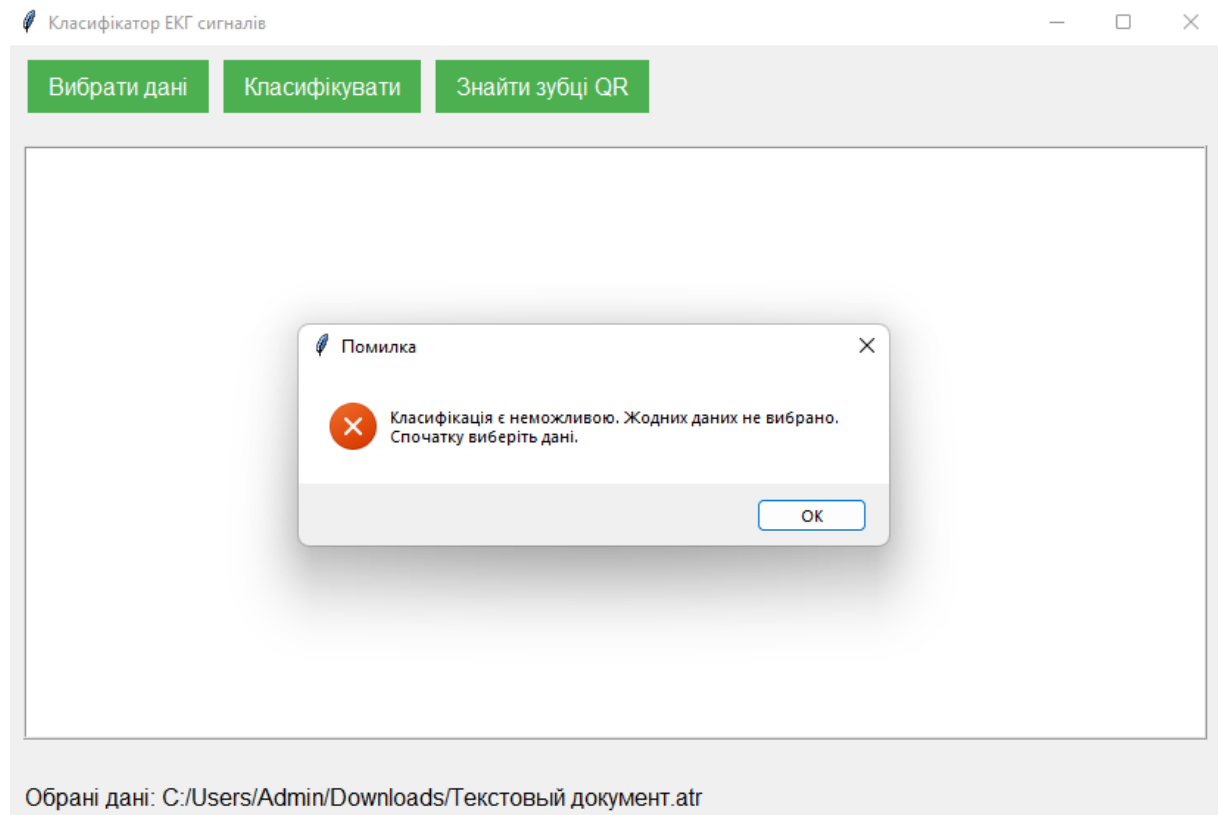


Рисунок 3.6 – Сповіщення про помилку класифікації

Опис тест-кейсу TC0003 в таблиці 3.7.

Таблиця 3.7 – Тест-кейс TC0003

<b>Тест-кейс ID:</b> TC0003	<b>Приоритет:</b> 2	<b>Створено:</b> 21.05.24, Гамбарян А.А
<b>Назва:</b> Тестування на помилку, якщо дані не були обрані до виконання певних операцій (наприклад, натискання кнопки «Знайти пікові точки»).		
<b>Натискання на кнопку</b> «Знайти пікові точки».		
<b>Кроки</b>		<b>Очікуваний результат</b>
Передумова: користувач вибрав дані, які не містять необхідних числових або		З'являється вікно, яке сповіщає користувача про те, що «не вибрано

<p>символьних елементів.</p> <ol style="list-style-type: none"> <li>1. Запустити програму.</li> <li>2. Натиснути кнопку «Знайти пікові зубці QR».</li> </ol>	<p>жодних даних».</p>
<p><b>Результат виконання тест-кейсу:</b> перевірку пройдено успішно.</p>	

Програма повідомить користувача, що файл даних не був завантажений після натискання кнопки «Знайти пікові точки».

Демонстрація сповіщення також зображено на рисунку 3.7.

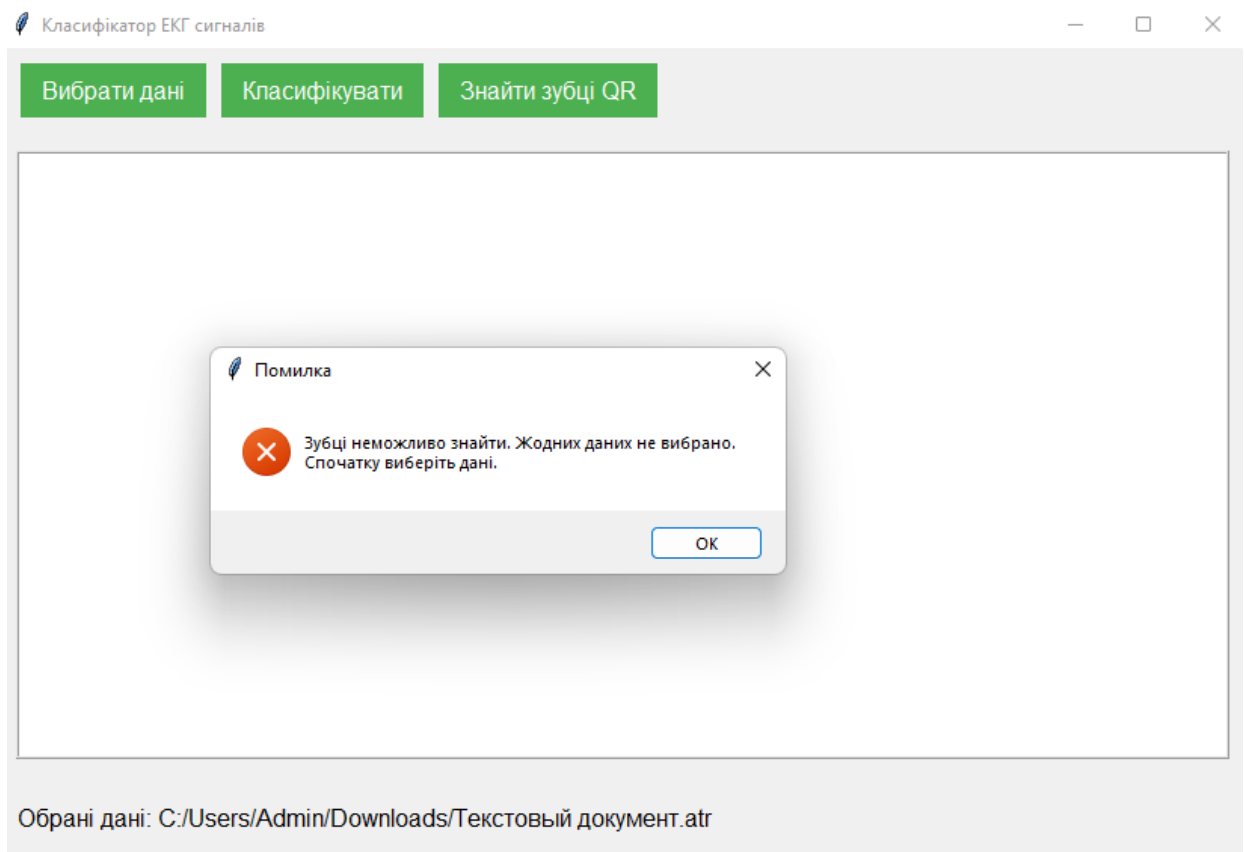


Рисунок 3.7 – Демонстрація сповіщення помилки у пошуках зубців

Опис тест-кейсу TC0004 в таблиці 3.8.

Отже, ЕКГ-сигнал відобразиться на графічному інтерфейсі після завершення завантаження файлу з даними. Коли файл завантажувється, інтерфейс показує шлях до директорії.

Таблиця 3.8 – Тест-кейс TC0004

<b>Тест-кейс ID:</b> TC0004	<b>Пріоритет:</b> 1	<b>Створено:</b> 21.05.24, Гамбарян А.А
<b>Назва:</b> Тестування на успішне завантаження даних.		
<b>Вхідні дані:</b> Натискання на кнопку «Вибрати дані»		
<b>Кроки</b>		<b>Очікуваний результат</b>
Передумова: Завантаження даних. 1. Запустити програму. 2. Натиснути кнопку «Вибрати дані». 3. Вибрати файл формату .atr з директорії MIT-BIH Arrhythmia Database. 4. Завантажити.		Після завершення процесу обробки файлу програмою на графічному інтерфейсі з'явиться вибраний сигнал ЕКГ.
<b>Результат виконання тест-кейсу:</b> перевірку пройдено успішно.		

Результат тестування зображено на рисунку 3.8.



Рисунок 3.8 – Результат успішної роботи функції «Вибрати дані»

## Опис тест-кейсу TC0005 в таблиці 3.9.

Таблиця 3.9 – Тест-кейс TC0005

<b>Тест-кейс ID:</b> TC0005	<b>Приоритет:</b> 1	<b>Створено:</b> 21.05.24, Гамбарян А.А
<b>Назва:</b> Тестування на успішне окреслення ЕКГ-сигналу Вхідні дані: <b>Натискання на кнопки</b> «Вибрати дані», «Класифікація» та «Знайти зубці QR»		
<b>Кроки</b>		<b>Очікуваний результат</b>
Передумова: Опис ЕКГ-сигналу. 1. Запустити програму. 2. Натиснути кнопку «Вибрати дані». 3. Вибрати файл формату .atr з директорії MIT-BIH Arrhythmia Database. 4. Завантажити. 5. Натиснути кнопку «Класифікація». 6. Натиснути кнопку «Знайти зубці QR».		Клас електрокардіограмного сигналу, а також зубці Q та R будуть відобразитися на графічному інтерфейсі. Назва запису, кількість каналів, частота дискретизації та види каналів включають додаткову інформацію.
<b>Результат виконання тест-кейсу:</b> перевірку пройдено успішно.		

На графічному інтерфейсі настільного застосунку відображається необхідна інформація про електрокардіографічний сигнал. Результат окреслення та класифікації сигналу зображено на рисунку 3.9.

Тест-кейс для перевірки сповіщення про помилку при невдачі завантаження даних, успішного завантаження та опису електрокардіографічного сигналу був використаний під час експериментального тестування настільного застосунку. У цьому тест-кейсі містилася послідовність дій, спрямованих на вирішення ситуації, коли завантаження даних не вдалося.

Під час тестування системи за допомогою цього тест-кейсу було виявлено, що система правильно сповіщає про помилку. Таке повідомлення дає змогу користувачам швидко реагувати на проблему і вживати необхідних заходів для її вирішення.



Рисунок 3.9 – Опис ЕКГ-сигналу

Перевірка сповіщення про помилку за допомогою тест-кейсу дає змогу переконатися, що система ефективно реагує на такі ситуації, надаючи користувачам важливу інформацію для подальших дій. Тест-кейси, що перевіряють роботу функцій настільного застосунку, будуть використовуватися для перевірки успішності виконання завдань.

### 3.5 Функціональний опис створеного настільного застосунку

Функціональне призначення містить опис використання настільного застосунку, що використовує автокодувальну нейромережу для ідентифікації патологій серця за допомогою ЕКГ сигналів. Нейромережа, заснована на архітектурі DenseNet, використовується для автоматичного аналізу і класифікації цих сигналів. Програмне забезпечення складається з наступних компонентів:

Інтерфейс користувача – графічний інтерфейс користувача (GUI) для завантаження ЕКГ сигналу, запуску класифікації та візуалізації результатів; модель машинного навчання – автокодувальна нейромережа, навчена на великому наборі даних ЕКГ сигналів з відомими патологіями (рисунок 3.10).

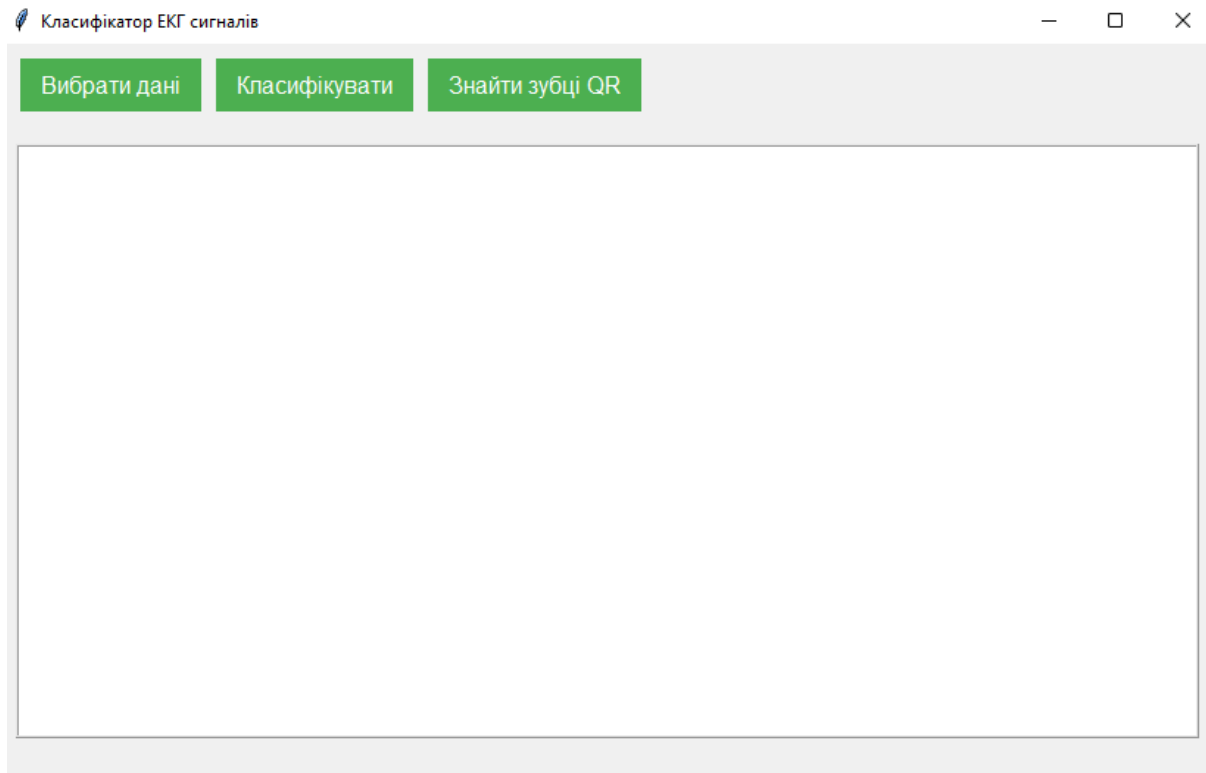


Рисунок 3.10 – Графічний інтерфейс настільного застосунку

Модуль обробки даних – модуль для обробки та підготовки ЕКГ сигналу до подачі на модель машинного навчання.

Завантаження та встановлення програмного забезпечення для класифікації ЕКГ сигналів за допомогою автокодувальної нейромережі на ваш комп'ютер є першим кроком. Після успішної інсталяції відкрийте програму і перейдіть до головного вікна інтерфейсу. Натисніть кнопку "Вибрати дані", щоб завантажити файл ЕКГ у форматі dat або atr. Після вибору файлу натисніть кнопку "Класифікувати", щоб запустити процес аналізу сигналу за допомогою нейромережі DenseNet.

Після завершення процесу класифікації програмне забезпечення відобразить результати: клас, до якого належить ЕКГ сигнал (наприклад,

нормальний, ішемія, інфаркт міокарда або інші патології), ймовірність класифікації для кожного типу патології та візуалізацію ЕКГ сигналу з позначками ймовірності для кожного класу.

Додаткові можливості програми можуть включати збереження результатів аналізу, налаштування параметрів моделі нейромережі та інтерактивну візуалізацію сигналу для детальнішого аналізу даних. Ця інструкція призначена для медичних фахівців, які шукають ефективні інструменти для автоматизованого аналізу та діагностики патологій серця з використанням сучасних методів машинного навчання (рисунок 3.11).



Рисунок 3.11 – Демонстрація ЕКГ-сигналу на графічному інтерфейсі

Ось детальний опис кожного типу ритму, що може бути класифіковано на основі ЕКГ-сигналів, разом з патологіями, які можуть відбуватися з ними.

Кожен з цих типів ритмів може відображати певні стани і патології серця, і їх класифікація дає змогу лікарям оцінити функцію серця і необхідність

подальшого обстеження або лікування пацієнта. На рисунку 3.10 показано демонстрацію ЕКГ-сигналу на графічному інтерфейсі.

Для більш детального аналізу ЕКГ сигналу користувач може скористатися функцією автоматичного визначення зубців QR. У головному вікні інтерфейсу користувача натисніть кнопку "Знайти зубці QR". Програмне забезпечення виконає аналіз сигналу і візуально виділить зубці QR на графіку ЕКГ, що допоможе в подальшій інтерпретації та діагностиці. На рисунку 3.12 показано результат кнопки «Знайти зубці QR».



Рисунок 3.12 – Демонстрація результату кнопки «Знайти зубці QR»

Для роботи програмного забезпечення, яке включає Jupyter Notebook та Anaconda Navigator, необхідно врахувати деталізовані вимоги до операційної системи, обсягу оперативної пам'яті, вільного місця на жорсткому диску, а також встановлення необхідних версій Python та інших компонентів.

Операційна система – Windows 10 або новіша. Програмне забезпечення розроблене для сумісності з операційною системою Windows 10 і вище. Це забезпечує сучасні можливості та оптимізацію для використання на оновлених платформах Microsoft. Обсяг оперативної пам'яті – 4 ГБ або більше.

Рекомендований обсяг оперативної пам'яті становить не менше 4 ГБ. Це забезпечує достатній запас для роботи з великими обсягами даних та запуску вимогливих програм, таких як Jupyter Notebook та Anaconda Navigator. Вільне місце на жорсткому диску – 1 ГБ або більше. Для встановлення програмного забезпечення та зберігання необхідних даних рекомендовано мати не менше 1 ГБ вільного місця на жорсткому диску. Це забезпечить достатню ємність для зберігання файлів Проектів, даних та робочих середовищ. Пакет програмного забезпечення Python 3.8 або новіший. Для виконання коду в Jupyter Notebook та використання платформи Anaconda Navigator важливо мати встановлену версію Python не нижче 3.8.

Загальна архітектура системи забезпечує високу зручність використання, надійність та швидкість обробки даних, що робить її ідеальним вибором для розроблення та тестування рішень у сфері аналізу ЕКГ-сигналів та медичного діагностування. Програмне забезпечення забезпечує автоматизоване визначення точок Q та R у ЕКГ сигналах, що є ключовим етапом у виявленні серцевих аномалій. Крім того, воно надає зручний інтерфейс для завантаження даних, їх класифікації та візуалізації.

Такий інструмент може значно спростити процес аналізу ритму серця та виявлення будь-яких відхилень у електрокардіографічних сигналах, забезпечуючи точність та швидкість у визначенні патологій. Використання сучасних методів машинного навчання, зокрема автокодувальних нейромереж, підвищує ефективність програми і робить її важливим інструментом для медичного діагностування.

Цей підхід дає змогу лікарям швидше і точніше аналізувати дані ЕКГ, що в свою чергу сприяє ранньому виявленню серцевих захворювань та вдосконаленню процесу лікування пацієнтів.

### 3.6 Висновки до розділу 3

У цьому розділі було описано програмну реалізацію способу ідентифікації патологій серця за сигналом електрокардіограми. Реалізація ґрунтується на використанні автокодувальної нейромережі DenseNet, яка навчається на великому наборі даних ЕКГ з анотаціями.

Створений на основі спроектованого способу настільний застосунок складається з декількох компонентів, включаючи модуль інтерфейсу користувача, який забезпечує зручний інтерфейс для взаємодії користувача з системою, модуль завантаження та попередньої обробки даних, який відповідає за завантаження ЕКГ-сигналів з файлів, їх попередню обробку та нормалізацію, модуль навчання моделі, який відповідає за навчання автокодувальної нейромережі DenseNet на наборі даних ЕКГ, модуль оцінювання та тестування моделі, який використовується для оцінки точності та надійності моделі на тестовому наборі даних, а також модуль візуалізації результатів, який використовується для візуалізації результатів аналізу ЕКГ-сигналів.

Система, розроблена з використанням мови програмування Python, Jupyter Notebook та платформи Anaconda Navigator, надає зручне та гнучке середовище для розроблення та тестування програмних компонентів.

Проведена оцінка показала, що розроблена система здатна успішно класифікувати ЕКГ-сигнали з високою точністю. Це свідчить про потенційні переваги використання нейромережі для аналізу серцевого ритму та виявлення патологій на основі ЕКГ-даних. Висока точність результатів підтверджує можливість застосування системи в клінічній практиці для покращення діагностики та лікування ССЗ.

## Загальні висновки

Результатом виконання кваліфікаційної роботи бакалавра стало досягнення мети роботи, а саме покращення якості ідентифікації патологій серця за сигналом ЕКГ через проектування способу ідентифікації патологій серця за сигналом ЕКГ та створення на його основі настільного застосунку.

У роботі спроектовано спосіб ідентифікації патологій серця за сигналом електрокардіограми, що ґрунтується на використанні автокодувальної нейромережі DenseNet. Проведене дослідження підтвердило низку переваг цієї нейронної мережі в порівнянні з наявними підходами. Нейромережева модель успішно класифікує ЕКГ-сигнали з високою точністю, що робить її перспективним інструментом для автоматичного діагностування ССЗ.

На основі спроектованого способу ідентифікації патологій серця було спроектовано архітектуру настільного застосунку зі зручним інтерфейсом користувача, який включає головне меню, панель інструментів, вікно відображення даних та панель інтерфейсу. Користувач може завантажувати сигнали ЕКГ, обробляти їх за допомогою різних функцій та зберігати результати.

Програмна реалізація виконана з допомогою мови програмування Python, Jupyter Notebook та платформи Anaconda Navigator, надає зручне та гнучке середовище для розроблення та тестування програмних компонентів.

Використання набору даних РТВ Diagnostic ECG Database (РТВ-db) дало змогу навчати та тестувати автокодувальну нейромережу, що сприяє поліпшенню її точності розпізнавати різні патології. Набір даних РТВ-db є одним із найбільш повних і надійних джерел для навчання моделей машинного навчання, що дає змогу забезпечити високу точність і надійність виявлення патологій серця.

З використанням автокодувальної нейромережі настільний застосунок дає змогу виявляти та аналізувати інформаційні аспекти ЕКГ-сигналів, що дає можливість лікарям краще розуміти результати аналізу та приймати інформовані рішення. Розроблений застосунок має потенціал значно поліпшити ранню

діагностику та успішність лікування ССЗ. Він може бути успішно впроваджений у клінічну практику та стати об'єктом подальших наукових досліджень у галузі автоматичного аналізу ЕКГ-сигналів.

Важливим аспектом програмної реалізації способу ідентифікації патологій серця є також стійкість системи до шуму та артефактів, що є критичним для її застосування в реальних умовах клінічної практики. Це дає змогу отримувати надійні результати навіть у важких умовах збору даних.

Проведене тестування програмної реалізації способу ідентифікації патологій серця у вигляді настільного застосунку продемонструвало, що подане рішення здатне успішно класифікувати ЕКГ-сигнали з високою точністю. Це свідчить про потенційні переваги використання автокодувальної нейромережі для аналізу серцевого ритму та виявлення патологій на основі ЕКГ-даних. Висока точність результатів підтверджує можливість застосування настільного застосунку в клінічній практиці для покращення діагностики та лікування ССЗ.

## Перелік посилань

1. Діагностика ССЗ: список необхідних процедур та сучасні методи виявлення хвороб. *UkrNova*. URL: <https://ukrnova.com/zdorovia/diagnostika-sertsevo-sudinnikh-zakhvoryuvan-spisok-neobkhdnikh-protседur-ta-suchasni-metodi-viyavlennya-khvorob.html>
2. ЕКГ (електрокардіографія): розшифровка, таблиця показників, результати. *Інформаційний блог А.Кіба: знання для гармонійних відносин з навколишнім світом*. URL: <https://akiba.com.ua/ekg-elektrokardiografiya-rozshifrovka-tablicya-pokaznikiv-rezultati/>
3. ECG signal classification using machine learning techniques / M. Kovalchuk et al. *Bulletin of Taras Shevchenko National University of Kyiv. Series: Physics and Mathematics*. 2022. No. 2. P. 70–77. URL: <https://doi.org/10.17721/1812-5409.2022/2.9>
4. Каплунова А.С. Метод та алгоритм обробки ЕКГ-сигналу в умовах невизначеності: кваліфікаційна робота магістра за спеціальністю “163 біомедична інженерія” / А.С. Каплунова. Тернопіль: ТНТУ, 2022. 54 с. URL: <https://elartu.tntu.edu.ua/handle/lib/39612>
5. Electrocardiogram data capturing system by using machine perception / U. M. Voppana et al. *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)* : Proceedings, Chennai, 25–26 April 2019. New York, NY, USA, 2019. P. 1–6. URL: <https://doi.org/10.1109/iciict1.2019.8741363>
6. Ведення хворого з порушенням провідності серця. Сучасна практика внутрішньої медицини з невідкладними станами : метод. вказ. для студентів та лікарів-інтернів / упоряд. О. Я. Бабак, Л. І. Овчаренко, Н. М. Железнякова та ін. Харків : ХНМУ, 2021. 20 с.
7. Савіцька Ю.В., Кланца А.І., Щепіна Н.В. Інфаркт міокарда правого шлуночка. *Кардіохірургія та інтервенційна кардіологія*. 2022. № 1–2. С. 69–73. URL: <http://doi.org/10.31928/2305-3127-2022.1-2.6973>

8. Analysis of deep learning methods in adaptation to the small data problem solving / I. Krak et al. In: Babichev, S., Lytvynenko, V. (eds) *Lecture Notes in Data Engineering, Computational Intelligence, and Decision Making. ISDMCI–2022*. Springer, Cham. 2023. Vol. 149. P. 333–352. URL: [https://doi.org/10.1007/978-3-031-16203-9\\_20](https://doi.org/10.1007/978-3-031-16203-9_20)
9. A novel feature vector for ECG classification using deep learning / Kovalchuk O. et al. *The 4th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntellITSIS-2023)* : CEUR-Workshop Proceedings. Vol. 3373. (Khmelnyskyi, Ukraine, 22–24 March 2023). CEUR-WS.org, Aachen, 2023. P. 227–238. URL: <https://CEUR-WS.org/Vol-3373/paper12.pdf>
10. Belkadi M. A., Daamouche A., Melgani F. A deep neural network approach to QRS detection using autoencoders. *Expert Systems with Applications*. 2021. Vol. 184. P. 115528. URL: <https://doi.org/10.1016/j.eswa.2021.115528>
11. Saini S. K., Gupta R. Artificial intelligence methods for analysis of electrocardiogram signals for cardiac abnormalities: state-of-the-art and future challenges. *Artificial Intelligence Review*. 2021. Vol. 55. P. 1519–1565. URL: <https://doi.org/10.1007/s10462-021-09999-7>
12. Robust R-peak detection using deep learning based on integrating domain knowledge / O. Kovalchuk et al. *The 6th International Conference on Informatics & Data-Driven Medicine (IDDM-2023)* : CEUR-Workshop Proceedings. Vol. 3609. (Bratislava, Slovakia, 17–19 November 2023) / ed. by N. Shakhovska et al. CEUR-WS.org, Aachen, 2024. P. 1–14. URL: <https://ceur-ws.org/Vol-3609/paper1.pdf>
13. Bank D., Koenigstein N., Giryas R. Autoencoders. *Machine Learning for Data Science Handbook*. Cham, Springer.org, 2023. P. 353–374. URL: [https://doi.org/10.1007/978-3-031-24628-9\\_16](https://doi.org/10.1007/978-3-031-24628-9_16)
14. Li P., Pei Y., Li J. A comprehensive survey on design and application of autoencoder in deep learning. *Applied Soft Computing*. 2023. P. 110176. URL: <https://doi.org/10.1016/j.asoc.2023.110176>
15. Radiuk P., Pavlova O., Hrypynska N. An ensemble machine learning approach for Twitter sentiment analysis. *The 6th International Conference on*

*Computational Linguistics and Intelligent Systems (COLINS-2022)*. Volume I: Main Conference : CEUR-Workshop Proceedings. Vol. 3171. (Gliwice, Poland, 12–13 May 2022). CEUR-WS.org, Aachen, 2022. P. 387–397. URL: <http://CEUR-WS.org/Vol-3171/paper32.pdf>

16. Atrial fibrillation detection using a feedforward neural network / Y. Chen et al. *Journal of Medical and Biological Engineering*. 2022. Vol. 42, no. 1. P. 63–73. URL: <https://doi.org/10.1007/s40846-022-00681-z>

17. Bui T. H., Hoang V. M., Pham M. T. Automatic varied-length ECG classification using a lightweight DenseNet model. *Biomedical Signal Processing and Control*. 2023. Vol. 82. P. 104529. URL: <https://doi.org/10.1016/j.bspc.2022.104529>

18. Radiuk P.M. Applying 3D U-Net architecture to the task of multi-organ segmentation in computed tomography. *Applied Computer Systems*. 2020. Vol. 25, No. 1, Pp. 43–50. (WoS, Q4). DOI: <https://doi.org/10.2478/acss-2020-0005>

19. An enhanced ResNet-50 deep learning model for arrhythmia detection using electrocardiogram biomedical indicators / R. Anand et al. *Evolving Systems*. 2023. Vol. 15. P. 83–97. URL: <https://doi.org/10.1007/s12530-023-09559-0>

20. Електрокардіографія високого розрізнення в діагностиці аритмій та прогнозуванні раптової серцевої смерті. *Compendium.ua*. Режим доступу: <https://compendium.com.ua/uk/clinical-guidelines-uk/cardiology-uk/section-5-uk/glava-6-elektrokardiografiya-visokogo-rozrznennya-v-diagnostitsi-aritmij-ta-prognozuvanni-raptovoyi-sertsevoyi-smerti/>

21. Cardioscape Research. *Escardio.org*. Режим доступу: <https://www.escardio.org/Research/Cardioscape>

22. CardioNet. *GoBio.com*. Режим доступу: <https://www.gobio.com/brand/cardionet/>

23. ECG-NET: A deep LSTM autoencoder for detecting anomalous ECG / M. Roy et al. *Engineering Applications of Artificial Intelligence*. 2023. Vol. 124. P. 106484. URL: <https://doi.org/10.1016/j.engappai.2023.106484>

24. The Evaluation Cooperation Group | *Evaluation Cooperation Group*. URL: <https://www.ecgnet.org/>

25. Identifying heart failure in ECG data with artificial intelligence—A meta-analysis / D. Grün et al. *Frontiers in Digital Health*. 2021. Vol. 2. P. 584555. URL: <https://doi.org/10.3389/fdgth.2020.584555>

26. Бармак О.В., Радюк П.М. Інформаційна технологія візуального аналізу рентгенівських зображень для інтерпретації результатів діагностування пневмонії. *Вісник Хмельницького національного університету. Технічні науки*. 2021. № 295(2). С. 52–55. DOI: <https://doi.org/10.31891/2307-5732-2021-295-2-52-55>

27. Densenet. *PyTorch*. URL: [https://pytorch.org/hub/pytorch\\_vision\\_densenet/](https://pytorch.org/hub/pytorch_vision_densenet/)

28. Embranchment CNN based local climate zone classification using SAR and multispectral remote sensing data / P. Feng et al. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium : Proceedings, Yokohama, 28 July – 2 August 2019. New York, NY, USA, 2019*. P. 6344–6347. URL: <https://doi.org/10.1109/igarss.2019.8898703>

29. Navigator | Anaconda. *Anaconda.com*.  
URL: <https://www.anaconda.com/products/navigator>

30. Welcome to Python.org. *Python.org*. URL: <https://www.python.org/>

# ДОДАТКИ

## Додаток А

### Лістинг програмного коду

```
import tkinter as tk
from tkinter import filedialog, messagebox
import os
import numpy as np
import wfdb
from keras.models import load_model
from scipy.signal import find_peaks
class ECGClassifierGUI:
    def __init__(self):
        self.root = tk.Tk()
        self.root.title("Класифікатор ЕКГ сигналів")
        bg_color = "#F0F0F0"
        button_color = "#4CAF50"
        font_style = "Helvetica"
        button_style = {"font": (font_style, 12), "bg": button_color, "fg": "white", "bd": 0,
"padx": 10, "pady": 5}
        label_style = {"font": (font_style, 12), "bg": bg_color}
        self.root.configure(bg=bg_color)
        toolbar = tk.Frame(self.root, bg=bg_color)
        toolbar.grid(row=0, column=0, columnspan=3, sticky="ew", padx=10, pady=10)
        self.btn_select_data = tk.Button(toolbar, text="Вибрати дані",
command=self.select_data, **button_style)
        self.btn_classify = tk.Button(toolbar, text="Класифікувати",
command=self.classify_signal, **button_style)
        self.btn_find_peaks = tk.Button(toolbar, text="Знайти зубці QR",
command=self.find_peaks, **button_style)
        self.btn_select_data.pack(side="left", padx=5)
```

```

self.btn_classify.pack(side="left", padx=5)
self.btn_find_peaks.pack(side="left", padx=5)
self.canvas = tk.Canvas(self.root, width=800, height=400, bg="white", bd=2,
relief="ridge")
self.canvas.grid(row=1, column=0, columnspan=3, padx=10, pady=10,
sticky="nsew")
info_panel = tk.Frame(self.root, bg=bg_color)
info_panel.grid(row=2, column=0, columnspan=3, sticky="ew", padx=10,
pady=10)
self.result_label = tk.Label(info_panel, text="", **label_style)
self.info_label = tk.Label(info_panel, text="", justify="left", **label_style)
self.coordinate_label = tk.Label(info_panel, text="", justify="left", **label_style)
self.result_label.pack(anchor="w", pady=5)
self.info_label.pack(anchor="w", pady=5)
self.coordinate_label.pack(anchor="w", pady=5)
self.root.grid_columnconfigure(0, weight=1)
self.root.grid_rowconfigure(1, weight=1)
self.model = None
self.data = None
self.load_model()
self.root.mainloop()
def load_model(self):
    model_path = 'best_model.keras'
    self.model = load_model(model_path)
def select_data(self):
    file_path = filedialog.askopenfilename(filetypes=[("Дані MIT-BIH", "*.atr")])
    if file_path:
        self.result_label.configure(text="Обрані дані: " + file_path)
        record_name = os.path.splitext(os.path.basename(file_path))[0]
        header_path = os.path.join(os.path.dirname(file_path), record_name + '.hea')

```

```

if not os.path.isfile(header_path):
    messagebox.showerror("Помилка", "Файл заголовка не знайдено для
обраного запису.")
    return
record_path = os.path.join(os.path.dirname(file_path), record_name)
record = wfdb.rdrecord(record_path)
self.data = record.p_signal[:, 0]
self.display_info(record)
self.display_coordinates(record)
self.plot_signal()
def plot_signal(self):
    if self.data is None:
        messagebox.showerror("Помилка", "Не вибрано жодних даних. Будь ласка,
спочатку виберіть дані.")
        return
    self.canvas.delete("all")
    signal = self.data - np.mean(self.data)
    signal /= np.max(np.abs(signal))
    signal *= 180
    x = np.arange(len(signal))
    y = 200 - signal
    for i in range(len(x) - 1):
        self.canvas.create_line(x[i], y[i], x[i + 1], y[i + 1], fill="blue")
def display_info(self, record):
    info = f"Назва запису: {record.record_name}\n" \
        f"Кількість каналів: {record.n_sig}\n" \
        f"Частота дискретизації: {record.fs} Гц"
    self.info_label.configure(text=info)
def display_coordinates(self, record):
    coordinates = ""

```

```

for i, sig_name in enumerate(record.sig_name):
    coordinates += f"Канал {i+1}: {sig_name}\n"
self.coordinate_label.configure(text=coordinates)
def classify_signal(self):
    if self.data is None:
        messagebox.showerror("Помилка", "Класифікація є неможливою. Жодних
даних не вибрано. Спочатку виберіть дані.")
        return
    X = self.data[:186].reshape(1, -1, 1)
    y_pred = self.model.predict(X)
    class_labels = ['«N»: Non-ectopic beats (нормальний ритм серця)', '«S»:
Supraventricular ectopic beats (суправентрикулярні ектопічні ритми)', '«V»:
Ventricular ectopic beats (шлунково-шлуночкові ектопічні ритми)', '«F»: Fusion
Beats (змішані ритми).', '«Q»: Unknown Beats (невідомий тип ритму)']
    prediction = class_labels[np.argmax(y_pred)]
    self.result_label.configure(text="Клас: " + prediction)
def find_peaks(self):
    if self.data is None:
        messagebox.showerror("Помилка", "Зубці неможливо знайти. Жодних
даних не вибрано. Спочатку виберіть дані.")
        return
    signal = self.data - np.mean(self.data)
    signal /= np.max(np.abs(signal))
    signal *= 180
    r_peaks, _ = find_peaks(signal, prominence=0.5, distance=100)
    q_peaks, _ = find_peaks(-signal, prominence=0.5, distance=100)
    self.plot_signal()
    for peak in r_peaks:
        x = peak
        y = 200 - signal[peak]

```

```

        self.canvas.create_text(x, y, text="R", fill="red", font=("Helvetica", 8),
anchor="s")
    for peak in q_peaks:
        x = peak
        y = 200 - signal[peak]
        self.canvas.create_text(x, y, text="Q", fill="green", font=("Helvetica", 8),
anchor="s")
if __name__ == "__main__":
    ecg_classifier_gui = ECGClassifierGUI()
import os
import numpy as np
import keras
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from keras.layers import Input, Convolution1D, BatchNormalization, MaxPool1D,
Flatten, Dense
from keras.models import Model
from keras.callbacks import EarlyStopping, ModelCheckpoint
from sklearn.utils import resample
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
from keras.utils import to_categorical
from sklearn.utils import class_weight
import warnings
warnings.filterwarnings('ignore')
train_df = pd.read_csv('D:/AI/mitbih_train.csv', header=None)
test_df = pd.read_csv('D:/AI/mitbih_test.csv', header=None)

```

```

train_df[187] = train_df[187].astype(int)
equilibre = train_df[187].value_counts()
print(equilibre)
df_1 = train_df[train_df[187] == 1]
df_2 = train_df[train_df[187] == 2]
df_3 = train_df[train_df[187] == 3]
df_4 = train_df[train_df[187] == 4]
df_0 = (train_df[train_df[187] == 0]).sample(n=20000, random_state=42)
df_1_upsample = resample(df_1, replace=True, n_samples=20000, random_state=123)
df_2_upsample = resample(df_2, replace=True, n_samples=20000, random_state=124)
df_3_upsample = resample(df_3, replace=True, n_samples=20000, random_state=125)
df_4_upsample = resample(df_4, replace=True, n_samples=20000, random_state=126)
train_df = pd.concat([df_0, df_1_upsample, df_2_upsample, df_3_upsample,
df_4_upsample])
equilibre = train_df[187].value_counts()
print(equilibre)
train_df.iloc[:, :186] = train_df.iloc[:, :186].apply(lambda x: (x - x.min()) / (x.max() -
x.min()))
test_df.iloc[:, :186] = test_df.iloc[:, :186].apply(lambda x: (x - x.min()) / (x.max() -
x.min()))
X_train = train_df.iloc[:, :186].values
y_train = train_df.iloc[:, 186].values
X_test = test_df.iloc[:, :186].values
y_test = test_df.iloc[:, 186].values
num_classes = 5
y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.1,
random_state=42)
def build_model(input_shape):

```

```

inputs = Input(shape=input_shape)
x = Convolution1D(32, 5, activation='relu')(inputs)
x = BatchNormalization()(x)
x = MaxPool1D(pool_size=2)(x)
x = Flatten()(x)
x = Dense(64, activation='relu')(x)
outputs = Dense(num_classes, activation='softmax')(x)
model = Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
return model

model = build_model((186, 1))
history = model.fit(X_train, y_train, epochs=100, batch_size=32,
validation_data=(X_val, y_val))
save_path = 'best_model.keras'
model.save(save_path)
print(f'Модель успешно сохранена по пути: {save_path}')
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test loss: {loss}, Test accuracy: {accuracy}')
y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true_classes = np.argmax(y_test, axis=1)
print(classification_report(y_true_classes, y_pred_classes))
cm = confusion_matrix(y_true_classes, y_pred_classes)
sns.heatmap(cm, annot=True, fmt='d')
plt.show()

```

## Додаток Б

### Презентаційний матеріал

---

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

# Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі

**Виконав:**

студент 4 курсу, групи КН-20-1

**Гамбарян Арсен Араїкович**

**Керівник:**

док. філ., ст. викл. каф. КН

**Радюк Павло Михайлович**

2

---

## Актуальність роботи

Серцево-судинні захворювання (ССЗ) є однією з основних причин смертності та інвалідності у світі. За даними Всесвітньої організації охорони здоров'я, у 2024 році від ССЗ померло 17,9 мільйона людей. В Україні ССЗ також є однією з головних причин смертності, на їх частку припадає близько 60% всіх летальних випадків захворювань. Рання діагностика та лікування ССЗ є ключовими факторами зниження їх смертності. Електрокардіографія (ЕКГ) є одним з найпоширеніших методів діагностики ССЗ.

ЕКГ - це метод запису електричної активності серця. За допомогою ЕКГ можна виявити різні патології серця, такі як ішемічна хвороба серця, інфаркт міокарда, аритмії та інші.

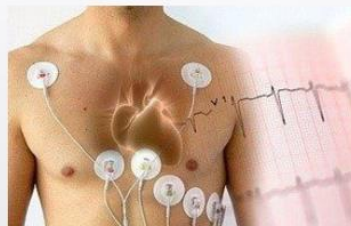
Традиційний аналіз ЕКГ виконується лікарями-кардіологами, що потребує багато часу та зусиль. Автоматичний аналіз ЕКГ з використанням методів машинного навчання може значно полегшити та прискорити процес діагностики ССЗ.

## Мета роботи

**Мета кваліфікаційної роботи бакалавра** покращення якості ідентифікації патологій серця за сигналом електрокардіограми. Завдання кваліфікаційної роботи бакалавра:

1. Провести аналіз нейромережевих методів для оброблення сигналів електрокардіограм.
2. Спроекувати спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі.
3. Спроекувати архітектуру настільного застосунку для програмної реалізації способу ідентифікації патологій серця за сигналом електрокардіограми.
4. Розробити настільний застосунок за спроектованим способом ідентифікації патологій серця.
5. Провести експериментальне тестування створеного настільного застосунку за еталонними наборами даних.
6. Продемонструвати покращення якості ідентифікації патологій серця за сигналом електрокардіограми через впровадження створеного настільного застосунку.

## Електрокардіограма

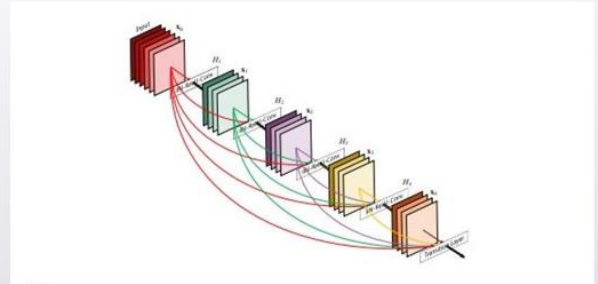


Електрокардіограма (ЕКГ) - це один із основних методів діагностики серця. Вона дозволяє зафіксувати частоту, періодичність та інтенсивність серцевих скорочень. Якщо всі показники в нормі, найімовірніше пацієнт здоровий, тоді як наявність відхилень говорить про захворювання.

5

## Проектування архітектури автокодувальної нейромережі для окреслення сигналів

- У цій кваліфікаційній роботі розглядається використання нейромережі DenseNet для аналізу сигналів електрокардіограм. Використання DenseNet обґрунтовується її перевагами, що можуть призвести до поліпшення результатів у виявленні патологій серця
- DenseNet - (Densely Connected Convolutional Networks) є типом глибокої згорткової нейронної мережі, яка відома своїми щільними зв'язками між шарами. Ця архітектура була розроблена для подолання проблем зникаючих градієнтів і поліпшення потоку градієнтів у глибоких нейронних мережах, що часто виникають під час тренування



6

## Функціональна структура інформаційної системи на основі автокодувальної нейромережі

Основна мета застосунку – отримання ЕКГ результату, зручним способом для визначення патологій, а також покращення та спрощення цього процесу для рядового користувача.

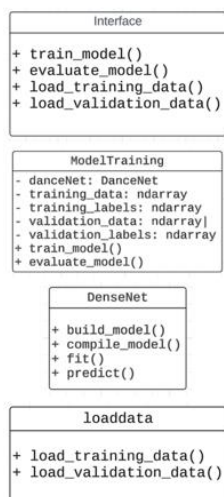


## Набір даних

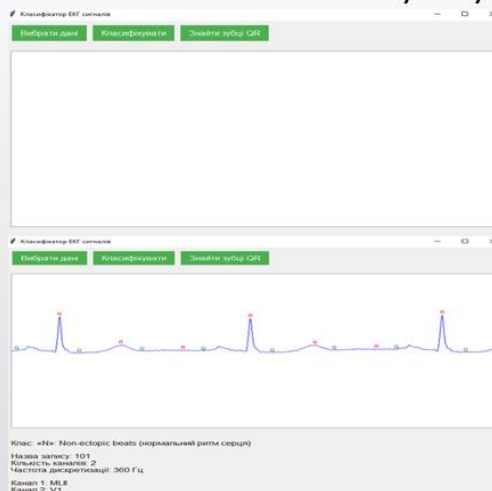
Набір даних PTB [Diagnostic ECG Database](#) (PTB-db) є одним з найбільших та найвідоміших наборів даних ЕКГ, що використовується для досліджень та розробки методів аналізу ЕКГ. Цей набір містить записи ЕКГ від понад 200 пацієнтів з різними серцевими захворюваннями, а також від здорових людей. Записи оцифровані з частотою дискретизації 250 Гц і містять 12 [відведень](#).

Використання набору даних PTB-db в інформаційній системі передбачає навчання та тестування автокодувальної нейромережі. Для цього з набору [вибираємо](#) підмножину записів ЕКГ для навчання, а інші записи використовувати для тестування. Це дозволяє розробляти та вдосконалювати методи аналізу ЕКГ, забезпечуючи високу точність і надійність результатів.

## Діаграма класів



## Реалізація застосунку



## Висновки

Результатом виконання кваліфікаційної роботи бакалавра стало досягнення мети роботи, а саме покращення якості ідентифікації патологій серця за сигналом ЕКГ шляхом проєктування способу ідентифікації патологій серця за сигналом ЕКГ та створення на його основі настільного застосунку.

У роботі спроєктовано спосіб ідентифікації патологій серця за сигналом електрокардіограми, що ґрунтується на використанні автокодувальної нейромережі DenseNet. Нейромережева модель успішно класифікує ЕКГ-сигнали з високою точністю, що робить її перспективним інструментом для автоматичного діагностування ССЗ.

На основі спроєктованого способу ідентифікації патологій серця було спроєктовано архітектуру настільного застосунку зі зручним інтерфейсом користувача, який включає головне меню, панель інструментів, вікно відображення даних та панель інтерфейсу. Користувач може завантажувати сигнали ЕКГ, обробляти їх за допомогою різних функцій та зберігати результати.

Проведене тестування програмної реалізації способу ідентифікації патологій серця у вигляді настільного застосунку продемонструвало, що подане рішення здатне успішно класифікувати ЕКГ-сигнали з високою точністю.



ДЯКУЮ ЗА УВАГУ

# Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 14%

ID: 132269 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодуювальної нейромережі Додано в БД: 2024-06-23 Автора: Арсен ГАМБАРЯН Керівники: Павло РАДЮК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	80710	1220	4291 (5%)	68 (6%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:  
Кафедра КН

Дата перевірки:  
23.06.2024 16:11:57 EEST

Дата звіту:  
23.06.2024 16:14:21 EEST

ID перевірки:  
1016383683

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100005671

Назва документа: КН-20-1 Гамбарян\_ЗАПИСКА

Кількість сторінок: 71 Кількість слів: 12879 Кількість символів: 102802 Розмір файлу: 1.58 MB ID файлу: 1016194158

## 11.8% Схожість

Найбільша схожість: 4.2% з джерелом з Бібліотеки (ID файлу: 1014975805)

9.03% Джерела з Інтернету	806	Сторінка 73
8.85% Джерела з Бібліотеки	136	Сторінка 77

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 2

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі

Автор: студент групи КН-20-1 Гамбарян Арсен

Спеціальність: 122 Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: док. філ., ст. викл. Радюк П.М.

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.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в 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 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в 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 допрацьована 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.	

*Підтвердження:*

*Запозичення, що виявлені в роботі Гамбаряна А.А., не є плагіатом, оскільки: запозичення знайдені в розділі огляду наявних методів та не описують безпосередньо авторську роботу й не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять цитування джерел у переліку посилань; поміж запозичень є загальновідомі терміни та скорочення.*

*Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:*

*– за системою Anti-Plagiarism: 2.0 %;*

*– за системою Unichек: 11.8 %.*

*Отже, запозичення є допустимими та належать до описаних вище й адресуються до першоджерел, що, з урахуванням наведених обґрунтувань, свідчить на користь кваліфікаційної роботи.*

Керівник роботи



Павло РАДЮК

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



**ВІДГУК НАУКОВОГО КЕРІВНИКА  
на кваліфікаційну роботу бакалавра**

студента КН-20-1 Гамбаряна Арсена Араїковича  
за темою Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі

**1. Актуальність теми**

Ідентифікація патологій серця за сигналами ЕКГ є важливою для боротьби із серцево-судинними захворюваннями. Розроблення інформаційної системи для аналізу ЕКГ із використанням автокодувальних нейромереж може суттєво покращити діагностику. Впровадження таких передових підходів є необхідним кроком для вдосконалення медичних інформаційних систем та покращення якості кардіологічної допомоги.

**2. Відповідність роботи предметній області Стандарту спеціальності  
122 Комп'ютерні науки**

Відповідно до стандарту бакалавра вищої освіти України спеціальності 122 Комп'ютерні науки, описом предметної галузі, об'єктом та предметом вивчення є математичні, інформаційні та імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи й технології отримання, зберігання, обробки, передачі та використання інформації. Метою поданої роботи є покращення якості ідентифікації патологій серця за сигналом електрокардіограми. Мету роботи досягнуто завдяки використанню методів, засобів та технологій розв'язання теоретичних і прикладних задач, що виникають у процесі проєктування та розроблення інформаційних технологій. Як наслідок, результати виконання кваліфікаційної роботи відповідають стандарту бакалавра спеціальності 122 Комп'ютерні науки.

**3. Професійні та особистісні якості бакалавра**

Під час виконання кваліфікаційної роботи студент Гамбарян А.А. проявив себе кваліфікованим фахівцем та відповідальним студентом. Студент опанував компетентності та засвоїв результати навчання, що відповідають виконанню освітньо-професійної програми рівня вищої освіти «Бакалавр» за спеціальністю 122 Комп'ютерні науки.

**4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Студент особисто одержав результати кваліфікаційної роботи та обґрунтував їхню практичну значущість унаслідок виконання ним усіх поставлених завдань.

### **5. Ступінь опанування методами дослідження**

У процесі проєктування способу ідентифікації патології серця за сигналом електрокардіограми з використанням автокодувальної нейромережі та розроблення на його основі настільного застосунок студент Гамбарян А.А. продемонстрував задовільний рівень компетентностей та володіння необхідними інструментами й обладнанням, методами, методиками та технологіями галузі 12 Інформаційні технології.

### **6. Повнота та якість розкриття теми роботи**

Тема роботи достатньо обґрунтована й розкрита; актуальність предметної галузі та відомі дослідження проаналізовані повною мірою. Студент виконав усі поставлені перед ним завдання. Розроблений настільний застосунок для валідування способу ідентифікації патології серця за сигналом електрокардіограми відповідає технічним вимогам спеціальності 122 Комп'ютерні науки.

### **7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу**

Структура роботи та послідовність її викладення є логічними та такими, що відповідають поставленій меті. Викладення матеріалу є послідовним, аргументованим та літературно грамотним.

### **8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин**

Спроєктований у роботі спосіб ідентифікації патології серця за сигналом електрокардіограми та його програмна реалізація може бути використаний медичними закладами для покращення процесу виявлення патологій серця за ЕКГ-сигналом.

### **9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота**

З огляду на рівень виконання та забезпечення всіх необхідних вимог, вважаю, що кваліфікаційна робота Гамбаряна Арсена Араїковича може бути допущена до захисту. Рекомендована оцінка – «задовільно».

Керівник



док. філ., ст. викл. каф. КН Павло РАДЮК



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента КН-20-1 Гамбаряна Арсена Араїковича  
за темою: Спосіб ідентифікації патологій серця за сигналом електрокардіограми з використанням автокодувальної нейромережі

1. Актуальність обраної теми

Серцево-судинні захворювання залишаються головною причиною смертності у світі, що підкреслює важливість точної та своєчасної діагностики. Автоматизований аналіз ЕКГ із використанням сучасних технологій стає ключовим інструментом у боротьбі із цією проблемою. Прокрування способів для ідентифікації патологій серця на основі ЕКГ-сигналів із використанням автокодувальних нейромереж є необхідною для покращення якості кардіологічної допомоги.

2. Повнота розкриття мети та завдань роботи

У роботі достатньо розкрито мету та завдання дослідження. Мету роботи досягнуто в результаті прокрування способу ідентифікації патологій серця за сигналом ЕКГ та створення на його основі настільного застосування.

3. Зміст кожного розділу роботи

У першому розділі роботи виконано аналіз сучасних підходів до ідентифікації патологій серця за сигналами електрокардіограм, методів інтелектуального аналізу даних та сучасних тенденцій у даній галузі. Визначено мету роботи та поставлено завдання. У другому розділі спроектовано спосіб ідентифікації патологій серця, що передбачає обробку сигналів ЕКГ із використанням автокодувальної нейромережі. Визначено критерії оцінки якості ідентифікації. У третьому розділі виконано програму реалізацію способу та проведено її тестування за визначеними критеріями.

4. Оцінка розробленого програмного забезпечення, його практична цінність

Розроблений спосіб ідентифікації патологій серця та його програмна реалізація у вигляді настільного застосування можуть бути використані в медичних закладах для покращення якості діагностування серцево-судинних захворювань.

5. Якість оформлення кваліфікаційної роботи бакалавра

Кваліфікаційна робота бакалавра оформлена відповідно до вимог. Виклад матеріалу структурований та логічно послідовний. Текст роботи написаний чітко та зрозуміло, з використанням наукової термінології.

6. Недоліки кваліфікаційної роботи бакалавра

Записка до роботи містить низку орфографічних та граматичних помилок, які негативно впливають на читабельність. Автор надав недостатньо посилань на авторитетні джерела, що підтримували б його аргументи. У роботі не проведено повною мірою експериментальні обчислення щодо покращення якості ідентифікації патологій серця за сигналом електрокардіограми на основі спроектованого способу.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка, на яку заслуговує кваліфікаційна робота.

З огляду на рівень виконання та забезпечення всіх необхідних вимог, вважаю, що подана кваліфікаційна робота бакалавра може бути допущена до захисту.

Рекомендована оцінка – «задовільно».

Рецензент

Гадельчук Тамара Іванівна, канд. техн. наук, доцент кафедри ІТІЗ ХНУ  
21.06.2024р