

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:

Автоматизована система аналізу сигналів електрокардіограм

Виконав: студент 2 курсу, групи
2АКІТ-22м спеціальності 151 –
Автоматизація та комп'ютерно-
інтегровані технології



Євгеній СЛОБОДЯНЮК
Ім'я ПРІЗВИЩЕ

Керівник:

к.т.н., доцент кафедри КСУ
ступінь, звання, посада



Олена НИКИТЕНКО
Ім'я ПРІЗВИЩЕ

« 1 » 12 2023 р.

Опонент: к.т.н., доцент кафедри АІТ
ступінь, звання, посада

Юрій ІВАНОВ
Ім'я ПРІЗВИЩЕ

« 5 » 12 2023 р.

Допущено до захисту
Т.в.о. Зав. кафедри КСУ
Марія ЮХИМЧУК
« 7 » 12 2023

Вінницький національний технічний університет
 Факультет інтелектуальних інформаційних технологій та автоматизації
 Кафедра комп'ютерних систем управління
 Рівень вищої освіти другий (магістерський)
 Галузь знань – 15 – Автоматизація та приладобудування
 Спеціальність – 151 – Автоматизація та комп'ютерно-інтегровані технології
 Освітньо - професійна програма – Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ

Т.в.о.Зав. кафедри КСУ



Марія ЮХИМЧУК

“09” жовтня 2023 року

**ЗАВДАННЯ
 НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ**



студенту Слободянюку Євгенію Сергійовичу
 (прізвище, ім'я, по батькові)

1. Тема роботи. Автоматизована система аналізу сигналів електрокардіограм керівник роботи Никитенко Олена Дмитрівна
 затверджені наказом ВНТУ від “18” вересня 2023 року №247
2. Термін подання студентом роботи “1” грудня 2023 року
3. Вихідні дані до роботи: збір інформації для формування навчального датасету, оптимізація алгоритму навчання, керування базою даних, формування звітів.

4. Зміст текстової частини: вступ, виявлення інформаційних потреб прикладної області, відомі методи класифікації та аналізу даних, розробка методу класифікації та аналізу даних, програмна реалізація методу, дослідження ефективності методу

5. Перелік ілюстративного матеріалу
схема нормального запису ЕКГ сегменту, задачі, приклад PQRS, метод локальних мінімумів та максимумів, алгоритм поділу на R-інтервали, алгоритм розбиття ЕКГ на опорні точки інтервалів, схема структурна класів, концепція шаблону Model-View-Presenter, окремі результати роботи програмного забезпечення, аналіз ефективності запропонованого підходу, матриця оцінки результатів алгоритму класифікації

Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
6	Бурснікова Н. В., д.е.н., професор кафедри ЕПВМ.		
3 . 4	Ковтун В. В., д.т.н., професор, зав. кафедри КСУ		

Дата видачі завдання "09" жовтня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Ознайомлення з предметною областю. Вивчення літератури	9.10.2023	16.10.2023	
2	Аналіз наявних даних та методів	17.10.2023	21.10.2023	
3	Аналіз вимог до програмного забезпечення	22.10.2023	28.10.2023	
4	Розробка архітектури програмного забезпечення	29.10.2023	20.11.2023	
5	Тестування програмного забезпечення	21.11.2023	26.11.2023	
6	Оформлення матеріалів для захисту	27.11.2023	28.11.2023	

Студент



(підпис)

Євгеній Слободянюк

(Ім'я ПРІЗВИЩЕ)

Керівник роботи



(підпис)

Олена НИКИТЕНКО

(Ім'я ПРІЗВИЩЕ)

АНОТАЦІЯ

УДК 004.93

Слободянюк Є. С. Автоматизована система аналізу сигналів електрокардіограм. Магістерська кваліфікаційна робота зі спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології, освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2023.

На укр. мові. Бібліогр.: 51 назв; рис.: 41; табл. 30.

У ході виконання роботи була розроблена автоматизована система аналізу сигналів електрокардіограм. У вступі обґрунтовується актуальність дослідження, розглядаються основні фактори ризику серцево-судинних захворювань. У наступних розділах роботи детально вивчаються електрокардіографічні методи та їх застосування, аналізується існуюче програмне забезпечення в цій області, і розкриваються проблеми, пов'язані із шумовим сигналом, недостатньою точністю та обмеженою базовою симптоматикою серцевих аномалій. Серцевинною роботи є розробка алгоритму для аналізу електрокардіограми, включаючи їхню класифікацію за допомогою методів машинного навчання. Робота охоплює вивчення та порівняння існуючих алгоритмів, створення власного алгоритму, реалізацію його в автономній частині бібліотеки та розробку програмного забезпечення для визначення захворювань за електрокардіограмою.

Ключові слова: ЕКГ, електрокардіограма, електрокардіограф, машинне навчання, штучний інтелект, аналіз.

ABSTRACT

UDC 004.93

Slobodyaniuk E. S. Automated system of analysis of electrocardiogram signals. Master's thesis on specialty 151 - Automation and computer-integrated technologies, educational program - Intelligent computer systems. Vinnytsia: VNTU, 2023.

In Ukrainian speech Bibliography: 51 titles; Fig.: 41; table 30.

In the course of the work, an automated system for analyzing electrocardiogram signals was developed. The introduction substantiates the relevance of the research, considers the main risk factors of cardiovascular diseases. In the following sections, electrocardiographic methods and their applications are studied in detail, existing software in this area is analyzed, and problems related to signal noise, insufficient accuracy, and limited baseline symptomatology of cardiac abnormalities are revealed. The core of the work is the development of an algorithm for electrocardiogram analysis, including their classification using machine learning methods. The work covers the study and comparison of existing algorithms, the creation of an own algorithm, its implementation in the autonomous part of the library, and the development of software for the detection of diseases by electrocardiogram.

Keywords: ECG, electrocardiogram, electrocardiograph, machine learning, artificial intelligence, analysis.

ЗМІСТ

ВСТУП.....	8
1. ВИЯВЛЕННЯ ІНФОРМАЦІЙНИХ ПОТРЕБ ПРИКЛАДНОЇ ОБЛАСТІ...	11
1.1 Опис даних електрокардіограм	11
1.2 Норми показників ЕКГ у людей в таблицях та критерії аналізу стану ..	15
2. ВІДОМІ МЕТОДИ КЛАСИФІКАЦІЇ ТА АНАЛІЗУ ДАНИХ.....	20
2.1 Методи виділення опорних точок	20
2.2 показано результати цього методу Переваги цього методу:.....	25
2.3 Різноманітні підходи до глибинного дослідження числового ряду	27
2.3.1 Прогнозування часових рядів	28
2.3.2 Класифікатор за піками та їх протяжністю.....	30
2.3.3 Класифікатор за допомогою методу кластеризації	33
3. РОЗРОБКА МЕТОДУ КЛАСИФІКАЦІЇ ТА АНАЛІЗУ ДАНИХ.....	39
3.1 Постановка завдання	39
3.2 Метод модифікації вхідних даних.....	39
3.2.1 Вейвлет Хаара.....	40
3.2.2 Алгоритм виявлення періоду.....	41
3.3 Метод аналізу даних для проектування програмного забезпечення.....	43
3.3.1 Метод аналізу числових даних	44
3.4 наведено алгоритм розбиття кожної пари інтервалів R-R на числові показники:	44
3.4.1 Метод аналізу інтервальних даних.....	45
3.5 Алгоритм класифікації числових даних для проектування ПЗ	49
3.6 Алгоритм класифікації інтервальних даних для проектування ПЗ	52
4. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ	56
4.1 Вимоги до програмного забезпечення	56
4.2 Засоби розробки	57
4.2.1 ML.NET[15].....	58
4.2.2 WINDOWS FORMS	59
4.3 Архітектура ПЗ.....	60
4.3.1 Програмне забезпечення для взаємодії з користувачем	60
4.3.2 База даних прогнозів захворювань	63
4.4 Інструкція користувача	64

	7
5. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДУ	67
5.1 Ключові критерії оцінки ефективності роботи методу.....	67
5.2 Оцінка ефективності роботи методу.....	70
6. ЕКОНОМІЧНА ЧАСТИНА	75
6.1 Проведення комерційного та технологічного аудиту науково-технічної розробки.....	75
6.2 Визначення рівня конкурентоспроможності розробки	79
6.3 Розрахунок витрат на проведення науково-дослідної роботи	82
6.3.1 Витрати на оплату праці	82
6.3.2 Відрахування на соціальні заходи	86
6.3.3 Сировина та матеріали	86
6.3.4 Розрахунок витрат на комплектуючі.....	87
6.3.5 Спецустаткування для наукових (експериментальних) робіт	88
6.3.6 Програмне забезпечення для наукових (експериментальних) робіт	89
6.3.7 Амортизація обладнання, програмних засобів та приміщень	90
6.3.8 Паливо та енергія для науково-виробничих цілей	92
6.3.9 Службові відрядження	93
6.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	94
6.3.11 Інші витрати.....	94
6.3.12 Накладні (загальновиробничі) витрати	94
ВИСНОВКИ	101
ДОДАТОК А.Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	108
ДОДАТОК Б.Технічне завдання	109
ДОДАТОК В.Фрагмент лістингу коду.....	112
ДОДАТОК Г.Ілюстративна частина	135

ВСТУП

За статистичними даними серцево-судинні захворювання є основною причиною смертності людей працездатного віку. Основними факторами ризику цього захворювання є неправильне харчування, недостатня фізична активність, куріння, зловживання алкоголем або наркотиками. Групи високого ризику потребують допомоги через раннє виявлення, консультування та лікування

Дуже важливий процес своєчасної діагностики проблеми пацієнта. У сучасній клінічній практиці електрокардіографічні методи широко застосовуються в діагностиці серцево-судинних захворювань. Електрокардіограма (далі - ЕКГ) - це зручна, дешева, безболісна і абсолютно нешкідлива процедура збору даних про електричні поля, створювані серцем. По суті, це графічне зображення різниці потенціалів. Як правило, ЕКГ подаються в програму у вигляді так званого часового ряду. Часовий ряд — це статистичний матеріал про будь-яке значення параметра (у найпростішому випадку одне), зібраний у різні моменти часу.

Сучасне програмне забезпечення повинно подавати звуковий сигнал, якщо пацієнт потрапив у небезпечну ситуацію. Часто патологічні зміни в серці швидко відображаються в потоці інформації, задовго до появи видимих симптомів захворювання. Люди називають такі прояви варіації аномаліями - інтервалами в часовому ряді, де поведінка синусоїди не відповідає нормальній поведінці. Важливість виявлення цих змін полягає в тому, що виявлення найменших аномалій у даних може призвести до значних порушень у прикладній області, описаній часовими рядами виявлення.

В даний час розробки в області дослідження електрокардіограми програмними методами вважаються перспективними, і багато вчених намагаються знайти ефективний спосіб автоматизації процесу виявлення відхилень. Однак існуюче програмне забезпечення часто дає хибні спрацьовування, має низьку точність прогнозування та застосовне до вузького кола проблем. Це відбувається через високий рівень шуму сигналу, слабкі

місця в алгоритмах обробки та недостатню базу для симптомів серцевої аномалії.

Тобто якість аналізу та класифікації за допомогою такого програмного забезпечення в реальних умовах може бути незадовільною. Ось чому пошук успішної обробки ЕКГ-сигналу та алгоритмів його аналізу став головним пріоритетом у розробці кардіологічної терапії

Метою даної роботи є розробка математичного забезпечення аналізу ЕКГ-сигналів та вибір найкращого алгоритму машинного навчання для класифікації цих даних. Інтеграція розроблених алгоритмів у відповідне програмне забезпечення дозволить покращити ідентифікацію захворювань пацієнтів на основі ЕКГ. Точність алгоритму і універсальність.

Для досягнення поставлених цілей необхідно вирішити складні проблеми

Наступні взаємопов'язані завдання:

- Вивчити та порівняти наявні алгоритми аналізу та розділення часових рядів ЕКГ та набору параметрів;
- Вивчити та порівняти існуючі методи класифікації часових рядів ЕКГ у ситуаціях захворювання;
- Створити свій власний алгоритм поділу, аналізу та класифікації часових рядів ЕКГ, який працюватиме з прийнятною точністю та охоплюватиме різні аномалії синусоїд;
- Реалізувати запропонований алгоритм у вигляді автономної бібліотеки;
- Розробити програмне забезпечення для визначення захворювання у пацієнтів за електрокардіограмою;
- Дослідження та розробка алгоритмів для підвищення ефективності роботи.

Об'єкт дослідження. Процес аналізу ЕКГ, поділ ознак часового ряду та класифікація на основі цих ознак.

Предмет дослідження: Алгоритми та методи аналізу та класифікації часових рядів ЕКГ на основі машинного навчання.

Наукова новизна отриманих результатів полягає у використанні штучного інтелекту для створення нового методу аналізу та класифікації синусоїдальних часових рядів на прикладі даних електрокардіограми з більшою ефективністю.

Практичне значення отриманих результатів полягає в застосуванні розробленої методики до аналізу даних електрокардіограми.

1. ВИЯВЛЕННЯ ІНФОРМАЦІЙНИХ ПОТРЕБ ПРИКЛАДНОЇ ОБЛАСТІ

Список захворювань, які можна виявити при дослідженні електрокардіограми, вражає:

- аритмія;
- Міокардит;
- Вроджена вада серця;
- Миготлива аритмія;
- Тахікардія;
- серцева недостатність;
- інфаркт.

Тому, незважаючи на те, що електрокардіографічне дослідження є одним з найдавніших методів діагностики, воно все ще займає лідируюче місце серед використовуваних методів діагностики. Розуміння основ ЕКГ та вміння оцінювати зубці, сегменти та інтервали ЕКГ необхідні для швидкої діагностики захворювання.

ЕКГ грає в діагностиці допоміжну роль, і її результати можуть бути інтерпретовані тільки в комплексі з клінічними проявами захворювання і даними інших методів дослідження. Тому програмне забезпечення, яке може поставити попередній діагноз пацієнту, стане проміжною ланкою між пацієнтом і реальним лікарем, що значно зменшить кількість необхідних досліджень і зменшить черги в медичних установах.

Не менш важливо, що програмна реалізація аналізу даних ЕКГ може слугувати додатковим страховим фактором у діагностиці. Таким чином, ймовірність помилок через людський фактор буде мінімізована.

1.1 Опис даних електрокардіограм

Дані зчитуються з пацієнта за допомогою спеціального приладу (електрокардіографа). Пацієнт повинен прийняти зручне положення, лежачи

плазом на воді з трохи зігнутими кінцівками. Для запису електрокардіограми кабелі пацієнта (її розгалужені частини) з'єднують з кінцівками шляхом розміщення металевих пластин (електродів).[51]

Швидкість запису ЕКГ дуже важлива для аналізу довжини сегментів сигналу. Зазвичай 1 мм відповідає 0,02 секунди при швидкості запису 50 мм/секунду, а 1 мм відповідає 0,04 секунди при швидкості запису 25 мм/секунду.

Електрокардіографічний комплекс складається із сегментів, перегородок і зубців.[51] На рисунку 1.1 показано основні досліджувані сегменти серцевого ритму. Зазвичай основний аналіз стану пацієнта проводиться саме по зубцях в такій послідовності: зубець P, інтервал P-Q, комплекс QRS і його зубці, сегмент ST, зубець T і U. Між хвилями U і P також є лінія, яка має спеціальну назву: ізоелектрична лінія.

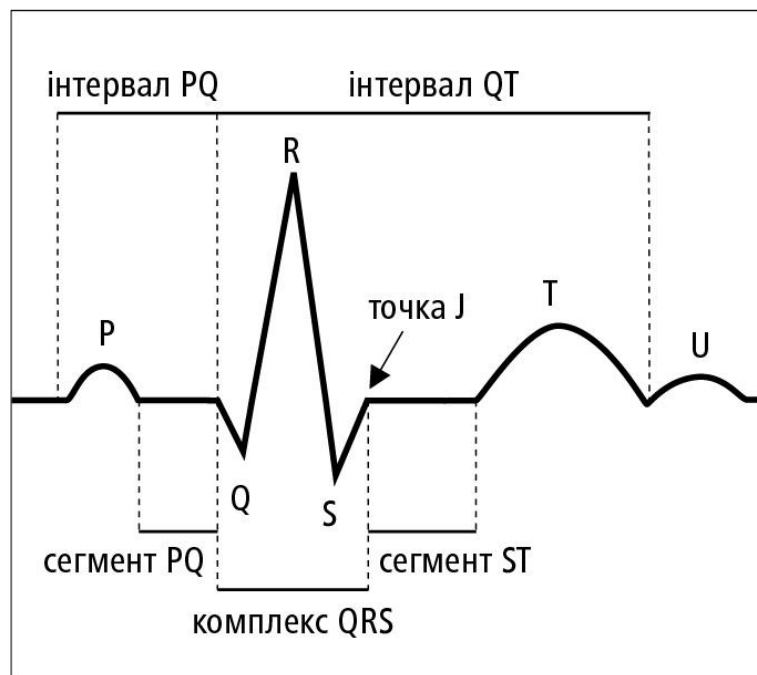


Рисунок 1.1 – Схема нормального запису ЕКГ сегменту

Основна увага приділяється вимірюванню часу інтервалів, форми зубців і співвідношення їх амплітуд, аналізу морфології шлуночків. Нижче наведені

основні стандартні принципи вимірювання [1], згідно з якими в подальшому аналізується серцевий ритм.

Пульс.

Знайдемо середню відстань між сусідніми піками RR (за потреби можна взяти інший розпізнаний пік). Якби точки ЕКГ записувалися зі швидкістю 50 мм/с, відстань між двома сусідніми точками на графіку становила б 0,002 секунди. Далі ділимо отримане число на 60 секунд, щоб отримати ЧСС за 1 хвилину.

При розрахунку цього показника необхідно враховувати відхилення в роботі серця. Тому, крім розрахунку «правильної» частоти, необхідно враховувати також ситуаційні частоти. Тому вам потрібно обчислити середню кількість інтервалів QRS, які виникають під час 6-секундного вимірювання, і помножити результат на 10.

Регулярність серцевого ритму.

Для здорових людей інтервал RR однаковий, але допускається відхилення до 10% в обидві сторони.

Провідність.

Робота спеціальної провідної системи дозволяє джерелу стимуляції швидко передаватись по серцю. Неправильна робота може бути описана трьома типами блокади: синусова блокада (періодична недостатність інтервалу P-T), внутрішньопередсердна блокада (подовження зубця P) і атріовентрикулярна блокада (подовження інтервалу P-Q під час недостатності Q-T).

Зубець P

Відповідає за скорочення лівого і правого передсердь. Виміряйте тривалість хвилі та її амплітуду. В ідеалі ці характеристики не повинні перевищувати 0,12 с і 0,2 мВ відповідно.

Зубець Q

Показує початок поширення імпульсу по [10] міжшлуночковій перегородці. Це початкове негативне відхилення QRS і не повинно

перевищувати однієї чверті висоти піку R або тривати довше 0,04 секунди. Нормальні зубці Q не повинні бути нерівними.

Зубці P

Відповідає розподілу пульсу міокарда лівого і правого шлуночків. Позитивне відхилення зубця QRS, викликане збудженням міокарда лівого і правого шлуночків, має бути чітко вираженим. Висота менше 24 мм вважається нормальною.

Зубці S

Кінець, що відображає поширення збудження по міжшлуночкової перегородці. Це друге негативне відхилення QRS, викликане збудженням в основі лівого шлуночка, і воно не повинно перевищувати однієї чверті висоти піку R або тривати довше 0,03 секунди. Це найглибша негативна хвиля на ЕКГ.

Зубці T

Відзначається реполяризація міокарда[10] шлуночків (підйом шлуночкового комплексу сегмента ST). Швидка кінцева реполяризація шлуночків від епікарда до ендокарда не повинна перевищувати однієї чверті висоти піку R, але може мати більшу тривалість до 0,24 с.

Інтервал P-Q

Показує тривалість імпульсу, що повертається в шлуночок через шлуночок, пучок Гіса і атріовентрикулярний вузол. Вимірюється від початку P до початку

Q. Суть цього інтервалу: час затримки відображуваного імпульсу. Інтервал PQ подовжується з віком. Зазвичай цей часовий інтервал подовжується на 0,12-0,18 с.

Інтервал Q – S

Вимірюється, починаючи з Q і закінчуючи S. Суть цього інтервалу: відображення деполяризації шлуночків. Тривалість може коливатися від 0,06 до 0,1 секунди.

Інтервал S – T

Вимірюється від кінця П. до початку Т. Це дуже важливо для виявлення аномалій.

Інтервал Q – Т

Суть цього інтервалу: відображення збудження всіх шлуночків серця.

Довжина не повинна перевищувати половини відстані R – R.

Індекс Маркоса

Відношення довжини зубця Р до інтервалу Р-Q. Значення для здорових людей становить 1,1 - 1,6.

1.2 Норми показників ЕКГ у людей в таблицях та критерії аналізу стану

Статеві відмінності в результатах ЕКГ відсутні.

Частота пульсу в стані спокою вважається нормальною, якщо вона коливається від 60 до 90 ударів.

Нормальні параметри часових інтервалів пацієнта мають приблизно такі значення [2]:

Таблиця 1.1 – Нормальні параметри часових інтервалів

Інтервал/пік	Довжина
PQ	0,12 – 0,2 с
QRS	0,06– 0,1 с
QT	<30% від RR 0,35 – 0,44 с
RR	0,6 – 0,66 с

Нормальними піковими показаннями для пацієнтів є приблизно такі значення, наведені в таблиці 1.2:

Таблиця 1.2 – Нормальні параметри часових інтервалів

Пік	Довжина	Амплітуда	Додатково
P	0,07–0,12 с	<2,5 мм	В деяких випадках може бути від'ємним.
Q	<0,03 с	<4мм, < 25% R	
R	0,06 - 0,1 с	6 – 25 мм	
S	<0,06 с	2,5 –6 мм	Амплітуда зазвичай мала
T	0,16 – 0,24 с	2 – 6 мм	Завжди додатній

Багато небезпечних захворювань серця можна виявити за допомогою електрокардіограми.

Аритмія - це патологічне захворювання, що характеризується порушенням частоти, ритму і послідовності серцевих скорочень. Виявляється по частоті і регулярності серцевого ритму.[7]

Нормальні показники:

ЧСС у спокої становить 60-90 ударів. Патологічні показники:

Тахікардія: частота 140-200 ударів на хвилину, стеноз комплексу Q-S.

Брадикардія: частота менше 60 ударів на хвилину, зберігаючи правильний ритм, і зубець P часто накладається на зубець T. Комплекс Q-S подовжується до 0,12-0,16 мілісекунд.

Фібриляція передсердь[10]: частота від 90 до 140 ударів на хвилину, відсутність зубців P, нерівні та непередбачувані інтервали R-R, однакові Q-S.

Інфаркт міокарда - це патологічне захворювання, при якому внаслідок недостатнього кровообігу в міокарді виникає запальний процес, що призводить до некрозу тканин. Це фактично смерть клітин серця і може призвести до інвалідності або навіть смерті.

Патологічні показники:

Істотні зміни інтервалу S-T і двох і більше послідовних інтервалів зубців T свідчать про виникнення некрозу тканин. Характерним показником може бути підйом сегмента S-T більше 2 мм.

Патологія зубця Q, що виникає через 2 або більше інтервалів поспіль. Тривалість на 0,03 секунди більша за норму вважається патологічною.

Метаболічна кардіоміопатія - це різновид ураження серця без запального процесу і обумовлена порушенням обміну речовин в організмі.

Патологічні показники:

Зміни інтервалу S-T, що проявляється підвищенням зубців T. Зубці T можуть бути сильно деформованими, негативними та низькою величиною.

Інтервал Q-T подовжується, а амплітуда Q-S загалом збільшується. Гіпертрофія шлуночків є основною причиною смерті від хвороб серця.

Патологічні показники:

Лівий шлуночок: пікові амплітуди S і R збільшуються, зубець T негативний або негативний і позитивний, а інтервал S-T знижується.[7]

Правий шлуночок: підвищена пікова амплітуда R, негативний зубець T і низька депресія інтервалу S-T.

Стенокардія – це синдром, який викликає біль і дискомфорт у серці, лівій руці та в області грудини. Це викликано гіпоксією міокарда.

Патологічні показники:

Депресія інтервалу S-T є найпоширенішим симптомом. Іноді S – T збільшується відносно інших інтервалів.

Висота піку R зменшується.

Перикардит - це синдром, [51]що призводить до захворювання перикарда та тампонади серця, скупчення рідини в серцевих камерах. Причина – запалення.

Патологічні показники:

Сегмент S-T підвищений, іноді зубець T інвертований, а сегмент P-Q пригнічений.

Змін зубців R і Q порівняно з ЕКГ здорової людини немає.

Міокардит - це запалення, яке вражає клітини серцевого м'яза, кровоносні судини і тканини перикарда. Патологічні показники:

Чітко помітні зміни спостерігалися не в одному інтервалі P-T, а майже в кожному інтервалі P-T. У більшості випадків інтервал S-T і T пік можуть змінюватися.

У порівнянні з ЕКГ у здорової людини зубець Q зазвичай залишається незмінним.

Тому в цьому розділі описано представлення даних ЕКГ. Шляхом поділу діаграми на частини були визначені основні орієнтири, інтервали та характеристики, на основі яких можна було б проводити подальші електрокардіографічні дослідження різних захворювань серця. Детальний опис допустимих меж характеристик діаграм може створити умови для вирішення питання «Чи хворіє?»

Також описані захворювання, які можна визначити за допомогою електрокардіографії. Визначено їх характерні кількісні ознаки, які можуть бути використані як орієнтир для машинних алгоритмів судження про стан хворого. У таблиці нижче серцеві захворювання згруповані за симптомами, видимими на електрокардіограмі.

Таблиця 1.3 – Патологічні параметри ЕКГ

Параметр	Характеристика	Вірогідні захворювання
R - R	Нерівномірна відстань	миготлива аритмія серцева блокада
R пік	>25 мм >0,1 с Подвійний пік Відсутність Пилкообразність	потовщення міокарда передсердь мерехтіння передсердь
P – Q інтервал	>0.2 с <0.03с	атріовентрикулярна блокада серця
Q–S інтервал	Вид флага Відсутність горизонтальної лінії	гіпертрофія міокарда шлуночків блокада ніжок пучка Гіса фібриляція шлуночків інфаркт міокарда
Q пік	>4 мм >0,03 с	інфаркт міокарда
S пік	> 10 мм	Гіпертрофія лівого шлуночка.
S–T інтервал	>6 мм	стенокардія інфаркт міокарда ішемічна хвороба
T пік	>50% R Гострий пік Подвійний пік	перевантаження серця ішемічна хвороба інфаркт міокарда

2. ВІДОМІ МЕТОДИ КЛАСИФІКАЦІЇ ТА АНАЛІЗУ ДАНИХ

Вивчення даних ЕКГ прийнято ділити на 2 основних етапи:

- Аналіз поверхні графіки, вибір опорних точок, інтервалів і особливостей;
- Виконайте поглиблений аналіз на основі отриманих даних, порівняйте з типовими даними в базі даних і зробіть висновки.

Протягом останнього десятиліття дослідники так чи інакше намагалися винайти алгоритм, який міг би точно визначити хворобу людини на основі електрокардіограми. Більшість із них призначені для задоволення потреби у точній діагностиці захворювань серця без втручання людини. У цьому розділі описані найпопулярніші з цих методів.

2.1 Методи виділення опорних точок

Перший етап є вирішальним для точності цього аналізу. Щоб успішно знайти всі опорні точки на початку, потрібно призначити R-інтервали. Те, що апарат безпомилково виявив зубець R, було надзвичайним успіхом.

По-перше, різні шуми можуть викликати спотворення ЕКГ:

- Міоелектричні перешкоди – високочастотні перешкоди, пов'язані з м'язовою діяльністю пацієнта;
- Низькочастотний дрейф контурів через поганий контакт серцевого підсилювача між електродами

Недостатня обробка шкіри людини або її поверхні;

Вказівки по мережі живлення.

По-друге, існує великий ступінь варіабельності форми, частоти, амплітуди та часових характеристик піку R, що зумовлено фізіологічними особливостями пацієнта або патологією серцево-судинної системи.

Метод локальних мінімумів і максимумів [3] є простою спробою вирішити цю проблему. Оскільки ми точно знаємо, чи має бути опорна точка позитивною чи негативною, ми можемо шукати Rpeaks, а потім безперервно

рухатися в обох напрямках, до локального мінімуму та максимуму. Детальний алгоритм пошуку показано на рисунку 2.1-2.2.

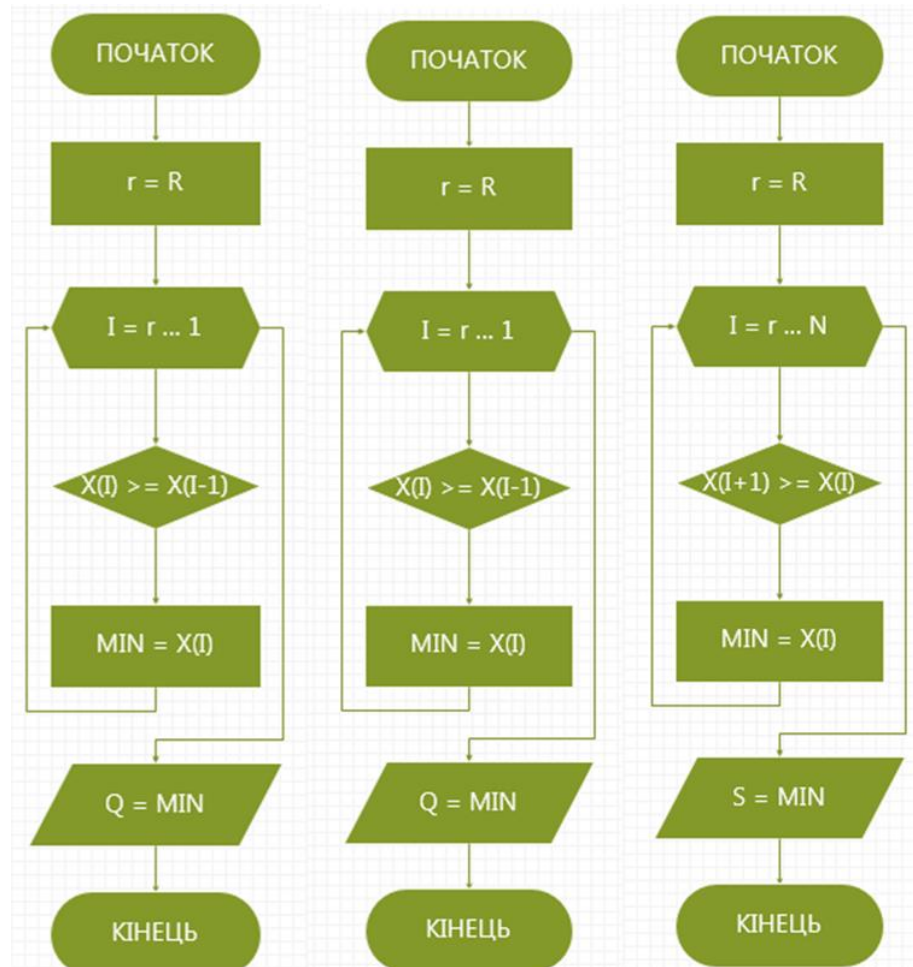


Рисунок 2.1 – Алгоритми пошуку опорних точок Q, R, S

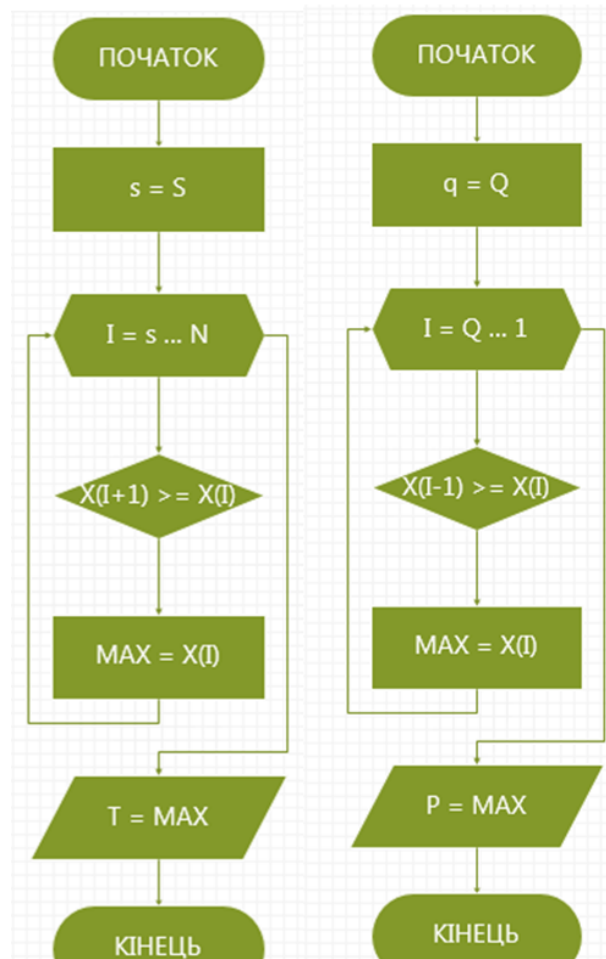


Рисунок 2.2 – Алгоритми пошуку опорних точок P, T

Переваги цього методу:

- Простий в реалізації;
- Не потребує багато ресурсів;
- Хороша здатність розпізнавання електрокардіограми здорових людей.

Недоліки цього методу:

- Нездатність ідентифікувати патологічну електрокардіограму пацієнта;
- Чутливий до перешкод даних.

Алгоритм Пан-Томпкінса [4] базується на аналізі нахилу та ширини в межах Q-Sinterval і складається з наступної послідовності фільтрів і методів:

- фільтр низьких частот;
- Фільтр високих частот;
- Знайти похідні;
- Майдан;

- Інтеграція;
- Обмежте процес пошуку.

Фільтр низьких частот описується рівнянням 2.1:

Фільтр високих частот описується рівнянням 2.2:

Похідна функція описується рівнянням 2.3:

Подальше зведення в квадрат робить результат позитивним, посилюючи компоненти комплексу, що призводить до нової електрокардіограми, на якій посилюються високочастотні піки Q, R і S, а низькочастотні хвилі P і T посилюються. гладкий.

Інтегральна функція (фільтр типу ковзного вікна) описується такою формулою:

$$y(n) = 1/N * [x(n - (N - 1)) + x(n - (N - 2)) + \dots + x(n)] \quad (2.3)$$

де N - ширина вікна.

Рисунок 2.3 показує результати цього підходу:



Рисунок 2.3 – Метод Пана–Томпкінса

Переваги цього методу:

- Простий в реалізації;

- Добре показує контрольні точки Q, R і S на ЕКГ. Недоліки цього методу:

- Покращені асиметричні Т-піки можна записати як R-піки.
- Щоб визначити пікове значення вихідного сигналу, потрібно врахувати затримку, викликану фільтром.

Завдяки простоті виявлення та підрахунку переходів через нуль алгоритм розрахунку зміни знака [5] є ефективним вирішенням проблеми виявлення комплексів QRS.

На першому етапі сигнал фільтрується через смуговий фільтр з нижньою і верхньою частотами зрізу 18 Гц і 35 Гц відповідно.

Далі виконуємо розрахунок за формулою 2.4:

$$y(n) = \text{sign}(f(n))f(n ** 2) \quad (2.4)$$

Оцінка амплітуди розраховується згідно з рівнянням 2.5:

$$y(n) = \lambda * K(n - 1) + (1 - \lambda * K) * |y(n)| * c \quad (2.5)$$

Серед них $\lambda K \in (0; 1)$ є пропущеним множником, а параметр c представляє константу підсилення, наприклад, $c = 4$.

Перетворить сигнал на більш високі частоти відповідно до рівняння 2.6:

$$z(n) = y(n) + b(n) \quad (2.6)$$

Рівняння 2.7 використовується для обчислення кількості перехресть:

$$d(n) = \left[\text{sign}[z(n)] - \frac{\text{sign}[z(n - 1)]}{2} \right] \quad (2.7)$$

Виявлення події виконується за допомогою адаптивного порогу θ , розрахованого відповідно до рівняння 2.8:

$$\theta(n) = \lambda * \theta(n - 1) + (1 - \lambda) * d(n), \text{ де } \lambda \theta \in (0; 1) \quad (2.7)$$

Рисунок 2.4 показує результати цього підходу:



Рисунок 2.4 – Метод підрахунку зміни знака

2.2 показано результати цього методу Переваги цього методу:

- Простий в реалізації;
- Добре показує контрольні точки Q, R і S на ЕКГ. Недоліки цього методу:
- Щоб визначити положення піку вихідного сигналу, необхідно врахувати затримку смугового фільтра.

Алгоритм безперервного вейвлет-перетворення [6] є новою технологією ЕКГ, яка забезпечує вдосконалені методи. Перевагою такого перетворення є можливість запису деталей ЕКГ-сигналу в оптимальний час.

Вейвлет-перетворення базується на наборі функцій аналізатора, які розкладають сигнал ЕКГ на послідовність коефіцієнтів. Цей тип функціонального аналізатора (wavelet) є загальною назвою сімейства математичних функцій, які є локальними за часом і частотою і в яких усі функції є похідними від базової (генеруючої) функції за допомогою її зсувів і розтягуються в часі. осі, кожен вейвлет має певну тривалість, часову позицію та діапазон частот. Результатом вейвлет-перетворення є те, що коефіцієнти відповідають компонентам ЕКГ в межах певного періоду часу та діапазону частот.

Вони діляться на дві основні групи:

- Дискретні - використовуються для обробки і стиснення сигналу та його кодування;

- Безперервний - використовується для аналізу сигналів у наукових дослідженнях.

Вейвлет Морле (рис. 2.5):

існує:

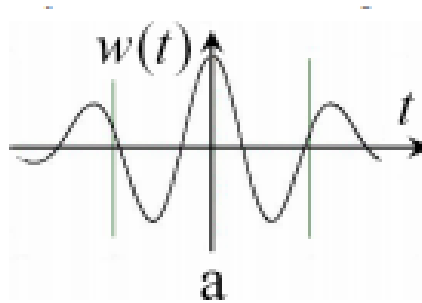


Рисунок 2.5 – Вейвлет Морле з масштабуванням 0.2

Вейвлет Хаара (рис. 2.6):

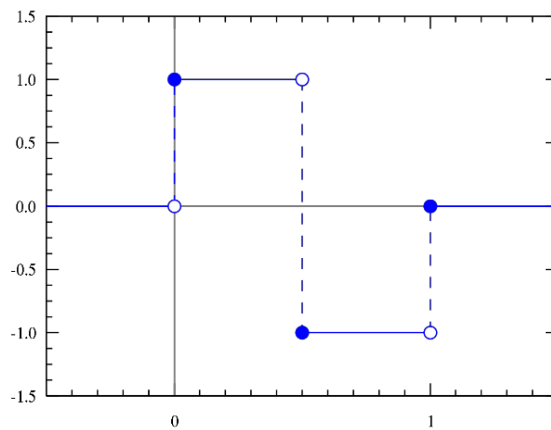


Рисунок 2.6 – Вейвлет Хаара

Вейвлет Гауса (рис. 2.7):



Рисунок 2.7 – Вейвлети Гаусса різних порядків

Вейвлет «Мексиканський капелюх» (рис. 2.8):

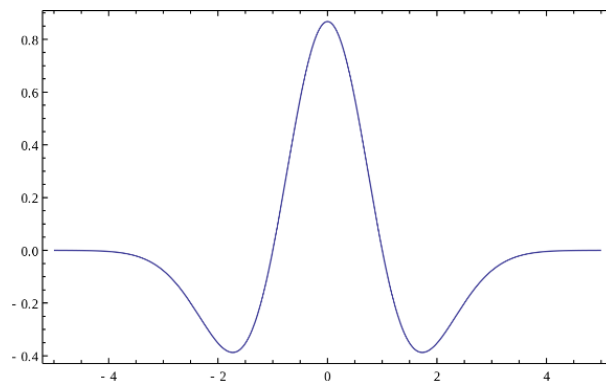


Рисунок 2.8 – Вейвлет MEXICANHAT

Переваги цього методу:

- Сигнали добре ранжовані як за часом, так і за частотою, з яких можна вибрати лише необхідний рівень деталізації;
- Варіативність функцій забезпечує широкий спектр використання методу. Недоліки цього методу:
- Складність алгоритму.

2.3 Різноманітні підходи до глибинного дослідження числового ряду

Справжнє виявлення аномалій у часових рядах залишається складним завданням. Це особливо важливо для періодичних або квазіперіодичних часових рядів. Часові ряди ЕКГ є яскравими прикладами квазіперіодичних сигналів у реальному світі. Протягом багатьох років люди намагалися знайти найкращий алгоритм виявлення порушень ЕКГ за допомогою машин, без втручання людини.

2.3.1 Прогнозування часових рядів

У 2019 році вони винайшли метод класифікації сигналів ЕКГ за допомогою нейронних мереж [7]. У статті запропоновано алгоритм на основі нейронної мережі довготривалої короткочасної пам'яті (LSTM).

Ідея полягає в тому, щоб навчити модель передбачати наступні інтервали часових рядів. Інтуїція, яка лежить в основі цього підходу, полягає в тому, що регулярні квазіперіодичні шаблони в часових рядах ЕКГ повинні бути передбачуваними (допускати лише невеликі помилки), тоді як аномальна поведінка повинна призводити до великих відхилень у прогнозах.

Задача прогнозування вивчається з використанням 2-рівневої архітектури LSTM. Кожен поверх складається з 64 одиниць. Модель LSTM навчається за допомогою визначення Adamoptimizer (алгоритм середньоквадратичної помилки MSE). Інші функції, такі як log-cosh (логарифм гіперболічного косинуса) і MAE (середня абсолютна похибка), також враховуються.

Відстань Махаланобіса використовується для оцінки аномалій у часових рядах. Більші помилки в одному чи кількох вимірах призведуть до більших значень відстані Махаланобіса. Тому відстань Махаланобіса можна використовувати як індикатор ненормальної поведінки сигналів часового ряду.

На малюнку 2.9 чітко показано часову відстань Махаланобіса вибраного сигналу ЕКГ до та після віконного алгоритму корекції помилок.

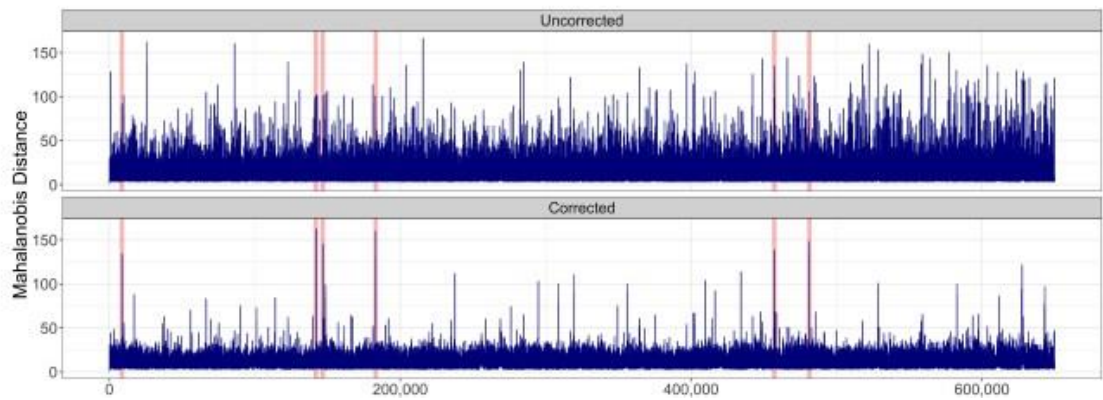


Рисунок 2.9 – Відстань махланобіса

Вивчаючи аномалії, необхідно перевірити дві гіпотези:

- H_0 : у даних немає відхилень.
- H_1 : у даних є до $v\%$ аномалій.

Алгоритм покрокового аналізу наведено нижче.

- 1) Вважаємо, що кожне $k \in \{1, \dots, m\}$ має наближений нормальний розподіл.
- 2) Ітеративно обчислити багатовимірний нормальний розподіл середнього μ та коваріаційної матриці Σ .
- 3) Ми обчислюємо різницю спостережуваного вектора порівняно з цим основним багатofакторним нормальним розподілом шляхом обчислення відстані Махаланобіса
- 4) Вибираємо максимальне значення відстані Махаланобіса M_j і оцінюємо, чи перевищує це значення квантиль $1 - \alpha$ нормального розподілу.
- 5) Якщо так, то спостереження вважається аномалією.
- 6) Програма зупиняється, коли щонайбільше $v\%$ даних позначено як аномалії

результат:

Подібно до традиційних завдань класифікації, похибка результатів в основному вимірюється за допомогою чотирьох показників: кількість істинно позитивних результатів (TP), кількість помилкових позитивних результатів (FP), кількість помилкових негативних результатів і кількість справжніх

негативних результатів (TN). У результаті ми отримуємо справжню позитивну метрику TPR та справжню негативну метрику FPR.

Точність цього алгоритму становить 83%.

2.3.2 Класифікатор за піками та їх протяжністю

У статті [8] запропоновано алгоритм розпізнавання електрокардіограми, заснований на виявленні та позиціонуванні в часі максимального значення модуля вейвлет-перетворення та класифікації серцевих циклів за допомогою нейронних мереж.

Завдання класифікації аеробних велосипедів полягає в тому, щоб підібрати аеробний велосипед до однієї з чотирьох категорій:

- відсутність патології;
- Заблокувати ліву ніжку його пакета;
- Заблокуйте праву ніжку свого пакета;
- Передчасні скорочення шлуночків.

Вхідний термінал першого шару нейронів вводить 24 елементи ознак (див. табл. 2.1):

Таблиця 2.1 – Ознаки класифікації кардіоциклів

Номер	Ознака
1	Тривалість QRS-комплексу (QRS)
2	Тривалість інтервалу PQ (PQ)
3	Тривалість інтервалу QT (QT)
4	Тривалість попереднього RR-інтервалу
5	Тривалість наступного RR-інтервалу
6 - 15	Нормалізовані морфологічні ознаки сигналу ЕКГ між початком і кінцем QRS (піки та початок – кінець)
16 - 24	Нормалізовані морфологічні ознаки сигналу ЕКГ між кінцем QRS-комплекса і кінців T-хвилі (піки та початок – кінець)

Виходи нейронів першого шару надходять на входи нейронів другого шару, а виходи нейронів другого шару утворюють вихідний вектор мережі (див. рис. 2.10).

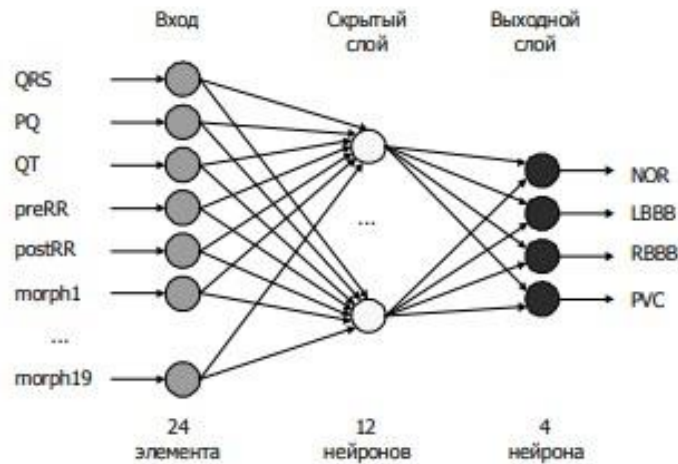


Рисунок 2.10 – Схема нейронної мережі

Дослідження проводилося на цифрових записах сигналів електрокардіограми з PhysioBank, постійно оновлюваної бази даних складних фізіологічних сигналів. База даних QTDB містить 105 записів ЕКГ по 15 хвилин на відведення з частотою дискретизації 250 Гц. Кожен запис містить щонайменше 30 ударів і анотований початком зубця Р і комплексу QRS, піком і кінцевим інтелектом, а також маркерами піку і кінця зубця Т.

результат:

Для оцінки якості роботи використовуються дві метрики: чутливість (здатність алгоритму видавати коректні результати) і передбачуваність позитивних результатів (positive Predictability).

Він може досягти справжнього позитивного співвідношення 99,7% усіх позитивних випадків.

Подібне дослідження було проведено в статті [9].

Для створення початкових даних використовувався структурований масив цифрових записів реальних фізіологічних сигналів і пов'язаних із ними

даних, які використовуються в дослідженнях біомедичної спільноти. Ці записи були отримані за допомогою холтерівського моніторингу в умовах стаціонару.

Алгоритм зворотного поширення використовується для навчання штучних нейронних мереж. Було підготовлено набір даних із 458 подій ЕКГ (50% навчальних зразків, 30% перевірочних зразків, 20% тестових зразків). Підготовлені вхідні дані подаються на вхідний рівень нейронної мережі, а вихідний рівень нейронної мережі діагностує стан пацієнта.

Вхідні параметри мережі показані на малюнку 2.11:

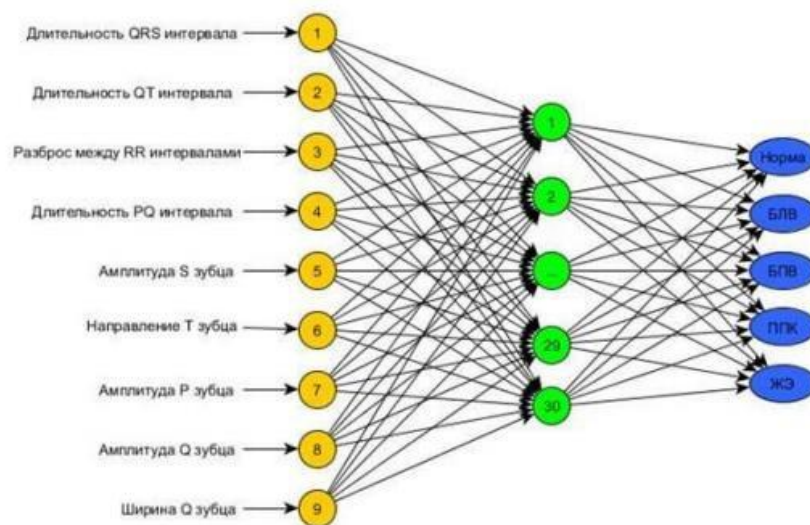


Рисунок 2.11 – Схема нейронної мережі

З метою знаходження оптимальної кількості нейронів у прихованому шарі нейронної мережі з багатшаровою структурою персептрона в роботі було оцінено критерії ефективності штучних нейронних мереж – чутливість, специфічність та похибка навчання.

В ідеалі показники чутливості і специфічності повинні прагнути до 100%. У реальних ситуаціях при вирішенні діагностичного завдання система повинна вибрати один з декількох варіантів діагностики. При цьому очікується, що критерії чутливості та специфічності системи для всіх варіантів діагностики матимуть рівномірно розподілені значення, але не нижче порогу, при якому результат не можна вважати достовірним.

результат:

Специфічність цієї системи була підтверджена як 81% у дослідженнях контрольної групи без серцево-судинних захворювань.

Чутливість розробленої системи була перевірена в контрольній групі з серцево-судинними захворюваннями і склала 79%.

2.3.3 Класифікатор за допомогою методу кластеризації

У статті [10] медіанний фільтр і фільтр низьких частот використовуються для покращення продуктивності алгоритму класифікації напруг. Після застосування медіанного фільтра використовуйте фільтр низьких частот, щоб видалити шум.

Характерні точки на основі інтервалу R-R і різниці між піковими значеннями R і S витягуються з отриманого сигналу ЕКГ. Для класифікації сигналу ЕКГ пікове значення R-S було встановлено на 1,2 як орієнтир. Якщо воно більше цього значення, сигнал ЕКГ зміщується вправо, якщо менше цього значення, то ліворуч.

Використовуючи кластеризацію K-means, він розділений на дві ситуації: тиск і відсутність тиску. Результати кластеризації показані на малюнку 2.12:

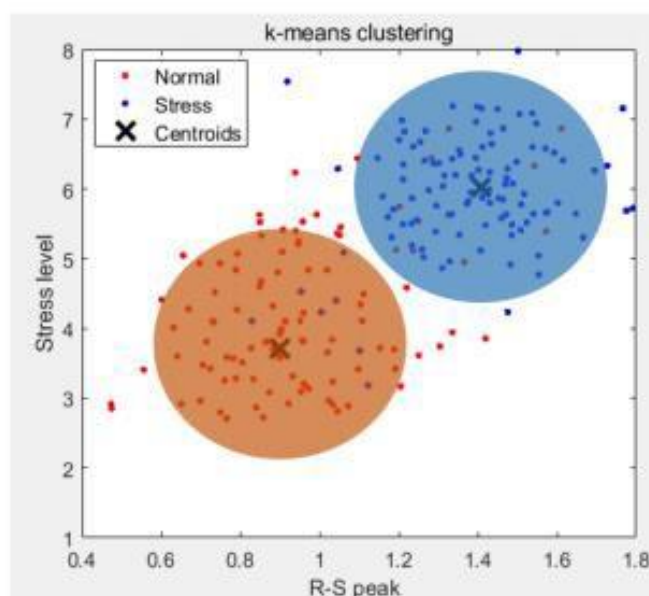


Рисунок 2.12 – Результат алгоритму K-means

Точність запропонованого в цьому дослідженні алгоритму класифікації тиску була розрахована шляхом порівняння всіх даних ЕКГ і відстаней до всіх центроїдів. Цей показник точності становить 85,07%.

У статті [9] автором запропоновано метод неконтрольованого дослідження електрокардіограми (Unsupervised representation learning using deep generative models).

Пропонований метод складається з двох завдань:

- Навчання алгоритмам на основі даних ЕКГ без супроводу;
- Подальші завдання з виявлення аномалій.

Модель, яка використовується для навчання, базується на варіаційному автокодері, параметризованому рекурентною нейронною мережею.

Вхідними даними для моделі є часові ряди.

Принцип виявлення аномалій на основі автокодерів полягає в тому, щоб навчити модель на часових рядах, які здебільшого мають нормальні моделі, щоб вона вивчала багато нормальних значень даних.

Мета полягає в тому, щоб ідентифікувати два кластери, що містять приховані коди нормальних і ненормальних серцевих скорочень на ЕКГ. Ця стратегія передбачає, що більшість серцебиття є нормальними, тому ненормальне серцебиття проектуватиметься в прихованому просторі інакше, ніж нормальне серцебиття.

Було розглянуто три алгоритми кластеризації:

- Ієрархічна кластеризація;
- Спектральна кластеризація;
- K означає.

Кластери призначаються до цих класів на основі їх розміру: кластер із більшою кількістю точок даних призначається до нормального класу, тоді як інший кластер призначається до ненормального класу.

результат:

AUC розраховується на основі кривої ROC із відповідними істинно позитивними та хибнопозитивними показниками.

Алгоритм має точність 92%.

2.2.4 Аналіз синусоїдального ритму за допомогою SVM

У статті [11] розглянуто алгоритми вибору та класифікації інформаційних сигнатур ЕКГ-сигналу.

Основна ідея дискретного вейвлет-перетворення (DVP) полягає в тому, щоб розкласти сигнал на дві складові - грубу (наближення) і уточнену (деталі), а потім розбити їх, щоб змінити рівень розкладання сигналу. У той же час сам сигнал розглядається як нульовий шар розкладання, а наступні шари утворюють низхідне дерево вейвлетів того чи іншого виду. При переході на більш низькі рівні точність подання сигналу знижується, але збільшується можливість вейвлет-фільтрації сигналу, видалення завад і ефективного стиснення сигналу.

Розглянемо задачу класифікації двох класів об'єктів, невизначених:

- багато навчальних об'єктів, представлених векторами ознак;
- Багато відповідей для навчальних об'єктів.

Рішення задачі класифікації зводиться до побудови такої функції (класифікатора), яка оптимально виділяє точки з навчальних вибірок різних класів.

У методі SVM гіперплощина вибирається як функція, де найближчі векторні точки (називаються опорними точками) двох класів є рівновіддаленими (див. рис. 2.13).

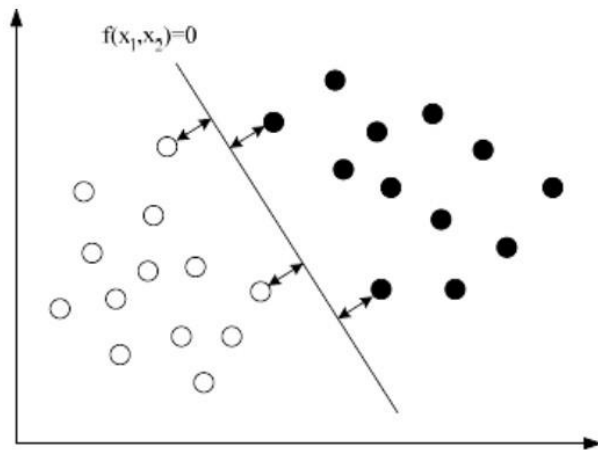


Рисунок 2.13 – SVM для двовимірного простору

На першому етапі з набору навчальних прикладів вибираються еталонні вектори і на їх основі будується площина сегментації. Фаза розпізнавання полягає в тому, що класифікатору вводять приклад, який нічого не знає про свою приналежність до класу. Класифікатор повинен дати відповідь, до якого класу належить вектор.

У цій статті в якості даних для аналізу обрано електрокардіограму нормального синусового ритму та електрокардіограму аритмії.

Частотно-часові параметри варіабельності серцевого ритму, група середньоквадратичного відхилення, детальні та наближені коефіцієнти, отримані за результатами дискретного вейвлет-розкладу, були використані як вихідні дані для вирішення задачі класифікації в цій роботі (див. табл. 2.2).

Таблиця 2.2 – Ознаки класифікації серцевого ритму

Номер	Ознака
1	Пульс
2	Відхилення пульсу
3	Коефіцієнт варіації
4	Потужність спектру TP
5	Потужність спектру LF
6	Потужність спектру HF
7	Потужність спектру VLF
5	Стрес-індекс

Після фази навчання на вхід надходить сигнал ЕКГ, але про категорію, до якої він належить, нічого не відомо. Результатом класифікації є віднесення сигналу до категорії сигналів ЕКГ зі здоровим ритмом, що є доказом того, що система функціонує правильно.

Точність цього алгоритму становить 84%.

Тому в цьому розділі представлені основні методи, які використовуються в даний час при дослідженні ЕКГ.

На першому етапі (визначення параметрів ЕКГ, опорних точок) найчастіше використовують два алгоритми:

- локальні мінімуми і максимуми;
- Вейвлет-перетворення.

Вони полегшують подальшу обробку сигналу, звільняючи його від шумів і виявляючи певні особливості, за якими два інтервали найбільш вигідно порівнюють один з одним.

Існує багато варіантів дослідження для фази II. У таблиці 2.3 наведено порівняльну характеристику найбільш популярних методів. Кожен з них має певні характеристики, які відрізняють його від інших. Однак жоден з них не є ідеальним, тому дослідження в цій галузі науки киплять. Зеленим кольором позначені ознаки, які суттєво впливають на позитивний результат, червоним — ознаки, що призводять до негативного.

Таблиця 2.3 – Ознаки класифікації серцевого ритму

	Прогнозування часових рядів	Класифікатор за піками	Кластеризація	Аналіз на основі SVM
Вхідні дані	Бази даних	Бази даних, вже з параметрами	Бази даних	Бази даних
Опорні точки	Попередні повні інтервали	Локальні мінімуми, максимуми та час	Локальні мінімуми, максимуми та інші опорні значення	Частотні параметри
Попередня обробка ряду	Ні	Ні	Так	Так
Головний алгоритм	LSTM, Melanobis distance	Нейронна мережа	K-means	SVM
Навчання	Без учителя	З учителем	Без учителя	З учителем
Складність програми	Складний	Складний	Простий	Простий
Класифікація	Однокласова	Багатокласова	Багатокласова	Однокласова
Опорний клас	Здорова ЕКГ	Кожна хвороба одночасно	Кожна хвороба одночасно	Здорова ЕКГ
Конкретний діагноз	Ні	Так	Так	Ні
Точність	83%	80%	92%	84%

Проаналізувавши таблицю, ви можете виділити вдалі особливості кожного алгоритму і звести їх до списку:

- Попередня обробка даних за допомогою вейвлетів і шумових фільтрів;
- ЕКГ, спрямована на виявлення захворювання, а не здоров'я;
- Простота програми дозволяє її подальше розширення;
- Найбільш детальні, відфільтровані та згруповані вхідні дані з різними значеннями для різних станів людини;
- Багатокатегорійна класифікація, результати - специфічна діагностика;
- Навчання з учителем.

3. РОЗРОБКА МЕТОДУ КЛАСИФІКАЦІЇ ТА АНАЛІЗУ ДАНИХ

Проаналізувавши існуючі алгоритми розпізнавання ЕКГ, врахувавши всі переваги та недоліки існуючих методів, було вирішено розробити власний алгоритм.

3.1 Постановка завдання

Мета: Основною метою магістерської роботи є підвищення точності та варіабельності аналізу даних ЕКГ захворювання шляхом використання інформаційно насиченого набору вхідних даних та створення алгоритму обробки, найбільш придатного для таких захворювань. аналізувати.

Мета: Створення програмного забезпечення, здатного самостійно вибирати ключові характеристики інтервалів ЕКГ, аналізувати їх больовий профіль та інформувати користувача про стан пацієнта.

завдання:

- Розробити алгоритм зчитування даних ЕКГ та розподілу їх на інтервали PQRST;
- Визначити ключові характеристики та провести відповідний аналіз даних;
- За допомогою створеного алгоритму досліджується електрокардіограма кожного захворювання окремо і ключові ознаки записуються в набір даних;
- Використовуйте машинне навчання для розробки алгоритмів аналізу ЕКГ для конкретних захворювань;
- Вивчення ефективності алгоритмів;
- Створити програмний інтерфейс для візуалізації результатів обробки ЕКГ пацієнта.

3.2 Метод модифікації вхідних даних

Набір даних у формі оригінальних часових рядів [12] був обраний як навчальні та тестові дані.

Алгоритм фільтрації даних в основному складається з двох кроків:

- Застосування вейвлет-перетворення;
- Виявляти періоди сигналу та їх несправності.

3.2.1 Вейвлет Хаара

Перша проблема, з якою програма має працювати з даними ЕКГ, це її розмір. У середньому приблизно 300 000 точок потрапляє в набір даних протягом 20 хвилин, необхідних для зняття ЕКГ.

За допомогою вейвлет-перетворення дані можна багаторазово стискати без втрати необхідної інформації. Вейвлет Хаара (див. рис. 3.1) є одним із перших і найпростіших вейвлетів [17].

Він базується на системі ортогональних функцій, запропонованій у 1909 році угорським математиком Альфредом Хара. Вейвлети Хаара ортогональні і добре розташовані в просторі, але не гладкі.

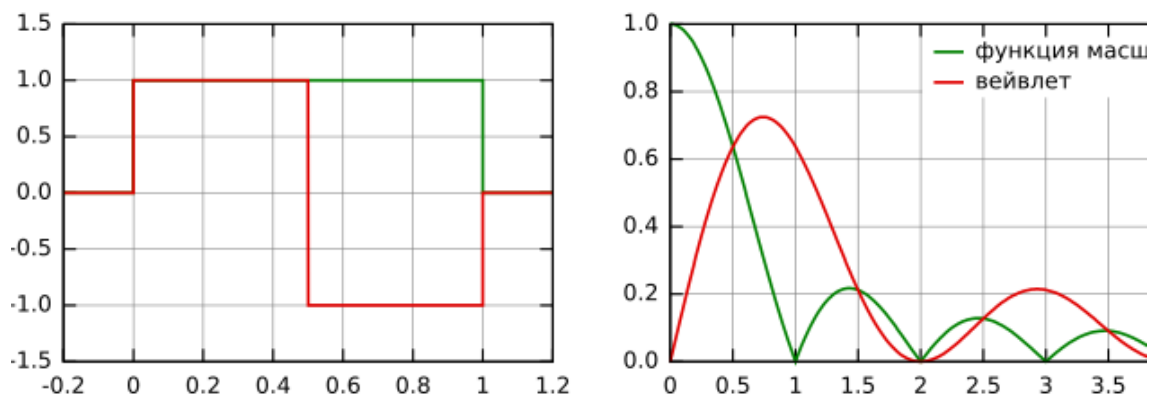


Рисунок 3.1 – Функція масштабування та її частотні складові

Функція визначається за формулою 3.1:

$$f(x) = \begin{cases} 1, & 0 < x < 1/2 \\ -1, & 1/2 < x < 1 \\ 0, & x > 1 \parallel x < 0 \end{cases} \quad (3.1)$$

Припустимо одновимірний дискретний вхідний сигнал S , кожній парі сусідніх елементів відповідають два числа (див. формулу 3.2)

$$a = \frac{s_{2i} + s_{2i+1}}{2} \quad (3.2)$$

$$b = \frac{s_{2i} - s_{2i+1}}{2} \quad (3.3)$$

Повторюючи цю операцію для кожного елемента вихідного сигналу, виводяться два сигнали, один з яких є приблизною версією вхідного сигналу, а другий містить деталі, необхідні для відновлення вихідного сигналу.

На кожному кроці перетворення сигнал розбивається на дві складові:

- наближення нижчої роздільної здатності (апроксимація);
- Інформація, яка дозволяє розробити приблизні значення.

Перетворення Хаара використовується для стиснення вхідного сигналу, стиснення зображення, переважно кольорового та чорно-білого, і згладжування переходів. Чудово підходить для рентгенівських зображень. Цей тип архівування відомий давно і походить безпосередньо з ідеї використання регіональної послідовності. Ступінь стиснення встановлюється від 5 до 100.

3.2.2 Алгоритм виявлення періоду

Після вейвлет-перетворення сигнал містить більш концентровані локальні мінімуми та максимуми. Алгоритм визначення періодичності сигналу базується на побудові R -інтервалів (див. рис. 3.2).

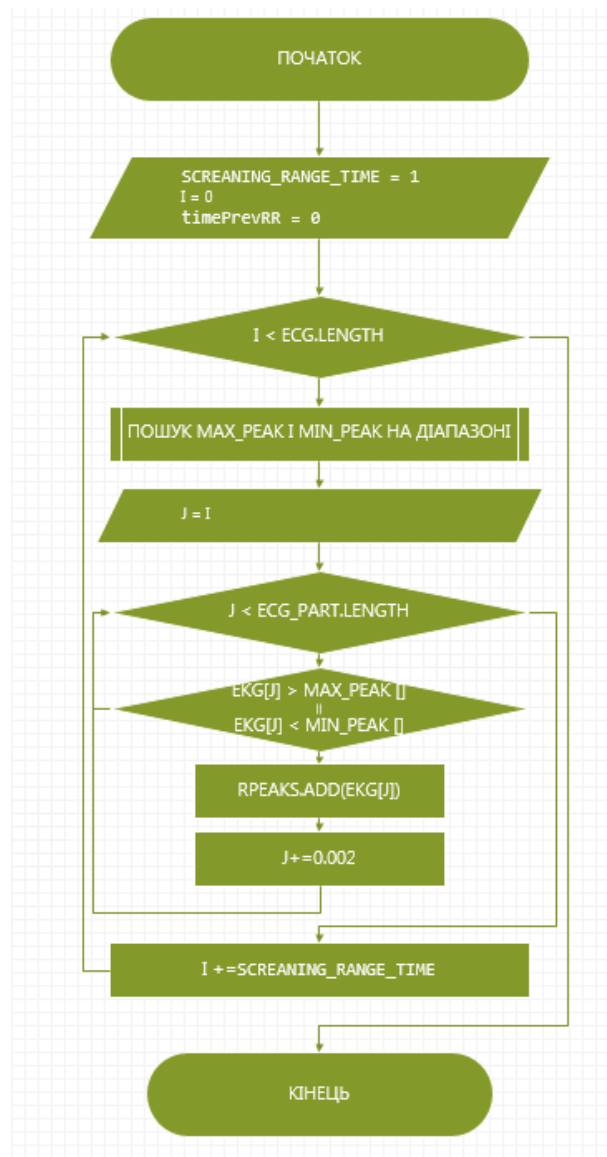


Рисунок 3.2 – Алгоритм поділу на R-Інтервали

Загальна довжина ЕКГ була розділена на інтервали по 500 точок (сподіваємося, щоб отримати 3-4 R-піки на точку). Це дозволяє програмі уникнути помилок у визначенні локальних мінімумів на різних інтервалах, коли є різниці висот і зміни поведінки кривої. Це також дозволяє виявляти сегменти всіх п'яти фаз у межах циклу.

Загалом сегменти ЕКГ можна розділити на здорові (див. рис. 3.3) і хворі (див. рис. 3.4). Як бачите, у деяких випадках наше загальне уявлення про комплекс PQRS може сильно відрізнятись від послідовності в реальному часі.

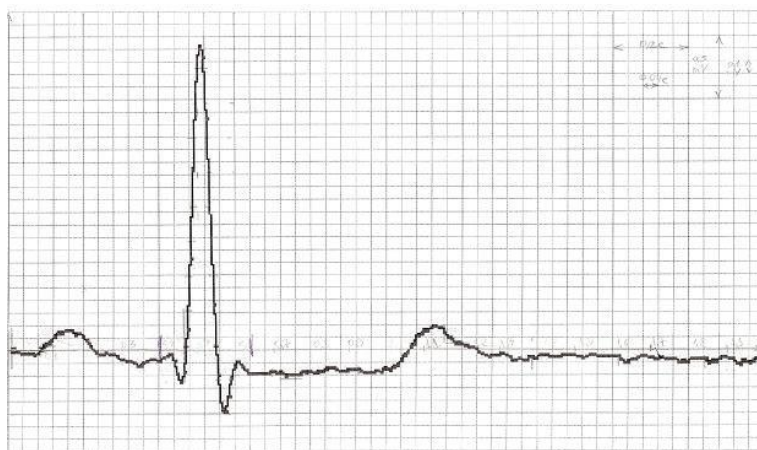


Рисунок 3.3 – Приклад QRS здорової людини



Рисунок 3.4 – Приклад викривлення QRS хворої людини

Такі зміни можуть призвести до неможливості визначити правильний сегмент R-R. Тому для цього випадку алгоритм тимчасово бере локальний мінімум у межах усього інтервалу як пік R.

3.3 Методіаналізу даних для проектування програмного забезпечення

У процесі пошуку орієнтирів, які формують набір даних з важливими характеристиками, ми проаналізували, які характеристики найчастіше перевіряють лікарі, досліджуючи електрокардіограми пацієнтів у лікарнях.

3.3.1 Методаналізу числових даних

У таблиці 3.1 наведено основні орієнтири, знайдені методом визначення локальних мінімумів і максимумів (див. главу 2).

Таблиця 3.1 – Основні точки та причини їх появи в алгоритмі

Точка	Причина
P локальний максимум	Вказує на блокаду серця, перикардит, міокардит
Q локальний мінімум	Вказує на інфаркт
R локальний максимум	Вказує на хворобу передсердь
S локальний мінімум	Вказує на аритмію, ішемічну хворобу
T локальний максимум	Вказує на аритмію, стенокардію

Крім того, про хворобливому стані може свідчити і часовий перебіг тієї чи іншої події під час серцебиття. Основні часові характеристики кожного інтервалу PQRST зібрані нижче:

- час P відносно режиму R;
- час Q відносно режиму R;
- час S відносно режиму R;
- час T відносно режиму R;
- інтервал часу PQ (від горба P до точки Q);
- Час QInterval (від початку відхилення Q до кінця відхилення S);
- Час STinterval (від точки S до кінця горба T);
- ЧСС (ударів в хвилину);
- Середнє відхилення ЧСС (аритмія).

3.4 наведено алгоритм розбиття кожної пари інтервалів R-R на числові показники:

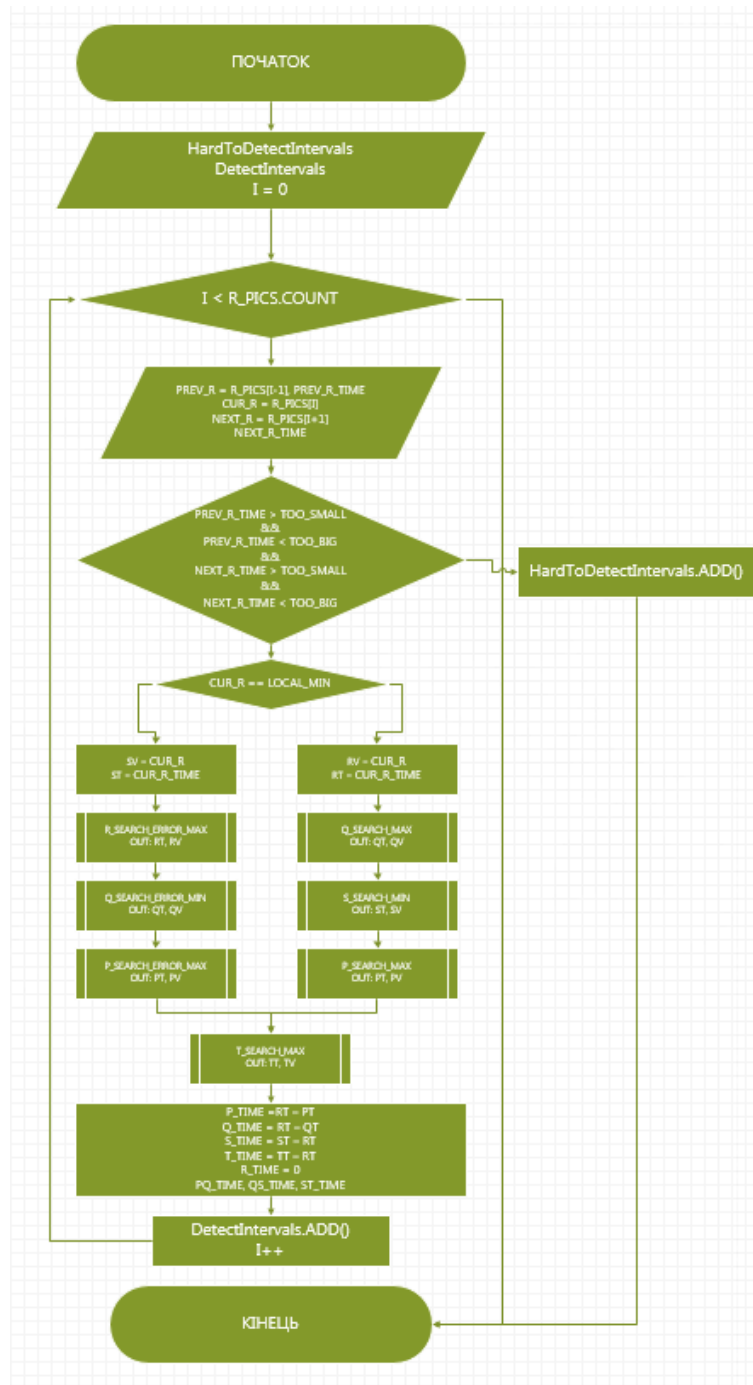


Рисунок 3.5 – Алгоритм розбиття ЕКГ на опорні точки інтервалів

3.4.1 Метод аналізу інтервальних даних

При аналізі готового набору інтервалів PQRST виникла проблема: програма могла описати людському оку два чітко різних інтервали, але з однаковими характеристиками. Це пов'язано з дуже розрідженою природою розглянутих даних (див. рис. 3.6).

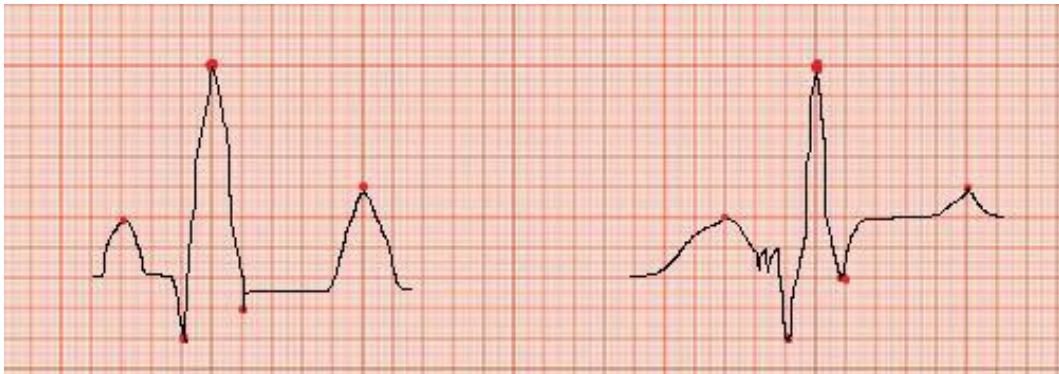


Рисунок 3.6 –Ілюстрація низької інформативності пікових показників

Мабуть, низька інформативність набору даних призводить до ідентифікації абсолютно різних інтервалів як схожих. Часові показники певною мірою покращують ситуацію, але, як видно з інтервалів ST на малюнку, вони не дають адекватної характеристики.

Рішення цієї проблеми може забезпечити виявлення відмінностей у значеннях ряду. Вивчивши поведінку кривої ділянки інтервалу PQRSST, варто зазначити, що в більшості випадків різниці значень відбуваються в діапазоні від 0 до 200 одиниць. Для позначення різних рівнів різниці (різниці між попереднім значенням інтервалу та поточним) було вирішено використовувати англійські літери (див. табл. 3.2).

Таблиця 3.2 – Алфавітні позначки інтервальних перепадів

Перепади в більшу сторону	
A	0
B	1-2
C	3-5
D	6-9
E	10-15
F	16-25
G	26-35
H	36-45
I	46-55
J	56-85
K	86-109
L	110-149
M	150-199
N	>200
Перепади в меншу сторону	
O	>200
P	150-199
Q	110-149
R	86-109
S	56-85
T	36-55
U	26-35
V	16-25
W	10-15
X	6-9
Y	3-5
Z	1-2

Інтервали ЕКГ досліджуються та розбиваються на буквені символи наступним чином:

- PQ: "UBFBZVJTZCX";
- QS: "CZYWUYGFXWW";
- CT: "YBEDVECYUJOJIGXFZ".

Таким чином звичайний набір чисел перетворюється на більш інформативний набір слів, який містить усі дані, необхідні програмі для аналізу. Поєднання числових і інтервальних наборів даних дозволяє охопити всі моменти поведінки кривої, які можуть вказувати на захворювання.

3.5 Алгоритм класифікації числових даних для проектування ПЗ

Щоб вибрати алгоритм, який найкраще підходить для прийняття рішень щодо стану пацієнта, розглянемо процес, за допомогою якого лікар-людина обробляє інформацію електрокардіограми. Спочатку лікар перевірить:

- Висота зубів;
- ширина гребнів і зазорів;
- частота і стабільність R-інтервалу.

Процес його мислення під час аналізу ЕКГ для можливого діагнозу можна описати за допомогою діаграми на малюнку 3.8:

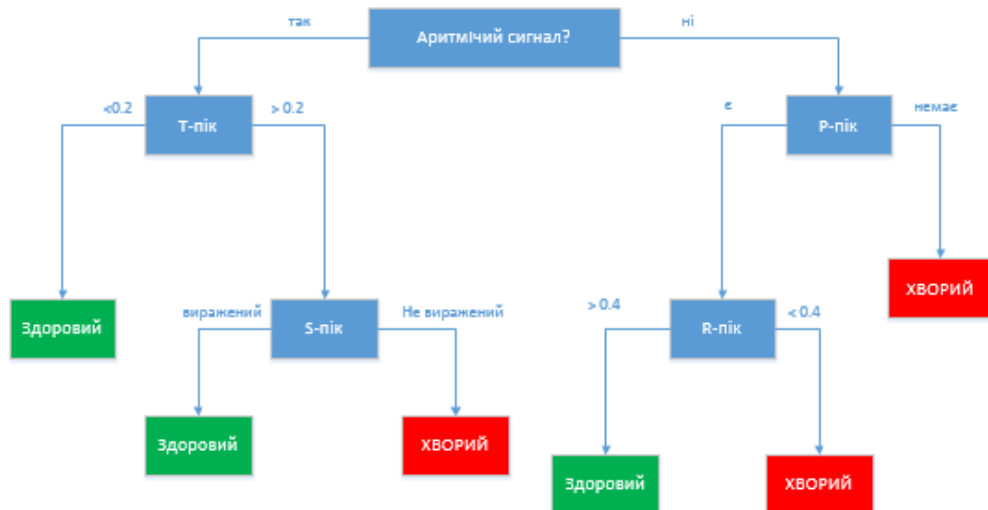


Рисунок 3.8 – Діаграма аналізу ЕКГ людиною

Така діаграма дуже схожа на приклад діаграми алгоритму машинного навчання дерева рішень (див. рис. 3.9).

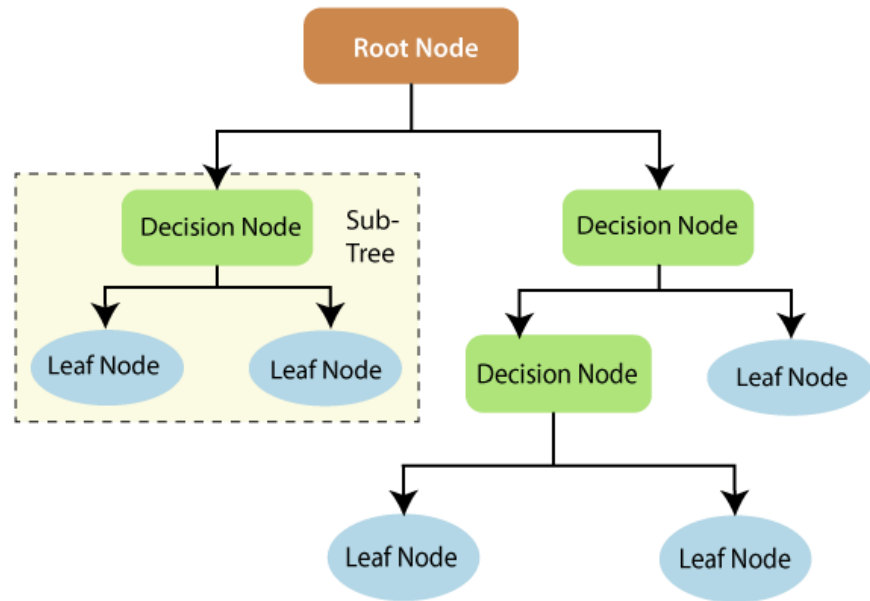


Рисунок 3.9 – Алгоритм «Дерево прийняття рішень»

DecisionTree складається з деревовидної структури, де верхній вузол є коренем дерева, а дерево рекурсивно ділиться на ряд вузлів від кореня до вузла, де остаточно приймається рішення.

Кожна вершина дерева (окрім листя) відповідає на певне запитання, тобто є декілька варіантів відповіді. Залежно від обраного варіанту буде здійснено перехід на початок наступного рівня. Кінцевим вершинам присвоюються мітки, що вказують на відповідність ідентифікованого об'єкта одному з класів.

Переваги використання цього алгоритму:

- Дозволяє тренувати моделі з використанням як кількісних, так і якісних вхідних даних;
- Захист від впливу занадто великої кількості змінних або змінних з високою кореляцією;
- Є кілька параметрів, які можна регулювати під час навчання моделі;
- Працює з викидами або відсутніми значеннями в наборі даних.

Недоліки використання цього алгоритму:

- Неправильне поводження з шумом;

- точно прогнозувати дані, подібні до даних про освіту, але погано прогнозувати дані, які значно відрізняються;
- Легко піддається перенавчанню.

Випадкові ліси (або Randomforest) є вдосконаленою модифікацією дерев рішень, які вирішують проблему перенавчання (див. рис. 3.10). Основний алгоритм навчається на різних підмножинах символів, які також вибираються випадковим чином.

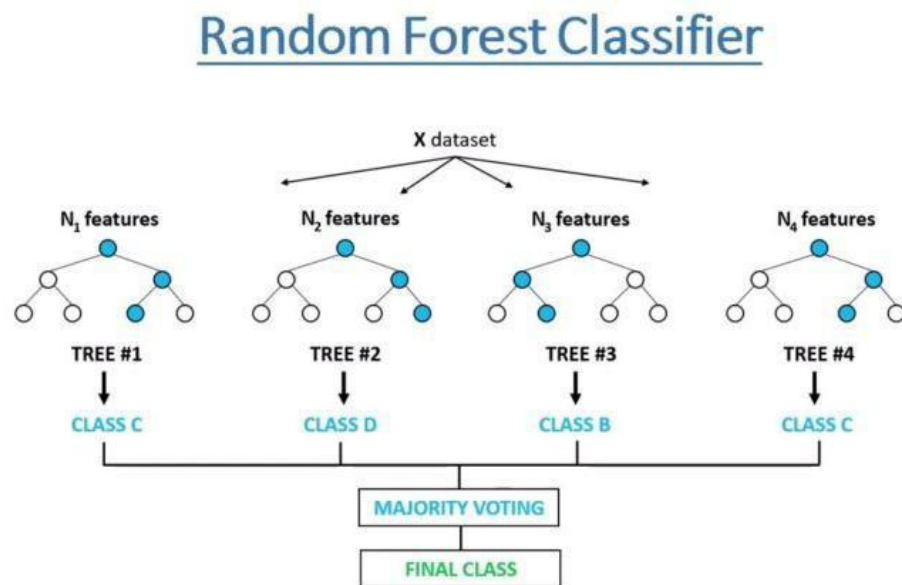


Рисунок 3.10 –Алгоритм «Випадковий ліс»

Ансамбль моделей за допомогою цього методу можна побудувати за таким алгоритмом:

- Нехай N позначає кількість об'єктів у наборі даних, а M позначає кількість ознак у цих об'єктах;
- представляє вибір L – окреме дерево рішень;
- Для кожного дерева в L ми визначаємо власне d_l -число ознак. d_l має бути менше M ;
- Тренуйте кожну модель окремо за допомогою L ;
- Для формування прогнозів ми об'єднуємо результати з окремих дерев.

У класифікаційних задачах рішення вибирається більшістю голосів, а в регресійних – середнім значенням.

Переваги використання даного алгоритму [13]:

- Має високу точність передбачення і кращий за лінійні алгоритми в більшості завдань;
- Нечутливий до масштабування та трансформації значень атрибутів;
- Не потрібно ретельно задавати параметри, добре працює «з коробки»;
- Можливість ефективної обробки даних з великою кількістю ознак і класів;
- Без повторного навчання додавання дерев майже завжди лише покращить композицію.

Недоліки використання алгоритму [13]:

- На відміну від окремого дерева, результати випадкового лісу важче інтерпретувати;
- Цей алгоритм працює гірше, ніж багато лінійних методів, коли у вибірці є велика кількість розріджених ознак (тексту);
- Погано сприймає шум.

3.6 Алгоритм класифікації інтервальних даних для проектування ПЗ

Оскільки інтервали ЕКГ представлені у вигляді речень з окремими словами, алгоритми, які їх обробляють, повинні бути ефективними для обробки текстових даних.

Коли ми говоримо про дані, зазвичай представляти величини або категорії, які містять елементи (наприклад, числа), у фіксованому списку. Якщо текст є сховищем неструктурованих даних (слів) і створюється як мова для взаєморозуміння між людьми, а не машинами, як перетворити його на класифікацію чи прогнозу модель, яку програма може зрозуміти?

За допомогою НЛП неструктурований текст можна розбити на більш короткі, більш структуровані частини інформації. Спочатку розділіть повний

текст на слова або речення. Далі корисно використовувати механізм, який розбиває слова на окремі лексеми, що підвищить точність аналізу.

Такі морфеми називають N-грамами, або комбінаціями суміжних слів або літер довжиною n , які можна знайти у вихідному тексті. Головною перевагою n -грам є те, що вони фіксують структуру мови, перетворюючи вільне мовлення на словник. Чим довша n -грама, тим більше контексту. Оптимальна довжина залежить від програми - якщо ваш n -грам закороткий, ви можете пропустити важливі відмінності. З іншого боку, якщо вони надто довгі, «здорові» рішення не будуть створені, і алгоритм буде перевчатися.

Далі ви можете застосувати якийсь метод до кластеризації або передбачити результати. Щоб визначити, чи хвора людина за її ЕКГ, потрібен алгоритм, який розділить дані на дві різні групи. Найбільш успішною групою таких алгоритмів є логістична регресія.

Логістична регресія (див. рис. 3.11) — це метод статистичного аналізу, який використовується для прогнозування цінності даних на основі попередніх спостережень. Отримана аналітична модель може враховувати багато вхідних критеріїв.

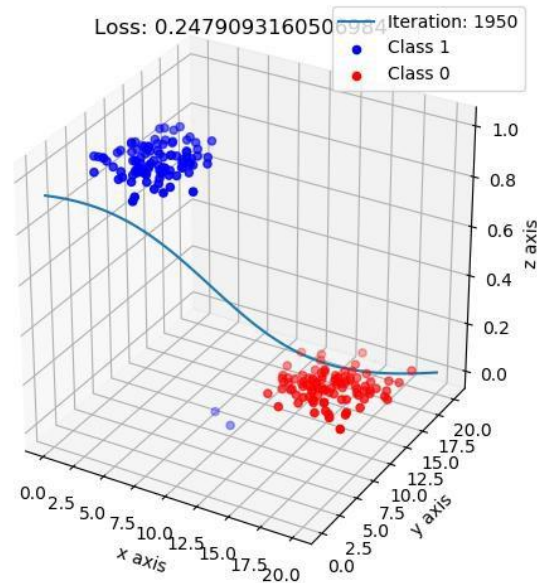


Рисунок 3.11 – Логістична регресія в тривимірному просторі

Бінарна логістична регресія використовується, коли залежна змінна (результат) може приймати лише два значення (зазвичай так або ні). Логістичну регресію можна використовувати для оцінки ймовірності події (у випадку набору даних ЕКГ, ймовірності захворювання) у конкретного суб'єкта.

У чому відмінність від лінійної регресії? Логістична регресія використовується, коли змінна відповіді є категоріальною змінною, а логістична регресія використовується, коли змінна відповіді є безперервною змінною, вираженою в числових еквівалентах.

Хоча створення таких моделей вважається добре дослідженою та оптимізованою областю машинного навчання та оптимізації, досягнення в області стохастичних градієнтів значно підвищили рівень якості за останні кілька років.

Велика кількість проблем машинного навчання та обробки сигналів є проблемами мінімізації втрат, а простим методом для RLM є стохастичний градієнтний спуск (SGD). Градієнтний спуск (рисунок 3.12) — метод виявлення мінімального значення функції похибки. Мінімізація будь-якої функції полягає в знаходженні її глобального мінімуму. Ця функція використовується для контролю помилок у прогнозах машинного навчання.

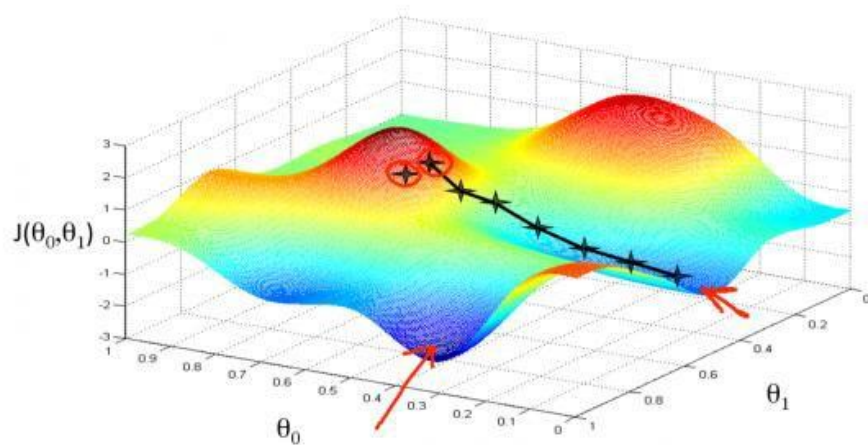


Рисунок 3.12 –Градієнтний спуск

Щоб знайти найглибшу помилку у функції, вам потрібно налаштувати параметри моделі. Нахил графічної функції є похідною. Нахил завжди вказує на найближчу западину.

Його час роботи не залежить від обсягу даних, тому він дуже популярний при роботі з великими масивами даних. Однак він не має чіткого критерію зупинки, а час, необхідний для отримання приблизного рішення, відносно короткий, і це надто повільно, щоб отримати точне рішення.

Інший підхід - SDCA.

Стохастичне подвійне координатне сходження (SDCA) [14] нещодавно з'явилося як найсучасніший метод для вирішення великомасштабних завдань навчання. Насправді це лише процес максимізації функції правильної відповіді, а не мінімізації функції помилки.

Стохастичний градієнтний підйом відрізняється від звичайного градієнтного підйому тим, що градієнт на кожному кроці обчислюється не як сума градієнтів кожного елемента в наборі, а як градієнт випадково вибраного елемента. Якщо використовується повний градієнтний спуск, результат завжди покращується з кожною ітерацією, а зі стохастичною функцією результат може навіть знизитися, але зазвичай він все одно буде прагнути до глобального максимуму.

Феномен подвійного підйому або спаду (див. рис. 3.13):



Рисунок 3.13 –Ефективність подвійного градієнтного спуску

4. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ

Впровадження програмного забезпечення має стати помічником лікаря в діагностиці захворювань серця. Фактично, це має бути перший крок в обстеженні пацієнта. Наприкінці процедури лікар повинен отримати найточніший опис прогнозу та кожну частину ЕКГ, на основі якої був створений прогноз.

4.1 Вимоги до програмного забезпечення

Залежно від поставленого завдання для впровадження такого програмного забезпечення необхідно виконати наступні кроки:

- Знайти базу даних, з якої можна витягти ЕКГ хворих і нехворих, ЕКГ не повинні бути попередньо оброблені або спрощені;
- Реалізувати запропоновані алгоритми читання та фільтрації даних;
- Реалізація запропонованого алгоритму аналізу ЕКГ за піками, а саме: алгоритм пошуку локальних мінімумів і максимумів з урахуванням можливих аномалій, класифікатор на основі алгоритму випадкового лісу;
- Реалізувати запропонований алгоритм аналізу електрокардіограми за значеннями інтервалів, а саме: алгоритм перетворення інтервалів у речення та алгоритм випадкового двокоординатного зростання;
- Розробка окремих модулів, що відповідають за формування експертних систем на основі штучного інтелекту;
- Розробляти інтерфейси користувача з використанням програмних алгоритмів;
- Вивчити роботу запропонованого алгоритму, оцінити його точність і швидкість обробки даних;
- Порівняти отримані результати з існуючими методами визначення електрокардіографічних захворювань.

На основі сформульованих завдань можна визначити деякі функціональні вимоги до програмного забезпечення.

Вимоги програми користувача:

- Зчитування даних ЕКГ з файлу .csv;
- Розділити дані ЕКГ на інтервали PQRST;
- Формувати мовний ланцюжок на основі різниці висот часового ряду;
- Запис даних про інтервали PQRST в набір даних з розширенням .tsv;
- Аналіз отриманих електрокардіограм щодо відомих захворювань;
- Відображення результатів аналізу на екрані у вигляді діаграм і прогнозів.

Вимоги до експертних систем:

- Зчитування даних ЕКГ з файлу .tsv;
- Створення систем штучного інтелекту, які навчаються з даних у файлах .tsv відповідно до запропонованих алгоритмів;
- Аналізуйте точність штучного інтелекту.

Також можна виділити деякі нефункціональні вимоги до програмного забезпечення:

- Можливість запуску програми на операційних системах Windows 7, 8 і 10;
- Інтуїтивно зрозумілий інтерфейс програми наочно представляє інформацію про результат;
- Можливість використання експертної системи як бібліотеки поза основною програмою користувача.

Метою даного дослідження є створення методу обробки та аналізу даних електрокардіограми, який дозволить підвищити ефективність і точність порівняно з існуючими методами.

4.2 Засоби розробки

У процесі розробки програмного забезпечення було вирішено використовувати мову C#. Цей вибір визначається наступними факторами:

- Легкодоступне об'єктно-орієнтоване програмування;
- Автоматична робота пам'яті;

- Суворі типізація;

— Велика бібліотека для всіх програмних потреб.

Зверніть особливу увагу на такі інструменти платформи .NET, які стануть у нагоді в процесі розробки.

4.2.1 ML.NET[15]

Усі звикли до того, що фахівці з машинного навчання зазвичай працюють з більшою кількістю людей

«Історично типова» мова для такої діяльності, наприклад Python. Однак не всі програмісти мають бажання використовувати іноземну, незнайому мову, а деякі програмісти просто не люблять Python.

На щастя, є подібні речі в інших мовах, які дозволяють вам заглибитися в алгоритми машинного навчання. Одним із таких прикладів є чудова бібліотека Microsoft ML.NET (див. рис. 4.1).

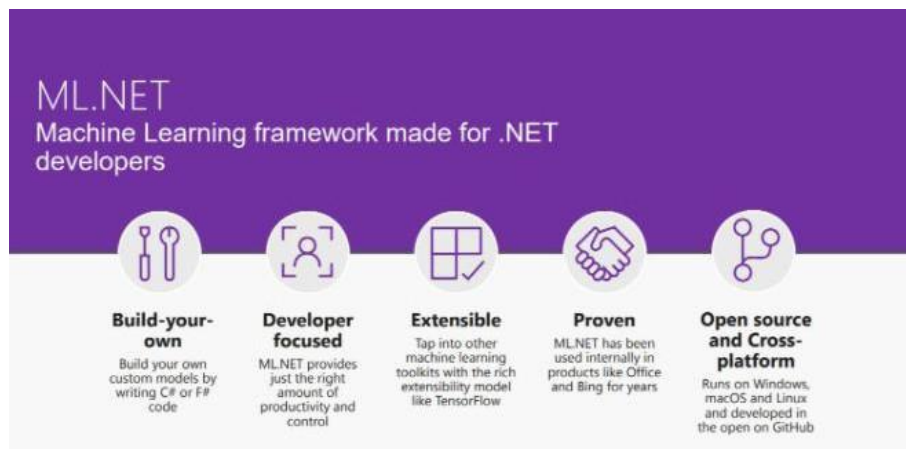


Рисунок 4.1 – Бібліотека машинного навчання C#

ML.NET надає програмістам інструменти для аналізу, прогнозування та створення моделей машинного навчання. Розробники можуть навчати моделі самостійно або повторно використовувати існуючі моделі з будь-якого джерела. Готова модель зберігається в зручному форматі .zip і може бути використана будь-якою програмою за потреби.

Використовуючи ML.NET, ви можете виконувати такі завдання:

- Класифікація;
- повернення;
- аномальне виявлення;
- глибоке навчання;
- Рекомендована система;
- Обробка природної мови.

Класи бібліотеки мають зручний та інтуїтивно зрозумілий інтерфейс, що забезпечує стандартну оцінку точності алгоритму для виведення результатів та оцінки ефективності машинного аналізу.

Якщо необхідно, базовий алгоритм можна параметризувати та розширити власною реалізацією функції. Бібліотека містить велику документацію та має власний репозиторій на GitHub, який містить приклади виконання найпопулярніших алгоритмів для відомих завдань: класифікація тюльпанів, пошук котів на зображеннях, виявлення негативних агресивних коментарів, прогноз погоди тощо.

4.2.2 WINDOWS FORMS

Це досить стара технологія, яка сягає часів народження мови C#. WindowsForms — це проста в освоєнні бібліотека, яка пропонує перевагу простоти порівняно з MFS і WCF [16].

Windows Forms — це інтерфейс для створення додатків для ПК в операційних системах Windows. Ця бібліотека забезпечує один із найефективніших способів створення настільних програм за допомогою інструментів візуального дизайнера. Такі функції, як елементи керування перетягуванням і скиданням, дозволяють зосередитися на реалізації логіки програми, а не витратити години на створення інтерфейсів користувача.

Ця бібліотека має потужний інтерфейс для роботи з графікою, що важливо при написанні програм, які обробляють дані ЕКГ - EssentialChart. Це інструмент для налаштування діаграми або графіка для відображення на панелі у спосіб, подібний до презентації PowerPoint. Це ідеальне рішення для

розробників, які хочуть додати вдосконалені, багатфункціональні та візуально привабливі діаграми до своїх програм Windows Forms. Діаграми використовуються як спосіб графічного відображення двох значень. Наприклад, лінійні діаграми можна використовувати у звітах про статистику здоров'я, які можуть відображати показники за певний час.

Нижче наведено деякі з основних функцій основної діаграми.

- Модель даних діаграми - об'єктна модель даних, яка може бути заповнена з файлу чи колекції, або безпосередньо з таблиці в базі даних;
- Може автоматично обчислювати інтервали в будь-якому діапазоні;
- Існують інструменти для експорту готової діаграми у файли найпопулярніших форматів, які можуть зберегти графічне зображення.

4.3 Архітектура ПЗ

Програма розділена на три окремі програми:

- ECGAnomalyDetection (програмне забезпечення для взаємодії з користувачем);
- ECGPeaksMLAnalyze (програмне забезпечення для навчання моделей аналізу опорних точок);
- ECGWordsAnalyze (програма для навчання моделей інтервального аналізу у формі речення).

4.3.1 Програмне забезпечення для взаємодії з користувачем

Побудований на основі архітектурного шаблону під назвою MVP (див. рис. 4.2).

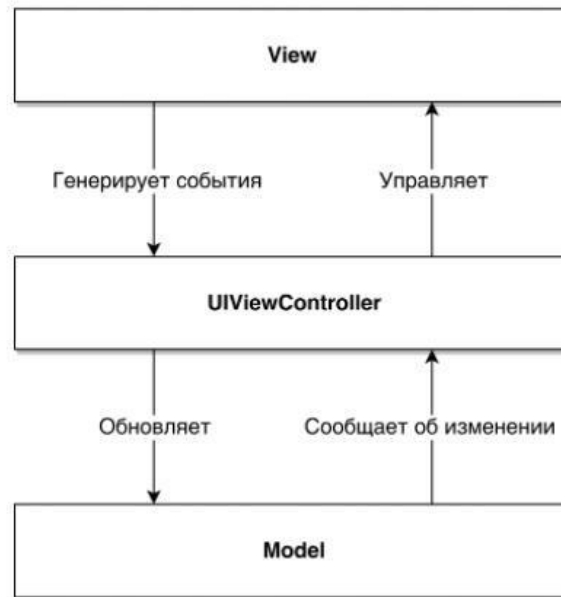


Рисунок 4.2 – Концепція шаблону модель-перегляд-презентатор

Цей метод дозволяє створити абстракцію частини перегляду вашої програми. Для цього вам потрібно призначити інтерфейс із певним набором властивостей і методів.

Презентатор отримує посилання на реалізацію інтерфейсу, підписується на події, що відбуваються (наприклад, натискання кнопки, завантаження файлу) і модифікує модель.

У програмному забезпеченні стандартна форма WindowsForms виконує роль View (див. рис. 4.3). Він містить такі елементи:

- Кнопка завантаження сигналів ЕКГ з файлу;
- Кнопка для перетворення даних ЕКГ в параметризований список;
- кнопка аналізу ЕКГ;
- Презентація графіків електрокардіограм;
- Перемикачі для вибору захворювань для аналізу.

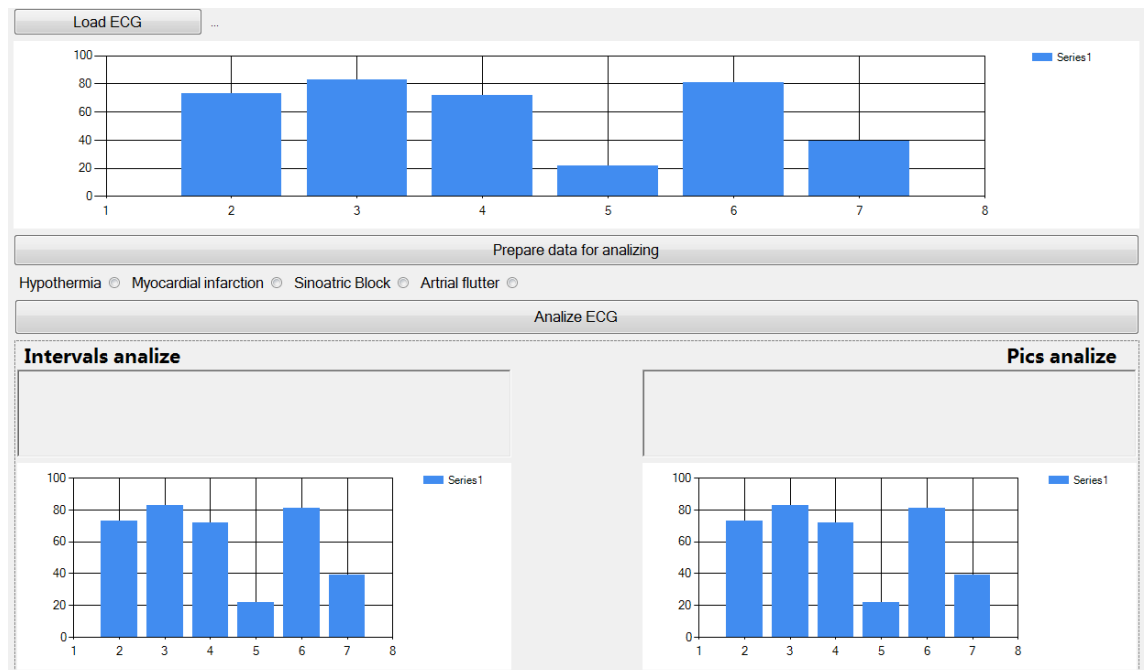


Рисунок 4.3 – Представлення View

Екранний клас виконує роль Ведучого. Під час взаємодії з формою вона перехоплює події, що відбуваються (тобто натискання однієї з кнопок), і змінює модель або отримує чи обробляє її дані.

Модель є класом ECGConverter, який зберігає інформацію про повні інтервали ЕКГ і за потреби формує свій параметризований список.

Також важливу роль відіграють допоміжні служби:

- Файловий менеджер – допомагає зберігати новостворені набори даних у файли;

- Вейвлет-перетворення - За допомогою вейвлет-перетворення кількість точок на електрокардіограмі зменшується в чотири рази без втрати обсягу інформації;

- PicksAnalyzer – аналізує параметризований список інтервалів ЕКГ для визначення наявності в них захворювання (інформація про локальні мінімуми та максимуми);

- IntervalsAnalyzer – аналізує параметризований список інтервалів ЕКГ, щоб визначити наявність у них захворювання (інформація про зміни інтервалів на діаграмі).

Повну інформацію про компонент програми ECGAnomalyDetection можна побачити на діаграмі класів (див. рис. 4.4).

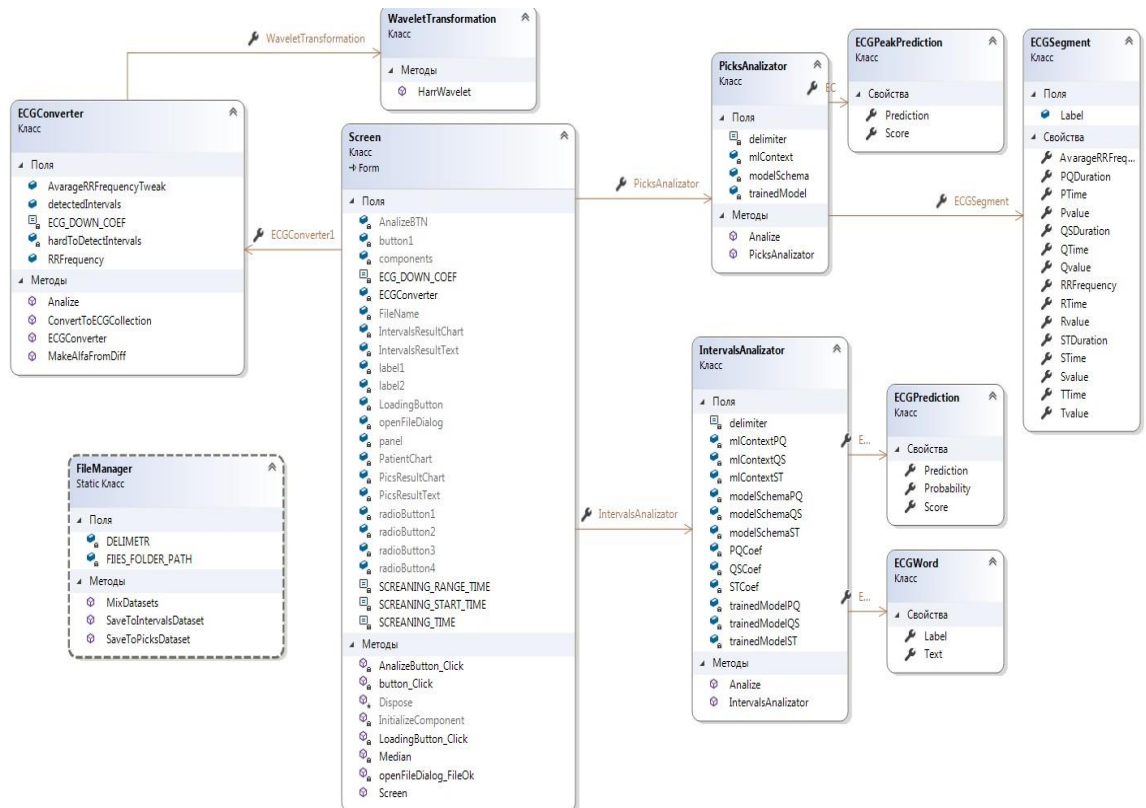


Рисунок 4.4 – Діаграма класів для клієнтської частини програми

4.3.2 База даних прогнозів захворювань

Метою допоміжної бібліотеки є формування штучного інтелекту, аналіз параметрів електрокардіограми для визначення наявності захворювання та прогнозування ймовірності захворювання пацієнта.

Для створення штучного інтелекту в ECGPeaksMLAnalyze використовується стандартний клас `MLContext.Regression.Trainers.FastForest`. Параметри його налаштування наведені в таблиці 4.1.

Таблиця 4.1 – Параметри налаштування алгоритму

NumberOfLeaves	Максимальна кількість листків у кожному дереві регресії
NumberOfTrees	Загальна кількість дерев рішень, які потрібно створити в ансамблі
MinimumExampleCountPerLeaf	Мінімальна кількість точок даних, необхідних для формування нового листа дерева

Стандартний

клас

`MLContext.BinaryClassification.Trainers.SdcaLogisticRegression`

використовується для створення штучного інтелекту в `ECGWordsAnalyze`.

Параметри його налаштування наведено в таблиці 4.2.

Таблиця 4.2 – Параметри налаштування алгоритму

L1Regularization	Додає штрафну величину до функції мінімізації лише для великих коефіцієнтів
L2Regularization	Додає штрафну величину до функції мінімізації
MaximumNumberOfIterations	Максимальна кількість ітерацій для пошуку максимуму

4.4 Інструкція користувача

Відкривши програму, користувач побачить панель із кнопками та порожніми полями (див. рис. 4.5).

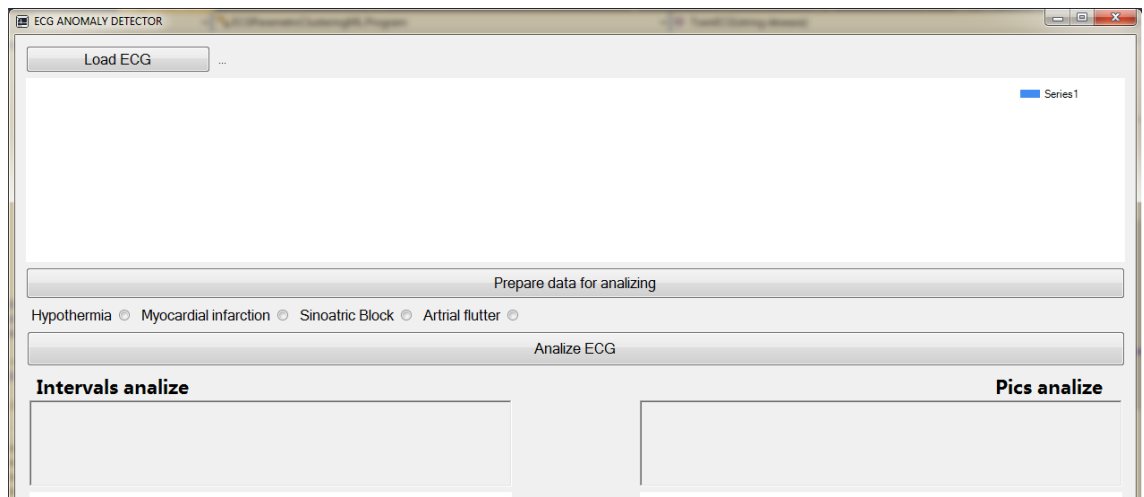


Рисунок 4.5 – Початок роботи з програмою

Щоб завантажити дані ЕКГ, натисніть кнопку «LOAD ECG» і виберіть файл в окремому вікні. Після цього у вікні користувач побачить графічне представлення даних ЕКГ (див. рис. 4.6). Якщо необхідно, він може скористатися цією функцією, щоб вибрати область на діаграмі для ближчого розгляду.

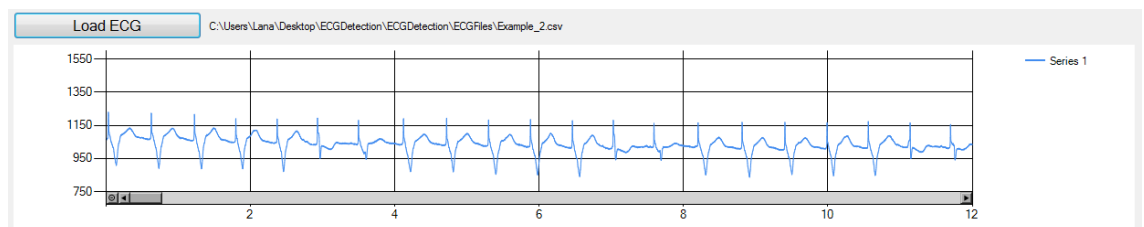


Рисунок 4.6 – ЕКГ графік

Після цього користувач повинен натиснути кнопку «Готов до аналізу даних» і отримати у відповідь графічне підтвердження розкладання ЕКГ на піки R-R (див. рис. 4.7).

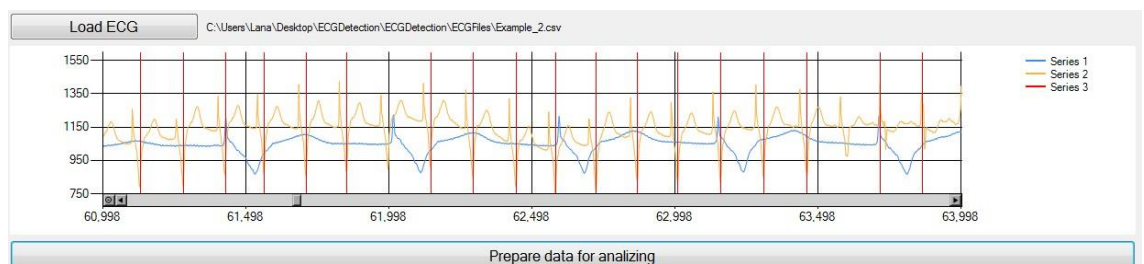


Рисунок 4.7 – Графікробиття ЕКГ на R-Рпіки

Далі користувач може вибрати захворювання для аналізу та натиснути кнопку «Аналіз ЕКГ». Через деякий час він отримує прогнози в полях Intervalsanalyze і PicksAnalyze (див. рис. 4.8).

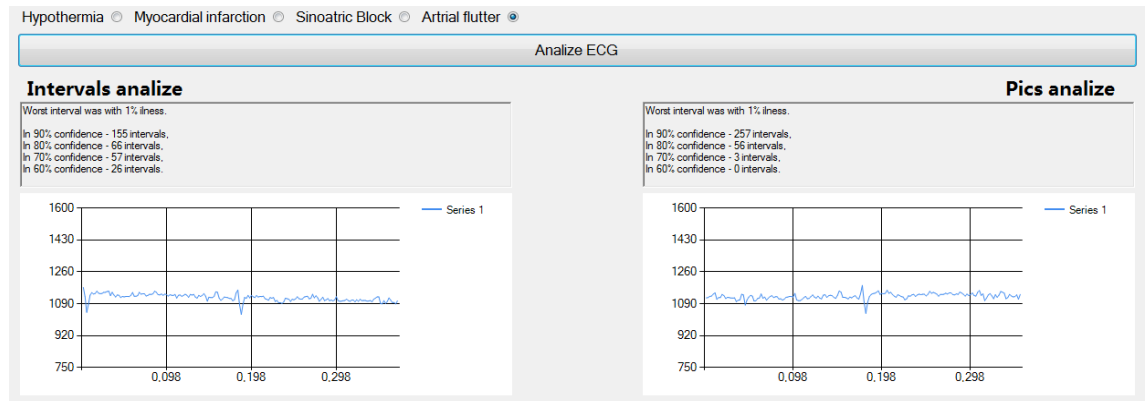


Рисунок 4.8 – Результати аналізу на хворобу

За допомогою таких інструментів, як C#, WindowsForms і ML.NET, розроблено програмне забезпечення для реалізації методу класифікації та аналізу даних ЕКГ у вигляді часових рядів.

Програма користувача працює без помилок, коректно обробляє набір даних і видає очікуваний інформативний вихід.

Це призвело до появи двох незалежних бібліотек: ECGPeaksMLAnalyze та ECGWordsMLAnalyze, які можна використовувати для інших майбутніх досліджень і вдосконалення створеного методу.

5. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДУ

Щоб виявити точність створеного алгоритму та порівняти його ефективність з існуючими аналогічними алгоритмами, необхідно вирішити, які показники використовувати при його аналізі.

5.1 Ключові критерії оцінки ефективності роботи методу

Першим критерієм оцінки є здатність ідентифікувати ключові параметри нормальних, зашумлених і хворобливих серій ЕКГ.

Така оцінка буде розрахована відповідно до рівняння 5.1:

$$k_{\text{розпізнавання}} = \frac{n_{\text{оброблених даних}}}{n_{\text{загальне}}} \quad (5.1)$$

Другим важливим критерієм є алгоритмічна складність аналізу даних, розрахована згідно з рівнянням 5.2:

$$O(f(n)) = 2 * \frac{n_{\text{заг}}}{4} + \sqrt{\frac{n_{\text{заг}}}{4}} \quad (5.2)$$

Третій важливий критерій — швидкість роботи алгоритму класифікації даних. Вимірюється експериментально.

Машинне навчання також можна використовувати для додаткового визначення ефективності методів класифікації часових рядів.

Перш ніж заглибитися в поняття точності, необхідно розрізнити два різних типи помилок:

- Помилка I типу або класифікація позитивного результату як негативного. Позначимо FP;

- Помилка типу II або класифікація негативного результату як позитивного FN.

Усі передбачення, створені алгоритмом, загалом можна розділити на чотири типи (див. рис. 5.1).

		Prediction	
		1	0
Actual	1	True Positive (TP)	False Negative (FN)
	0	False Positive (FP)	True Negative (TN)

Рисунок 5.1 – Матриця оцінки результатів алгоритму класифікації

Тепер, завдяки цій класифікації, можна визначити ключові показники для оцінки алгоритмів.

Чутливість алгоритму — це здатність алгоритму ідентифікувати об'єкти групи 1 серед усіх запропонованих об'єктів. Таким чином, чутливість дозволяє побачити частку справжніх позитивних класифікацій відносно загальної кількості позитивних класифікацій. Його формула розрахунку отримана з рівняння 5.3:

$$Sensitivity = \frac{TP}{TP+FN} \quad (5.3)$$

Чутливість характеризує точність моделі бінарної класифікації відносно активних спостережень. Точність моделі негативного спостереження відображає значення, яке називається специфічністю.

Специфіка алгоритму стосується його здатності не розглядати об'єкти групи 0 як об'єкти групи 1. Розраховується згідно з рівнянням 5.4:

$$Specific = \frac{TN}{TN+FP} \quad (5.4)$$

Чим нижча чутливість, тим більша кількість хибнонегативних спостережень, тобто позитивних спостережень, неправильно визначених як негативні. Якщо хибнонегативних спостережень немає, верхня межа чутливості становить 1. Модель з високою чутливістю має вищу ймовірність правильного визначення позитивних прикладів.

Площа під кривою ROCAUC дозволяє знайти оптимальний баланс між чутливістю та специфічністю вашої моделі, де чутливість і специфічність

дорівнюють 1, коли немає хибнопозитивних і хибнонегативних класифікацій. Має бути єдність.

Наприклад, $AUC=0,7$ можна інтерпретувати як 70% ймовірність того, що модель зможе правильно розділити дані. $AUC = 0$ – такий класифікатор завжди визначає позитивні приклади як негативні, тобто рівень помилок становить 100%.

На практиці ефективність моделі класифікується згідно з таблицею 5.1 за значенням AUC:

Таблиця 5.1 – Оцінка значень AUC

Значення	Оцінка
≤ 0.5	Модель не працює, її ефективність нижча за звичайний рандом.
> 0.5 ≤ 0.6	Модель може видавати правильні результати за наявності дуже характерних ознак у протестованому екземплярі. Проте для більшості випадків все ще не працює.
> 0.6 ≤ 0.8	Модель працює, може використовуватись для задач де ціна помилки не є критично важливою. Наприклад в контекстній рекламі в ігрових проектах для демонстрації можливостей машинного навчання, у програмах підбору товарів для користувача тощо.
> 0.8 ≤ 0.94	Модель працює з хорошою точністю, може використовуватись для будь яких цілей окрім тих, де ціною помилки є людське життя або втрата великих грошових сум.
> 0.94	Ідеальна модель, може використовуватись будь де. Це те значення до якого варто намагатись при побудові будь якого алгоритму.

5.2 Оцінка ефективності роботи методу

Коефіцієнт ідентифікації визначається на основі даних різної довжини та складності (див. табл. 5.2):

Таблиця 5.2 – Коефіцієнт розпізнавання ЕКГ

Складність даних	Розмір в мілісекундах	Розмір в інтервалах	Кількість розпізнаних інтервалів	Коефіцієнт
Легка (синусоїда нормального типу)	150000	491	434	88.39%
Середня (синусоїда неправильна, але інтервали яскраво виражені)	150000	512	414	80.85%
Важка (синусоїда зашумлена, інтервали не	150000	414	316	76.32%
Легка (синусоїда нормального типу)	350000	1132	990	87.45%
Середня (синусоїда неправильна, але інтервали яскраво виражені)	350000	1184	974	82.26%
Важка (синусоїда зашумлена, інтервали не	350000	1025	712	69.46%
Легка (синусоїда нормального типу)	600000	1953	1523	77.98%
Середня (синусоїда неправильна, але інтервали яскраво виражені)	600000	2021	1671	82.68%
Важка (синусоїда зашумлена, інтервали не	600000	1686	1079	63.99%

Як видно з таблиці, рівень розпізнавання практично ніколи не опускається нижче 75%. Це означає, що принаймні 3/4 усього обсягу даних визначено як дійсні інтервали та може бути подано на аналіз. Цього достатньо для постановки діагнозу.

Ефективність аналізатора числових характеристик:

- Гіпотермія - 98,26%;
- інфаркт міокарда - 98,13%;
- Синоатріальна блокада - 98,84%;
- фібриляція передсердь - 99,02%.

Ефективність аналізатора інтервальних ознак

Залежно від важливості інтервалу у конкретного пацієнта ефективність алгоритму може бути знижена (див. табл. 5.3). Тому найкращим рішенням є аналіз ЕКГ за найкращими ознаками.

Таблиця 5.3 – Ефективність інтервальних аналізаторів

Гіпотермія		
PQ	TP	88.46%
	FP	17.42%
	TN	80.53%
	FN	4.77%
	ROC AUC	94.01%
QS	TP	81.08%
	FP	14.88%
	TN	85.85%
	FN	3.83%
	ROC AUC	88.58%
ST	TP	97.26%
	FP	6.31%
	TN	91.94%
	FN	1.44%
	ROC AUC	98.53
Інфаркт міокарда		
PQ	TP	89.71%
	FP	7.21%

Продовження таблиці 5.3

	TN	95.34%
	FN	8.21%
	ROC AUC	97.4%
QS	TP	82.65%
	FP	15.64%
	TN	85.31%
	FN	10.12%
	ROC AUC	90.87%
ST	TP	87.83%
	FP	7.31%
	TN	95.17%
	FN	6.13%
	ROC AUC	96.73%
Мерехтіння передсердь		
PQ	TP	85.52%
	FP	14.39%
	TN	85.71%
	FN	16.03%
	ROC AUC	94.74%
QS	TP	83.67%
	FP	23.94%
	TN	82.44%
	FN	2.16%
	ROC AUC	83.15%
ST	TP	97.24
	FP	4.35%
	TN	94.44
	FN	2.16%
	ROC AUC	98.99%

Продовження таблиці 5.3

Синоатріальна блокада		
PQ	TP	84.81%
	FP	12.9%
	TN	89.92%
	FN	17.15%
	ROC AUC	95.51%
QS	TP	80.55
	FP	19.49%
	TN	80.47
	FN	10.47%
	ROC AUC	89.14%
ST	TP	85.52%
	FP	14.39%
	TN	85.71%
	FN	16.03%
	ROC AUC	93.53%

Середня ефективність інтервального показника найкращого інтервального аналізу становить приблизно 97,6%.

Тому в цьому розділі сформовані основні критерії дослідження ефективності алгоритмів. На основі цих критеріїв аналізуються алгоритми класифікації числових послідовностей за піками та інтервалами. Отримані результати наведені на рисунках 5.2-5.3.

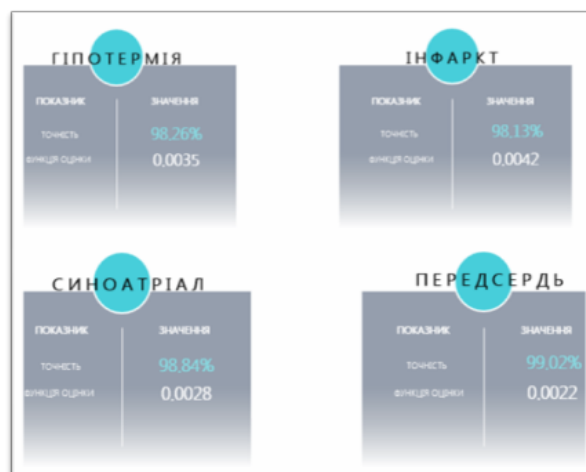


Рисунок 5.2 –Ефективність пікового методу

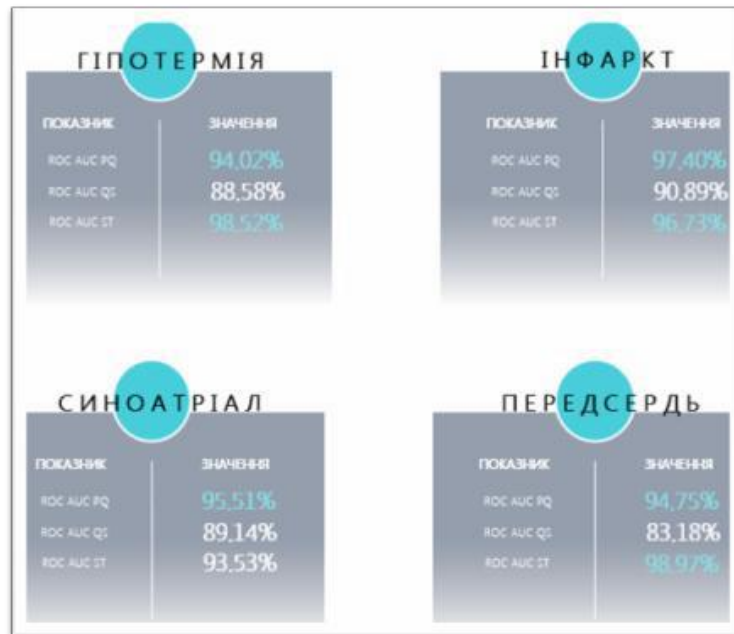


Рисунок 5.3 –Ефективність інтервального методу

Порівняно з аналогічними алгоритмами, ефективність цього алгоритму покращена на 7,13% і 6,08% для числових і інтервальних показників відповідно (див. рис. 5.4).

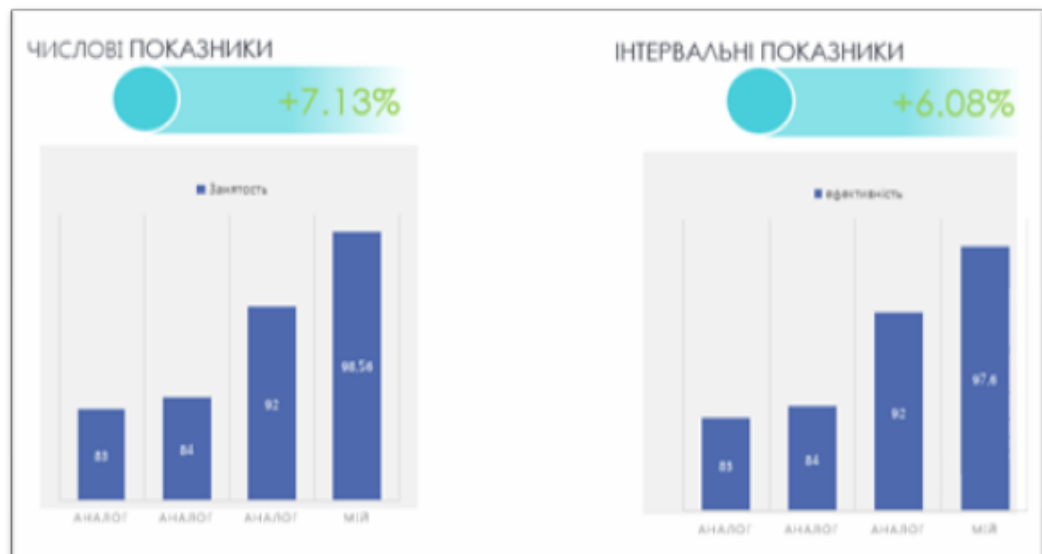


Рисунок 5.4 –Приріст ефективності нового методу

6. ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота «Автоматизована система аналізу сигналів електрокардіограм» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

6.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Автоматизована система аналізу сигналів електрокардіограм» є

оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 6.1 [19]

Таблиця 6.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту	Технічні та споживчі властивості продукту	Технічні та споживчі властивості продукту на	Технічні та споживчі властивості продукту	Технічні та споживчі властивості продукту
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційни	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної	Ринок малий, але має позитивну	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній не	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					

8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела	Потрібні незначні фінансові ресурси. Джерела фінансування	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовують
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 6.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	5	4
2. Ринкові переваги (наявність аналогів)	2	3	3
3. Ринкові переваги (ціна продукту)	2	2	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	2
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5
11. Практична здійсненність (термін реалізації)	3	4	5
12. Практична здійсненність (розробка документів)	4	5	4
Сума балів	36	42	41
Середньоарифметична сума балів $СБ_c$	39,7		

За результатами розрахунків, наведених в таблиці 6.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 6.3 [19]

Таблиця 6.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів $СБ_c$, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Автоматизована система аналізу сигналів електрокардіограм» становить 39,7 бала, що, відповідно до таблиці 6.3, свідчить про комерційну

важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

6.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

Одиничний параметричний індекс розраховуємо за формулою [19]:

$$q_i = \frac{P_i}{P_{\text{базі}}} \quad (6.1)$$

де q_i – одиничний параметричний індекс, розрахований за i -м параметром;
 P_i – значення i -го параметра виробу;
 $P_{\text{базі}}$ – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 6.4.

Таблиця 6.4 – Основні техніко-економічні показники аналога та розробки, що проектується

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Кількість параметрів електрокардіограми що підлягають обліку	од.	5	13	2,6	0,3
Кількість місць розміщення (хмара, мобільно, ПК)	од.	2	3	1,5	0,1
Рівень захищеності баз даних (за 10-бальною шкалою)	бал	5	8	1,6	0,15
Доступність інтерфейсу	бал	7	9	1,26	0,25
Можлива кількість додатково інтегрованих модулів	од.	2	4	2	0,2
Річні експлуатаційні витрати на підтримку автоматизованої системи	грн	860	720	0,84	0,45
Вартість розміщення на одному робочому місці (рік)	грн	2200	2000	0,91	0,55

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою [19]:

$$I_{HP} = \prod_{i=1}^n q_i, \quad (6.2)$$

де I_{HP} – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i -м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому $I_{HP} = 1$.

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра [19]:

$$I_{TP} = \sum_{i=1}^n q_i \cdot \alpha_i, \quad (6.3)$$

де I_{TP} – груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

q_i – одиничний параметричний показник i -го параметра;

α_i – вагомість i -го параметричного показника, $\sum_{i=1}^n \alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{HP} = 2,6 \cdot 0,3 + 1,5 \cdot 0,1 + 1,6 \cdot 0,15 + 1,26 \cdot 0,25 + 2 \cdot 0,2 = 1,89.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою [20]:

$$I_{EP} = \sum_{i=1}^m q_i \cdot \beta_i, \quad (6.4)$$

де I_{EP} – груповий параметричний індекс за економічними показниками;

q_i – економічний параметр i -го виду;

β_i – частка i -го економічного параметра, $\sum_{i=1}^m \beta_i = 1$;

m – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці .

$$I_{EP} = 0,84 \cdot 0,45 + 0,91 \cdot 0,55 = 0,88.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою [19]:

$$K_{INT} = I_{HP} \cdot \frac{I_{TP}}{I_{EP}}, \quad (6.5)$$

$$K_{INT} = 1 \cdot 1,89 / 0,88 = 2,15.$$

Інтегральний показник конкурентоспроможності $K_{INT} > 1$, отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

6.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Автоматизована система аналізу сигналів електрокардіограм», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

6.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [19]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (6.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=21$ дні.

$$Z_o = 16950,00 \cdot 28 / 21 = 22600,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 6.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту з розробки автоматизованої системи аналізу сигналів електрокардіограм	16950,00	807,14	28	22600,00
Інженер-програміст 1-ї категорії	18800,00	895,24	28	25066,67
Інженер-аналітик систем автоматизованого управління	16500,00	785,71	21	16500,00
Консультант (Лікар-кардіолог вищої категорії)	18000,00	857,14	7	6000,00
Провідний фахівець	9200,00	438,10	28	12266,67
Всього				82433,33

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Автоматизована система аналізу сигналів електрокардіограм» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (6.7)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (6.8)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=6700,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [19];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 21$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 6700,00 \cdot 1,10 \cdot 1,35 / (21 \cdot 8) = 59,22 \text{ грн.}$$

$$З_{р1} = 59,22 \cdot 4,50 = 266,50 \text{ грн.}$$

Таблиця 6.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника, грн
Установка обчислювального обладнання	4,50	2	1,10	59,22	266,50
Підготовка робочого місця розробника автоматизованої системи управління	3,60	2	1,35	72,68	261,66
Інсталяція програмного забезпечення розробки (моделювання) систем управління	4,35	4	1,70	91,53	398,14
Введення кодів модулю формування бази даних	2,20	5	1,70	91,53	201,36
Введення кодів модулю аналізу сигналів	0,50	5	1,10	59,22	29,61
Введення кодів модулю забезпечення онлайн доступу	2,50	5	1,70	91,53	228,82
Налагодження системної взаємодії	6,00	4	1,50	80,76	484,55
Інтеграція модулів автоматизованої системи	2,50	3	1,35	72,68	181,71
Всього					2052,35

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (6.9)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{од}} = (82433,33 + 2052,35) \cdot 11 / 100\% = 9293,43 \text{ грн.}$$

6.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{од}}) \cdot \frac{H_{zn}}{100\%} \quad (6.10)$$

де H_{zn} – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (82433,33 + 2052,35 + 9293,43) \cdot 22 / 100\% = 20631,40 \text{ грн.}$$

6.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Автоматизована система аналізу сигналів електрокардіограм».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\text{в}j}, \quad (6.11)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

$C_{\text{в}j}$ – вартість відходів j -го найменування, грн/кг.

$$M_1 = 3,0 \cdot 212,00 \cdot 1,1 - 0,0 \cdot 0,0 = 699,60 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 6.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір офісний А4, Crystal Pro80, клас С, 500 л, UPM	212,00	3,0	0,0	0,0	699,60
Папір офісний А3, Maestro Standart+, клас В, 80 г/м2, 500 л, Mondi	462,00	1,0	0,0	0,0	508,20
Папір офісний Офіс Центр А5 80г/м2 500 аркушів клас С	132,00	3,0	0,0	0,0	435,60
Органайзер для канцелярського начиння	211,00	2,0	0,0	0,0	464,20
ДИСК CD-R VERBATIM 700MB 80MIN 52X BULK 50	29,00	1,0	0,0	0,0	31,90
USB флеш накопичувач 32 ГБ	172,00	1,0	0,0	0,0	189,20
Всього					2889,70

6.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_6), які використовують при проведенні НДР на тему «Автоматизована система аналізу сигналів електрокардіограм», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_6 = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (6.12)$$

де H_j – кількість комплектуючих j -го виду, шт.;

C_j – покупна ціна комплектуючих j -го виду, грн;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$).

$$K_6 = 1 \cdot 4011,00 \cdot 1,1 = 4412,10 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 6.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Жорсткий диск WD 3.5" SATA III 1000Gb 7200rpm 64mb (WD1003FZEX)	1	4011,00	4412,10
Зовнішній жорсткий диск 1Tb Transcend StoreJet (TS1TSJ25M3S) Iron Gray	1	2299,00	2528,90
Всього			6941,00

6.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i, \quad (6.13)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.}i}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10 \dots 1,12$);

k – кількість найменувань устаткування.

$$B_{\text{спец}} = 43748,00 \cdot 1 \cdot 1,01 = 44185,48 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 6.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Сервер бази даних на основі PEOM Комп'ютер ARTLINE Overlord X81WHITE (X81WHITEv23)	1	43748,00	44185,48
Всього			44185,48

6.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{инрг}} \cdot C_{\text{прог.і}} \cdot K_i, \quad (6.14)$$

де $C_{\text{инрг}}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прог.і}}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{\text{прог}} = 245,00 \cdot 2 \cdot 1,01 = 494,90 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 6.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Доступ до мережі Internet (високошвидкісний) грн/місяць	2	245,00	494,90
Емулятор серверної інтернет-платформи для моделювання роботи автоматизованої системи обліку робочого часу	1	5600,00	5656,00
Система імітаційного комп'ютерного моделювання SIMULINK програмного середовища MATLAB	1	9500,00	9595,00
Система керування реляційними базами даних MySQL	1	8210,00	8292,10
Всього			24038,00

6.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{б}}{T_{е}} \cdot \frac{t_{вик}}{12}, \quad (6.15)$$

де $Ц_{б}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{е}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (42899,00 \cdot 2) / (3 \cdot 12) = 2383,28 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 6.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Програмно-аналітичний комплекс проектування на базі ПК HP i7-JD3200-UA23	42899,00	3	2	2383,28
Обладнання виводу інформації Лазерний принтер HP LaserJet Pro M102w c Wi-Fi (G3Q35A)	7499,00	4	2	312,46
Робоче місце інженера-дослідника спеціалізоване	7700,00	5	2	256,67
Офісна оргтехніка	8990,00	5	2	299,67
Приміщення лабораторії досліджень	629000,00	25	2	4193,33
ОС Windows 11	8460,00	3	2	470,00
Пакет Microsoft Office 2019	7840,00	3	2	435,56
Маршрутизатор Asus ZenWiFi AX Mini XD4 3PK White (90IG07M0-MO3C40)	9029,00	4	2	376,21
Всього				8727,17

6.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (6.16)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 7,50$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,36 \cdot 220,0 \cdot 7,50 \cdot 0,95 / 0,97 = 594,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 6.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Програмно-аналітичний комплекс проектування на базі ПК HP i7-JD3200-UA23	0,36	220,0	594,00
Обладнання виводу інформації Лазерний принтер HP LaserJet Pro M102w c Wi-Fi (G3Q35A)	0,25	4,0	7,50
Робоче місце інженера-дослідника спеціалізоване	0,07	220,0	115,50
Офісна оргтехніка	0,45	1,3	4,39
Маршрутизатор Asus ZenWiFi AX Mini XD4 3PK White (90IG07M0-MO3C40)	0,02	200,0	30,00
Сервер бази даних на основі ПЕОМ Комп'ютер ARTLINE Overlord X81WHITE (X81WHITEv23)	0,32	200,0	480,00
Всього			1231,39

6.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Автоматизована система аналізу сигналів електрокардіограм» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (6.17)$$

де H_{cb} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cb} = 22\%$.

$$B_{cb} = (82433,33 + 2052,35) \cdot 22 / 100\% = 18586,85 \text{ грн.}$$

6.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (6.18)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 33\%$.

$$B_{cn} = (82433,33 + 2052,35) \cdot 33 / 100\% = 27880,28 \text{ грн.}$$

6.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_s = (Z_o + Z_p) \cdot \frac{H_{is}}{100\%}, \quad (6.19)$$

де H_{is} – норма нарахування за статтею «Інші витрати», прийmemo $H_{is} = 50\%$.

$$I_s = (82433,33 + 2052,35) \cdot 50 / 100\% = 42242,84 \text{ грн.}$$

6.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків;

витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (6.20)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{нзв} = 115\%$.

$$B_{нзв} = (82433,33 + 2052,35) \cdot 115 / 100\% = 97158,54 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Автоматизована система аналізу сигналів електрокардіограм» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{од} + Z_n + M + K_e + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сн} + I_e + B_{нзв}. \quad (4.21)$$

$$B_{заг} = 82433,33 + 2052,35 + 9293,43 + 20631,40 + 2889,70 + 6941,00 + 44185,48 + 24038,00 + 8727,17 + 1231,39 + 18586,85 + 27880,28 + 42242,84 + 97158,54 = 388291,76 \text{ грн.}$$

Загальні витрати $ЗВ$ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (6.22)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta=0,95$.

$$ЗВ = 388291,76 / 0,95 = 408728,17 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження

результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Автоматизована система аналізу сигналів електрокардіограм» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості організацій-споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості організацій-споживачів	750	900	900	500

N – кількість організацій-споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 8500 організацій;

C_o – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 2200,00 грн на робоче місце;

$\pm \Delta C_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 207,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [19]:

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (6.23)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo $\rho = 45\%$;

\mathcal{D} – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\mathcal{D} = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (207,00 \cdot 8500,00 + 2407,00 \cdot 750) \cdot 0,83 \cdot 0,45 \cdot (1 - 0,18/100\%) = 1091775,98 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (207,00 \cdot 8500,00 + 2407,00 \cdot 1650) \cdot 0,83 \cdot 0,45 \cdot (1 - 0,18/100\%) = 1755248,68 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (207,00 \cdot 8500,00 + 2407,00 \cdot 2550) \cdot 0,83 \cdot 0,45 \cdot (1 - 0,18/100\%) = 2418721,38 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (207,00 \cdot 8500,00 + 2407,00 \cdot 3050) \cdot 0,83 \cdot 0,45 \cdot (1 - 0,18/100\%) = 2787317,33 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (6.24)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,2$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned}
 III &= 1091775,98/(1+0,2)^1 + 1755248,68/(1+0,2)^2 + 2418721,38/(1+0,2)^3 + \\
 &+ 2787317,33/(1+0,2)^4 = 909813,32 + 1218922,70 + 1399723,02 + 1344192,38 = 487265 \\
 &1,42 \text{ грн.}
 \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot ZB, \quad (6.25)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 2$;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 408728,17 грн.

$$PV = k_{инв} \cdot ZB = 2 \cdot 408728,17 = 817456,34 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = III - PV \quad (6.26)$$

де III – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 4872651,42 грн;

$$PV \text{ – теперішня вартість початкових інвестицій, } 817456,34 \text{ грн.}$$

$$E_{абс} = III - PV = 4872651,42 - 817456,34 = 4055195,08 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_{\epsilon} = T_{жс} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1, \quad (6.27)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 4055195,08 грн;

PV – теперішня вартість початкових інвестицій, 817456,34 грн;

$T_{жс}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_{\epsilon} = T_{жс} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 4055195,08/817456,34)^{1/4} - 1 = 0,56.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (6.28)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,09$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,28.

$\tau_{мін} = 0,09 + 0,28 = 0,37 < 0,56$ свідчить про те, що внутрішня економічна дохідність інвестицій E_{ϵ} , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Автоматизована система аналізу сигналів електрокардіограм» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (6.29)$$

де E_g – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,56 = 1,78 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Автоматизована система аналізу сигналів електрокардіограм» становить 39,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,89 рази.

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,15 рази.

Також термін окупності становить 1,78 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Автоматизована система аналізу сигналів електрокардіограм».

ВИСНОВКИ

У магістерській роботі було проаналізовано інформаційні потреби області застосування (виявлено, які дані та ознаки є вирішальними для визначення електрокардіографічних відхилень).

Критично проаналізовано існуючі методи аналізу та класифікації даних на прикладі визначення електрокардіографічних аномалій, представлених у формі часових рядів. На прикладі цих методів проаналізовано основні переваги та недоліки та визначено найкращі характеристики, що забезпечують підвищення точності алгоритму.

На основі аналізу існуючих методів запропоновано та реалізовано алгоритм зчитування та фільтрації даних, розроблено та реалізовано алгоритм аналізу пікових часових рядів (пошук локальних мінімумів та максимумів, класифікатор на основі алгоритму випадкового лісу) та запропоновано і реалізовано алгоритм аналізу часових рядів за значеннями інтервалів (перетворення інтервалів у речення, випадковий подвійний підйом координат).

Визначив технологічний стек і розробив архітектуру програмного забезпечення. Результатом розробки програмного забезпечення стало створення двох незалежних бібліотек для реалізації алгоритмів створення штучного інтелекту для класифікації часових рядів ЕКГ за піковими та інтервальними метриками відповідно. Програмне забезпечення сторонніх виробників може використовувати їх для аналізу даних ЕКГ.

Проаналізовано ефективність розробленого методу аналізу та класифікації даних (на прикладі визначення аномалій електрокардіограми), точність інтервальних та пікових показників підвищилась на 6% та 7% відповідно та порівняна з аналогічними методами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Циммерман Ф. Клиническая электрокардиография, 2013.– 424 с.
2. Мешков А.П. Азбука клинической электрокардиографии, 1998.– 149 с.
3. В.І. Дубровін Комп'ютерні методи інтелектуальної обробки даних– Запоріжжя ЗНТУ, 2013.– 105 с.
4. Волосатова Т.М., Малышев А.П. Улучшение сигнала электрокардиограммы на основе алгоритма удаления дрейфа его изолинии//Интернет журнал «НАУКОВЕДЕНИЕ», 2017.–9 с.
5. Зотов Д.Д. Современные методы функциональной диагностики в кардиологии–СПбГПМА, 2000.–51 с.
6. Волосатова Т. М., Малышев А. П. Автоматизированная система анализа и интерпретации электрокардиосигнала//Радиооптика МГТУ им. Н. Э. Баумана, 2016.–18 с.
7. M. Thill, S.Däubener, W.Konen, T.Bäck Anomaly Detection in Electrocardiogram Readings with Stacked LSTM Networks, 2016.– 9 с.
8. Дубровин В. И., Твердохлеб Ю. В., Харченко В. В. Автоматизированная система анализа и интерпретации ЭКГ, 2014. – 8с.
9. Мустафаев А.Г., Темирбулатов М.А., Омаров Р.С. Определение аномалий сердечного ритма и выявление заболевания сердца при помощи нейронных сетей, 2017.– 11с.
10. J. Pereira, M. Silveira Unsupervised Representation Learning and Anomaly Detection in ECG Sequences, 2017 – 18 с.
11. Волосатова Т.М., Спасенов А.Ю., Логунова А.О. Автоматическая система анализа и интерпретации кардиосигнала, 2016 – 18 с.
12. MIT-BIH Arrhythmia Database, [Электронный ресурс]. – Режим доступа: www.physionet.org/content/mitdb/1.0.0/
13. Кіншаков Е.В. Моделювання та прогнозування великих наборів даних засобами машинного навчання, 2020. – 72с.

14. S. Shalev-Shwartz, T. Zhang Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization, 2013. – 33с.
15. Что такое ML.NET и принципы работы этой системы, [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/machine-learning/how-does-mldotnet-work>
16. WindowsForms и интеллектуальные клиентские приложения, [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/windows-forms-overview?view=netframeworkdesktop-4.8>
17. Мішура К.А. Виділення складного тренду сигналів на основі масштабно- часових перетворень, 2020.– 129 с.
18. I. Baklan, A. Oliynyk, I. Mukha, K. Lishchuk, O. Gavrilenko, S. Reutska, A. Tsitsyliuk, Y. Oliynyk ECG signal processing based on linguistic chain fuzzy set // Computational Linguistics and Intelligent Systems (CoLins2021), 2021. – 11с.
19. В. О. Козловський О. Й. Лесько, В. В. Кавецький Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. – Вінниця: ВНТУ. – 2021. – 42 с.
20. В. В. Кавецький, В. О. Козловський, І. В. Причепя Економічне обґрунтування інноваційних рішень: [практикум] – Вінниця : ВНТУ. – 2016. – 113 с.
21. Glowinski D. Toward a minimal representation of affective gestures / D. Glowinski et al. // IEEE Trans. Affect. Comput. – 2011. – vol. 2, no. 2. – P. 106–118.
22. Castellano G. Recognising human emotions from body movement and gesture dynamics / G. Castellano, S.D. Villalba, A. Camurri // Affect. Comput. Intell. Interact. – 2007. – vol. 4738, no. 1. – P. 71–82.
23. Camurri A. Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques / A. Camurri, I. Lagerlog, G. Volpe // Int. J. Hum. Comput. Stud. – 2003. – vol. 59, no. 1. – P. 213–225.

24. Singh G. Negative emotion recognition from stimulated EEG signals / G. Singh, A. Jati, A. Khasnobish // Third International Conference on Computing Communication & Networking Technologies. – 2012.
25. Petrantonakis P.C. Emotion recognition from brain signals using hybrid adaptive filtering and higher order crossings analysis / P.C. Petrantonakis, L.J. Hadjileontiadis // IEEE Trans. Affect. Comput. – 2010. – vol. 1, no. 2. – P. 81–97.
26. Lin Y.P. EEG-based emotion recognition in music listening / Y.P. Lin, C.H. Wang, T.P. Jung // IEEE Trans. Biomed. Eng. – 2010. – vol. 57, no. 7. – P. 1798–1806.
27. Mehrabian A. Communication without words / A. Mehrabian // Psychol. Today. – 1968. – vol. 2, no. 4. – P. 53–56.
28. Analysis and Classification of ECG Signals using Ant Lion Optimized Neural Network; by G. Rajasekaran et al. (2017).
29. A Review on Cardiac Arrhythmia Classification using ECG Signals by Ajit Danti et al. (2020).
30. Automated ECG Classification for the Detection of Arrhythmias by Mohamed Elgendi (2019).
31. A novel automatic detection of R-peaks in electrocardiogram (ECG) signal by Sharma, D., Kumar, S., Arya, K. V. (2011).
32. Bioinformatics Challenges at the Interface of Biology and Computer Science: Mind the Gap by Pierre Baldi, Søren Brunak (2012).
33. Real-time QRS detection using combined adaptive threshold by Pan, J., Tompkins, W. J. (1985).
34. An Efficient Approach for QRS Detection of ECG Signal Using Empirical Mode Decomposition by Iyer, S. R., Manikandan, M. S., Vimal, C. (2010).
35. Handbook of Research on Advanced Hybrid Intelligent Techniques and Applications by Siddhartha Bhattacharyya, Susanta Chakraborty (2010).
36. Heartbeat classification using abstract features from the ECG signal by Melgani, F., Bazi, Y. (2008).

37. An Algorithm for Heart Sounds Segmentation Using Wavelet Transform by Acharya, U. R., Lim, C. M. (2011).
38. Intelligent Data Analysis in Medicine and Pharmacology by Yves Moreau (2013).
39. QRS complex detection using multiresolution analysis of electrocardiogram signals by Acharya, U. R., Lim, C. M. (2002).
40. An automatic method for ECG feature extraction and classification by Jiang, W., Kong, S. G., Lim, J. S. (2005).
41. Automatic Classification of Heartbeats using ECG Morphology and Heartbeat Interval Features by Adarsh Rajendran and Umamaheswari K. (2019).
42. Advanced Signal Processing for ECG Analysis by Samuel E. Schmidt, Gari D. Clifford (2017).
43. Classification of Electrocardiogram Signals with Support Vector Machines and Particle Swarm Optimization by Muhammad Usama and Waqas Anwar (2018).
44. A Novel Method for Detection of R Peaks in ECG Signals Using Kurtosis and Entropy by Aayush Dogra and A. K. Sahoo (2017).
45. Electrocardiography for the Family Physician: The Essentials by Faezeh Marzban-Richardson (2013).
46. Detection of Ventricular Arrhythmias Using a Novel Morphology-based ECG Complexity Measure by Mohamed Elgendi (2018).
47. QRS Detection using Wavelet Transform and Adaptive Threshold for Automated ECG Analysis by C. Sivagnanam and R. Ramakrishanan (2010).
48. Electrocardiography in Ischemic Heart Disease: Clinical and Imaging Correlations and Prognostic Implications by Hein J. Wellens and Milos Taborsky (2008).
49. ECG Feature Extraction Using Wavelet Transform and PCA for Ischemic Heart Disease Classification by B. S. Chakravarthi and M. S. Manikandan (2014).

50. A Review of Wavelet Transform Methodologies in Heart Rate Variability Studies by Kusum Deep, Millie Pant, and Jagdish Chand Bansal (2009).

51. Detection and Classification of Heart Abnormalities Using ECG Signal Processing and Neural Networks by Alaa Tharwat, Mohamed Gaber, and Aboul Ella Hassanien (2016).

ДОДАТКИ

ДОДАТОК А
(обов'язковий)

**ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: «Автоматизована система аналізу сигналів електрокардіограм»

Тип роботи: Магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ КСУ, ФШТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 91,7% Схожість 8,3%


Аналіз звіту подібності (відмітити потрібне)

✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.

Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку


(підпис)

Володимир ДУБОВОЙ
(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи

_____ (підпис)

Євгеній СЛОБОДЯНЮК

_____ (прізвище, ініціали)

Керівник роботи

_____ (підпис)

Олена НИКИТЕНКО


_____ (прізвище, ініціали)

ДОДАТОК Б
(Обов'язковий)

Додаток Б
(обов'язковий)
ВНТУ

ЗАТВЕРДЖЕНО

Т.в.о. зав. кафедри КСУ ВНТУ,
д.т.н., доцент


Марія ЮХИМЧУК

“ 6 ” жовтня 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

Автоматизована система аналізу сигналів електрокардіограм

08-33.МКР.15.00.000 ТЗ

Студент групи 2АКІТ-22м



Євгеній СЛОБОДЯНЮК

Керівник к.т.н., доцент кафедри КСУ



Олена НИКИТЕНКО

1. Назва та галузь застосування

1.1. Назва – Автоматизована система аналізу сигналів електрокардіограм

1.2. Галузь застосування – медичні технології

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ від №247 від 18-09-2023р.

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є в розробка математичного забезпечення для аналізу ЕКГ сигналів та підбору найкращого алгоритму машинного навчання для класифікації цих даних, інтеграції розробленого алгоритму в відповідне програмне забезпечення, що дасть можливість підвищити точність та універсальність алгоритмів розпізнавання хвороб у пацієнта по ЕКГ.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. "A Review on ECG Signal Processing Techniques" by Anurag Goel and D. K. Jain.
2. Computing in Cardiology Conference Proceedings
3. IEEE Transactions on Biomedical Engineering
4. "A Survey of Deep Learning Techniques for the Analysis of Bio-medical Signals and Images" by U. R. Acharya et al. (2018).

5. Стадії та етапи розробки.

5.1 Пояснювальна записка:

- дослідити та порівняти наявні алгоритми аналізу тарозділення часових рядів ЕКГ на набір параметрів;

- дослідити та порівняти наявні підходи до класифікації часових рядів ЕКГ на випадок хвороб;
- створити власні алгоритми розділення, аналізу та класифікації часових рядів ЕКГ, що будуть працювати з прийнятною точністю та охоплювати різні випадки аномалій в синусоїді;
- реалізувати запропоновані алгоритми у вигляді незалежних бібліотек;
- розробка програмного забезпечення для розпізнавання хвороб у пацієнта по ЕКГ;
- дослідити розроблені алгоритми на ефективність роботи.

6. Порядок контролю і приймання.

- 6.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до « 30 » 11 2023 р.
- 6.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до « 6 » 12 2023 р.
- 6.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до « 20 » 12 2023 р.

ДОДАТОК В

(довідковий)

Фрагмент лістингу програми

```

publicclassPicksAnalizator
{
    constchar delimiter = '\t';

    DataViewSchemamodelSchema; ITransformertrainedModel; MLContextmlContext =
newMLContext(seed: 4);

    publicPicksAnalizator(string disease)
    {
        trainedModel = mlContext.Model.Load($"C:\\{disease}_Picks.zip", outmodelSchema);
    }

    publicKeyValuePair<string, List<double>>Analyze(List<ECGItem>detectedIntervals,
doubleRRFrequency, doubleAvarageRRFrequencyTweak)
    {
        List<KeyValuePair<int, double>>problemsPeaks = new List<KeyValuePair<int, double>>();

        for (inti = 0; i<detectedIntervals.Count; i++)
        {
            ECGSegment test = newECGSegment()
            {
                PQDuration =
Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].PQ.DurationTime, 2) * 100)),
                QSDuration =
Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].QS.DurationTime, 2) * 100)),
                STDuration =
Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].ST.DurationTime, 2) * 100)),
                RRFrequency = Convert.ToSingle(Math.Floor(Math.Round(RRFrequency, 2) * 100)),
            }
        }
    }
}

```



```

    AvarageRRFrequencyTweak =
Convert.ToSingle(Math.Floor(Math.Round(AvarageRRFrequencyTweak, 2) * 100)),
    PTime = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].P.Time, 2) *
100)),
    Pvalue = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].P.PicValue, 2) *
100)),
    QTime = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].Q.Time, 2) *
100)),
    Qvalue = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].Q.PicValue, 2) *
100)),
    RTime = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].R .Time, 2) *
100)),
    Rvalue = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].R.PicValue, 2) *
100)),
    STime = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].S.Time, 2) *
100)),
    Svalue = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].S.PicValue, 2) *
100)),
    TTime = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].T.Time, 2) *
100)),
    Tvalue = Convert.ToSingle(Math.Floor(Math.Round(detectedIntervals[i].T.PicValue, 2) *
100)),
};
    varpredEngine = mlContext.Model.CreatePredictionEngine<ECGSegment,
ECGPeakPrediction>(trainedModel); varresultprediction = predEngine.Predict(test);

    if (resultprediction.Score > 0.6) problemsPeaks.Add( newKeyValuePair<int, double>(i,
resultprediction.Score));
}
    problemsPeaks.Sort((pair1, pair2) =>pair2.Value.CompareTo(pair1.Value));

    stringresultString;
    List<double>resultInterval = null;

```

```

if (problemsPeaks.Count == 0) resultString = "Everything is good"; else
{
    resultString = $"Worst interval was with {Math.Round(problemsPeaks[0].Value, 2)}%
illness. \n\n" +

        $"In 90% confidence - {problemsPeaks.Where(x =>x.Value>= 0.9).Count()} intervals, \n" +
        $"In 80% confidence - {problemsPeaks.Where(x =>x.Value< 0.9 && x.Value>=
0.8).Count()} intervals, \n" +
        $"In 70% confidence - {problemsPeaks.Where(x =>x.Value< 0.8 && x.Value>=
0.7).Count()} intervals, \n" +
        $"In 60% confidence - {problemsPeaks.Where(x =>x.Value< 0.7 && x.Value>=
0.6).Count()} intervals. \n";
    resultInterval =
detectedIntervals[problemsPeaks[0].Key].FullInterval;
}

return new KeyValuePair<string, List<double>>(resultString, resultInterval);
}

public class IntervalsAnalizator
{
    const char delimiter = '\t';

    ITransformer trainedModelPQ;
    ITransformer trainedModelQS;
    ITransformer trainedModelST;
    DataViewSchema modelSchemaPQ;
    DataViewSchema modelSchemaQS;
    DataViewSchema modelSchemaST;
    MLContext mlContextPQ = new MLContext(seed: 1);
    MLContext mlContextQS = new MLContext(seed: 2);
    MLContext mlContextST = new MLContext(seed: 3);

    double PQCoef;
    double QSCoef;
    double STCoef;

```

```

publicIntervalsAnalizator(string disease)
{
    trainedModelPQ          =          mlContextPQ.Model.Load($"C:\\PQ_{disease}.zip",
outmodelSchemaPQ); trainedModelQS = mlContextQS.Model.Load($"C:\\QS_{disease}.zip",
outmodelSchemaQS); trainedModelST = mlContextST.Model.Load($"C:\\ST_{disease}.zip",
outmodelSchemaST);

    var Lines_1 = File.ReadAllLines($"C:\\Words_ROC_values.tsv").ToList();
    PQCoef = Math.Round(Convert.ToDouble(Lines_1.Where(x =>x.Split(delimiter)[0] ==
    $"{disease}"&&x.Split(delimiter)[1] == "PQ").First().Split(delimiter)[2]), 3); QSCoef =
Math.Round(Convert.ToDouble(Lines_1.Where(x =>x.Split(delimiter)[0] ==
    $"{disease}"&&x.Split(delimiter)[1] == "QS").First().Split(delimiter)[2]), 3); STCoef =
Math.Round(Convert.ToDouble(Lines_1.Where(x =>x.Split(delimiter)[0] ==
    $"{disease}"&&x.Split(delimiter)[1] == "ST").First().Split(delimiter)[2]), 3); varsumCoef =
PQCoef + QSCoef + STCoef;
    PQCoef = PQCoef / sumCoef;
    QSCoef = QSCoef / sumCoef; STCoef = STCoef / sumCoef;
}

publicKeyValuePair<string, List<double>>Analyze(List<ECGItem>detectedIntervals)
{

double>>());

List<KeyValuePair<int, double>>problemsIntervals = new List<KeyValuePair<int,

for (inti = 0; i<detectedIntervals.Count; i++)
{
    ECGWordtestPQ = newECGWord() { Text = detectedIntervals[i].PQ.Word };
    varpredEnginePQ          =          mlContextPQ.Model.CreatePredictionEngine<ECGWord,
ECGPrediction>(trainedModelPQ); varresultpredictionPQ = predEnginePQ.Predict(testPQ);

```

```

ECGWordtestQS = newECGWord() { Text = detectedIntervals[i].QS.Word };

varpredEngineQS      =      mlContextQS.Model.CreatePredictionEngine<ECGWord,
ECGPrediction>(trainedModelQS); varresultpredictionQS = predEngineQS.Predict(testQS);

ECGWordtestST = newECGWord() { Text = detectedIntervals[i].ST.Word };
varpredEngineST      =      mlContextST.Model.CreatePredictionEngine<ECGWord,
ECGPrediction>(trainedModelST); varresultpredictionST = predEngineST.Predict(testST);

var prediction = resultpredictionPQ.Probability * PQCoef + resultpredictionQS.Probability
* QSCoef + resultpredictionST.Probability * STCoef;

if (prediction > 0.6) problemsIntervals.Add( newKeyValuePair<int, double>(i, prediction));
}
problemsIntervals.Sort((pair1, pair2) =>pair2.Value.CompareTo(pair1.Value));
stringresultString;
List<double>resultInterval = null;

if (problemsIntervals.Count == 0) resultString = "Everything is good"; else
{
resultString = $"Worst interval was with {Math.Round(problemsIntervals[0].Value, 2)}%
illness. \n\n"
+
$"In 90% confidence - {problemsIntervals.Where(x =>x.Value>= 0.9).Count()} intervals,
\n" +
$"In 80% confidence - {problemsIntervals.Where(x =>x.Value< 0.9 && x.Value>=
0.8).Count()} intervals,
\n" +
$"In 70% confidence - {problemsIntervals.Where(x =>x.Value< 0.8 && x.Value>=
0.7).Count()} intervals,
\n" +

```

```
$"In 60% confidence - {problemsIntervals.Where(x =>x.Value< 0.7 && x.Value>=
0.6).Count()} intervals.
```

```
\n";
```

```
resultInterval = detectedIntervals[problemsIntervals[0].Key].FullInterval;
```

```
}
```

```
return new KeyValuePair<string, List<double>>(resultString, resultInterval);
```

```
}
```

```
public static class FileManager
```

```
{
```

```
    static string FILES_FOLDER_PATH = "C:\\\\"; static string DELIMETR = "\\t";
```

```
    public static void MixDatasets(string name1, string name2)
```

```
{
```

```
    var Lines_1 = File.ReadAllLines($"{FILES_FOLDER_PATH}{name1}.tsv").ToList(); var
    Lines_2 = File.ReadAllLines($"{FILES_FOLDER_PATH}{name2}.tsv").ToList();
```

```
    var header = Lines_1[0];
```

```
    Lines_2.RemoveAt(0); Lines_1.RemoveAt(0);
```

```
    List<string> LinesAll = new List<string>(Lines_1); LinesAll.AddRange(Lines_2);
```

```
    Random random = new Random();
```

```
    LinesAll = LinesAll.OrderBy(x => random.Next()).ToList();
```

```
    var result = new StringBuilder(); result.AppendLine(header); foreach (var item in LinesAll)
```

```
{
```

```
    if (item.Length > 5) result.AppendLine(item);
```

```
}
```

```

File.WriteAllText($"{FILES_FOLDER_PATH}{name2}_withGood.tsv", result.ToString());
}

publicstaticvoidSaveToIntervalsDataset(string name, intisWrong, string disease,
List<ECGItem>detectedIntervals)
{
varcsvWordsPQ = newStringBuilder(); varcsvWordsQS = newStringBuilder();
varcsvWordsST = newStringBuilder();

csvWordsPQ.AppendLine($"Label{DELIMETR}Text"); foreach (var item
indetectedIntervals)
{
if (item.PQ.Word.Length> 0)
{
varnewPQParamsLine = $"{isWrong}{DELIMETR}{item.PQ.Word}";
csvWordsPQ.AppendLine(newPQParamsLine);
}
if (item.QS.Word.Length> 0)
{
varnewQSPParamsLine = $"{isWrong}{DELIMETR}{item.QS.Word}";
csvWordsQS.AppendLine(newQSPParamsLine);
}
if (item.ST.Word.Length> 0)
{
varnewSTParamsLine = $"{isWrong}{DELIMETR}{item.ST.Word}";
csvWordsST.AppendLine(newSTParamsLine);
}
}

File.WriteAllText($"{FILES_FOLDER_PATH}Words_{name}_PQ_{disease}.tsv",
csvWordsPQ.ToString());

```

```

File.WriteAllText($"{FILES_FOLDER_PATH}Words_{name}_QS_{disease}.tsv",
csvWordsQS.ToString());

File.WriteAllText($"{FILES_FOLDER_PATH}Words_{name}_ST_{disease}.tsv",
csvWordsST.ToString());

}

publicstaticvoidSaveToPicksDataset(string name, intisWrong, string disease,
List<ECGItem>detectedIntervals, doubleRRFrequency, doubleAvarageRRFrequencyTweak)
{
varcsvParams = newStringBuilder();

csvParams.AppendLine($"Label" +
${DELIMETR}PTime{DELIMETR}Pvalue{DELIMETR}" +
$QTime{DELIMETR}Qvalue{DELIMETR}" +
$RTime{DELIMETR}Rvalue{DELIMETR}" +
$STime{DELIMETR}Svalue{DELIMETR}" +
$TTime{DELIMETR}Tvalue{DELIMETR}" +
$PQDuration{DELIMETR}QSDuration{DELIMETR}STDuration" +
${DELIMETR}RRFrequency{DELIMETR}AvarageRRFrequencyTweak");

foreach (var item indetectedIntervals)
{
varnewcsvParamsLine = $"{isWrong}" +
${DELIMETR}{Math.Floor(Math.Round(item.P.Time, 2) * 100)}" +

${DELIMETR}{Math.Floor(Math.Round(item.P.PicValue, 2) * 100)}" +
${DELIMETR}{Math.Floor(Math.Round(item.Q.Time, 2) * 100)}" +
${DELIMETR}{Math.Floor(Math.Round(item.Q.PicValue, 2) * 100)}" +
${DELIMETR}{Math.Floor(Math.Round(item.R.Time, 2) * 100)}" +
${DELIMETR}{Math.Floor(Math.Round(item.R.PicValue, 2) * 100)}" +
${DELIMETR}{Math.Floor(Math.Round(item.S.Time, 2) * 100)}" +
${DELIMETR}{Math.Floor(Math.Round(item.S.PicValue, 2) * 100)}" +

```

```

    $"{DELIMETR}{Math.Floor(Math.Round(item.T.Time, 2) * 100)}" +
    $"{DELIMETR}{Math.Floor(Math.Round(item.T.PicValue, 2) * 100)}" +
    $"{DELIMETR}{Math.Floor(Math.Round(item.PQ.DurationTime, 2) * 100)}" +
    $"{DELIMETR}{Math.Floor(Math.Round(item.QS.DurationTime, 2) * 100)}" +
    $"{DELIMETR}{Math.Floor(Math.Round(item.ST.DurationTime, 2) * 100)}" +
    $"{DELIMETR}{Math.Floor(Math.Round(RRFrequency, 2) * 100)}" +
    $"{DELIMETR}{Math.Floor(Math.Round(AvarageRRFrequencyTweak, 2) * 100)}";
csvParams.AppendLine(newcsvParamsLine);
    }

    File.WriteAllText($"{FILES_FOLDER_PATH}Pics_{name}_{disease}.tsv",
csvParams.ToString());
    }

publicclassECGConverter
{
    privateconstint ECG_DOWN_COEF = 950; List<List<double>>hardToDetectIntervals;

    public List<ECGItem>detectedIntervals; publicdoubleRRFrequency;
publicdoubleAvarageRRFrequencyTweak;

    publicECGConverter(List<double>FullECG, List<double>RRTime, List<int>RIds,
doubleAVGIntervalTime)
    {
        detectedIntervals = new List<ECGItem>(); hardToDetectIntervals = new
List<List<double>>();

        RRFrequency = 0;
        AvarageRRFrequencyTweak = 0;

        ConvertToECGCollection(FullECG, RRTime, RIds, AVGIntervalTime);
    }
}

```



```

publicWaveletTransformationWaveletTransformation
{
    get =>default; set
    {
    }
}

publicvoidConvertToECGCollection(List<double>FullECG,           List<double>RRTime,
List<int>RIds, doubleAVGIntervalTime)
{

    for (inti = 1; i<RIds.Count - 1; i++)
    {
        intprevId = RIds[i - 1]; intcurId = RIds[i]; intnextId = RIds[i + 1];

        intSId = 0, QId = 0, PId = 0, RId = 0, TId = 0;
        doubleSVal = 0, QVal = 0, PVal = 0, RVal = 0, TVal = 0;

        doubleleftTime = RRTime[i - 1]; doublerightTime = RRTime[i];

        boolleftIsTooSmall = (leftTime< 0.75 * AVGIntervalTime); boolleftIsTooBig = (leftTime>
1.25 * AVGIntervalTime);

        if (leftIsTooBig || leftIsTooSmall)
        {
            hardToDetectIntervals.Add(FullECG.GetRange(prevId, curId - prevId)); if (leftIsTooSmall)
continue;

            leftTime = AVGIntervalTime;
        }
    }
}

```

```

boolrightIsTooSmall = (rightTime < 0.75 * AVGIntervalTime); boolrightIsTooBig =
(rightTime > 1.25 * AVGIntervalTime);

```

```

if (rightIsTooSmall || rightIsTooBig)
{
if (rightIsTooSmall) continue; rightTime = AVGIntervalTime;
}

```

```

RRFrequency += leftTime;
AvarageRRFrequencyTweak += Math.Abs(leftTime - AVGIntervalTime) * 10;

```

```

doubleRSign = FullECG[curId]; ECGItem item = newECGItem(); item.P = newECGPoint();
item.Q = newECGPoint(); item.R = newECGPoint(); item.S = newECGPoint(); item.T =
newECGPoint();

```

```

intsearchId = SId; intupCount = 0;

```

```

if (RSign < FullECG[curId + 2] || RSign < FullECG[curId - 2])
{
}

```

```

if (RId == 0)
{
RId = curId;
RVal = FullECG[curId];
}

```

```

if (QId == 0)
{
#region FindQ

```

```

QId = RId;

```

```
QVal = FullECG[RId]; searchId = RId; upCount = 0;
```

```
while (upCount < 2)
```

```
{
```

```
searchId--;
```

```
if (FullECG[searchId] > QVal) upCount++; else
```

```
{
```

```
QVal = FullECG[searchId]; upCount = 0;
```

```
}
```

```
}
```

```
QId = searchId + 2;
```

```
#endregion
```

```
}
```

```
if (SId == 0)
```

```
{
```

```
#region FindS
```

```
SId = curId;
```

```
SVal = FullECG[curId]; searchId = curId; upCount = 0;
```

```
while (upCount < 2)
```

```
{
```

```
searchId++;
```

```
if (FullECG[searchId] > SVal) upCount++; else
```

```
{
```

```
SVal = FullECG[searchId]; upCount = 0;
```

```
}
```

```
}
```

```
SId = searchId - 2;
```

```
#endregion
}
if (TId == 0)
{
#regionFindT

TId = SId;
TVal = FullECG[SId]; doubleTValPrev = FullECG[SId]; searchId = SId;
upCount = 0;

while (upCount < 5)
{
searchId++;

if (FullECG[searchId] < TValPrev)
{
upCount++;
}
else
{
TVal = FullECG[searchId]; upCount = 0;

}

TValPrev = FullECG[searchId];
}
TId = searchId - 5;
#endregion
}
if (Pid == 0)
{
#regionFindP
```

```
PId = QId;
PVal = FullECG[QId]; doublePValPrev = FullECG[QId]; searchId = QId;
upCount = 0;

while (upCount < 3)
{
searchId--;

if (FullECG[searchId] < PValPrev)
{
upCount++;
}
else
{
PVal = FullECG[searchId]; upCount = 0;
}

PValPrev = FullECG[searchId];
}

PId = searchId - 3;
#endregion
}

item.P.PicValue = 0;
item.Q.PicValue = 0;
item.R.PicValue = 0;
item.S.PicValue = 0;
item.T.PicValue = 0;

item.P.Time = 0;
item.Q.Time = 0;
item.R.Time = 0;
item.S.Time = 0;
```

```
item.T.Time = 0;
```

```
item.P.PicValue = PVal - ECG_DOWN_COEF; item.Q.PicValue = QVal - ECG_DOWN_COEF;
item.R.PicValue = RVal - ECG_DOWN_COEF; item.S.PicValue = SVal - ECG_DOWN_COEF;
item.T.PicValue = TVal - ECG_DOWN_COEF;
```

```
item.P.Time = (RId - PId) * 0.002; item.Q.Time = (RId - QId) * 0.002; item.R.Time = 0;
item.S.Time = (SId - RId) * 0.002; item.T.Time = (TId - RId) * 0.002;
```

```
item.PQ = newECGInterval(); item.QS = newECGInterval();
item.ST = newECGInterval();
```

```
item.FullInterval = FullECG.GetRange(prevId, nextId - prevId);
```

```
intPQBegin = PId - Convert.ToInt32((QId - PId) / 2); doublePQTime = (QId - PQBegin) * 0.2;
stringPQWord = "";
StringBuilder PQSb = newStringBuilder();
```

```
for (int s = PQBegin; s <= QId; s += 2)
```

```
{
```

```
int diff = Convert.ToInt32(FullECG[s]) - Convert.ToInt32(FullECG[s + 1]);
```

```
PQSb.Append(MakeAlfaFromDiff(diff));
```

```
}
```

```
if (PQSb.Length > 3)
```

```
{
```

```
for (int s = 0; s <= PQSb.Length - 3; s++)
```

```
{
```

```
PQWord += PQSb[s].ToString() + PQSb[s + 1].ToString() + PQSb[s + 2].ToString() + " ";
```

```
if (PQSb.Length > 4 && (s + 3 < PQSb.Length))
```

```

PQWord += PQSb[s].ToString() + PQSb[s + 1].ToString() + PQSb[s + 2].ToString() + PQSb[s
+ 3].ToString() + " ";
    if (PQSb.Length > 4 && (s + 4 < PQSb.Length))
        PQWord += PQSb[s].ToString() + PQSb[s + 1].ToString() + PQSb[s + 2].ToString() + PQSb[s
+ 3].ToString() + PQSb[s + 4].ToString() + " ";
    }
}
PQWord += PQSb.ToString();

intQSEnd = SId + Convert.ToInt32((SId - QId) / 3); intQSBegin = QId - Convert.ToInt32((SId
- QId) / 3); doubleQSTime = (QSEnd - QSBegin) * 0.2; stringQSWord = "";
StringBuilder QSSb = newStringBuilder();

for (int s = QSBegin; s <= QSEnd; s += 2)
{
    int diff = Convert.ToInt32(FullECG[s]) - Convert.ToInt32(FullECG[s + 1]);

    QSSb.Append(MakeAlfaFromDiff(diff));
}
if (QSSb.Length > 3)
{
    for (int s = 0; s <= QSSb.Length - 3; s++)
    {
        QSWord += QSSb[s].ToString() + QSSb[s + 1].ToString() + QSSb[s + 2].ToString() + " ";

        if (QSSb.Length > 4 && (s + 3 < QSSb.Length))
            QSWord += QSSb[s].ToString() + QSSb[s + 1].ToString() + QSSb[s + 2].ToString() + QSSb[s +
3].ToString() + " ";

        if (QSSb.Length > 4 && (s + 4 < QSSb.Length))
            QSWord += QSSb[s].ToString() + QSSb[s + 1].ToString() + QSSb[s + 2].ToString() + QSSb[s +
3].ToString() + QSSb[s + 4].ToString() + " ";
    }
}
}

```

```

QSTWord += QSSb.ToString();

intSTBegin = QSEnd;
intSTEnd = TId + Convert.ToInt32((TId - STBegin) / 2); doubleSTTime = (STEnd - STBegin) *
0.2;
stringSTWord = "";
StringBuilder STSb = newStringBuilder();

for (int s = STBegin; s <= STEnd; s += 2)
{
int diff = Convert.ToInt32(FullECG[s]) - Convert.ToInt32(FullECG[s + 1]);

STSb.Append(MakeAlfaFromDiff(diff));
}
if (STSb.Length > 3)
{
for (int s = 0; s <= STSb.Length - 3; s++)
{
STWord += STSb[s].ToString() + STSb[s + 1].ToString() + STSb[s + 2].ToString() + " ";

if (STSb.Length > 4 && (s + 3 < STSb.Length))
STWord += STSb[s].ToString() + STSb[s + 1].ToString() + STSb[s + 2].ToString() + STSb[s +
3].ToString() + " ";
if (STSb.Length > 4 && (s + 4 < STSb.Length))
STWord += STSb[s].ToString() + STSb[s + 1].ToString() + STSb[s + 2].ToString() + STSb[s +
3].ToString() + STSb[s + 4].ToString() + " ";
}
}
STWord += STSb.ToString();

item.PQ.DurationTime = 0;

```



```

item.QS.DurationTime = 0;
item.ST.DurationTime = 0;

item.PQ.DurationTime = PQTime; item.PQ.Word = PQWord;

item.QS.DurationTime = QSTime; item.QS.Word = QSWord;

item.ST.DurationTime = STTime; item.ST.Word = STWord;

detectedIntervals.Add(item);
}

AvarageRRFrequencyTweak = AvarageRRFrequencyTweak / detectedIntervals.Count;
RRFrequency = 10 / (RRFrequency / detectedIntervals.Count);
}
publicstringMakeAlfaFromDiff(int diff)
{
string res = "";

if (diff == 0) res = "A"; if (diff > 0)
{
if(diff < 3) res = "B"; elseif (diff < 6) res = "C"; elseif (diff < 10) res = "D"; elseif (diff < 16)
res = "E";

elseif (diff < 26) res = "F"; elseif (diff < 36) res = "G"; elseif (diff < 46) res = "H"; elseif (diff
< 56) res = "I"; elseif (diff < 86) res = "J"; elseif (diff < 110) res = "K"; elseif (diff < 150) res = "L";
elseif (diff < 200) res = "M"; else res = "N";
}
if (diff < 0)
{
if (diff > -3) res = "Z"; elseif (diff > -6) res = "Y"; elseif (diff > -9) res = "X"; elseif (diff > -16)
res = "W"; elseif (diff > -26) res = "V"; elseif (diff > -36) res = "U"; elseif (diff > -56) res = "T"; elseif

```

```

(diff > -86) res = "S"; elseif (diff > -110) res = "R"; elseif (diff > -150) res = "Q"; elseif (diff > -200)
res = "P"; else res = "O";
    }

    return res;
}

public List<KeyValuePair<string, List<double>>>Analyze(string disease)
{
    PicksAnalizatorpicksAnalizator = newPicksAnalizator(disease);
    IntervalsAnalizatorintervalsAnalizator = newIntervalsAnalizator(disease);

    List<KeyValuePair<string, List<double>>> returns = new List<KeyValuePair<string,
List<double>>>()
    {
        intervalsAnalizator.Analyze(detectedIntervals), picksAnalizator.Analyze(detectedIntervals,
RRFrequency, AvarageRRFrequencyTweak)
    };

    return returns;
}

classProgram
{
    conststring delimiter = "\t";

    staticvoid Main(string[] args)
    {
        StringBuilder sb = newStringBuilder();
        sb.AppendLine($"Disease{delimiter}Interval{delimiter}ROC");
    }
}

```

```

sb.AppendLine(TrainECG("gypotermia")); sb.AppendLine(TrainECG("infarktMiocarda"));
sb.AppendLine(TrainECG("synoatrialBlock")); sb.AppendLine(TrainECG("trepetPeredserd"));
}

```

```

publicstaticstringTrainECG(stringdesease)
{
varmlContext = newMLContext(seed: 0);

IDataViewdataView =
mlContext.Data.LoadFromTextFile<ECGSegment>($"C:\\Pics_ECGDataset_{desease}_withGood.
tsv", hasHeader: true);

TrainTestDatatrainTestSplit = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);
IDataViewtrainingData = trainTestSplit.TrainSet;
IDataViewtestData = trainTestSplit.TestSet;

vardataProcessPipeline = mlContext.Transforms.Concatenate("Features",
nameof(ECGSegment.PTime),
nameof(ECGSegment.Pvalue), nameof(ECGSegment.QTime),
nameof(ECGSegment.Qvalue), nameof(ECGSegment.Rvalue), nameof(ECGSegment.STime),
nameof(ECGSegment.Svalue), nameof(ECGSegment.TTime), nameof(ECGSegment.Tvalue),
nameof(ECGSegment.PQDuration), nameof(ECGSegment.QSDuration),
nameof(ECGSegment.STDuration), nameof(ECGSegment.RRFrequency),
nameof(ECGSegment.AvarageRRFrequencyTweak)
)
.AppendCacheCheckpoint(mlContext);

var trainer = mlContext.Regression.Trainers. FastForest(labelColumnName: "Label",
featureColumnName: "Features", numberOfLeaves: 7, numberOfTrees: 14,
minimumExampleCountPerLeaf: 20); vartrainingPipeline = dataProcessPipeline.Append(trainer);
ITransformertrainedModel = trainingPipeline.Fit(dataView);

```

```

var predictions = trainedModel.Transform(testData);
var metrics = mlContext.Regression.Evaluate(data: predictions, labelColumnName:
"Label", scoreColumnName: "Score");

mlContext.Model.Save(trainedModel, trainingData.Schema, $"C:\\{desease}_Picks.zip");

Console.WriteLine($"\\n\\n\\n===== ML FOR :{desease}
=====\\n"); Console.WriteLine($"RSquared: {metrics.RSquared}\\n");
Console.WriteLine($"LossFunction: {metrics.LossFunction}");
Console.WriteLine($"MeanAbsoluteError: {metrics.MeanAbsoluteError}");
Console.WriteLine($"MeanSquaredError: {metrics.MeanSquaredError}");
Console.WriteLine($"RootMeanSquaredError: {metrics.RootMeanSquaredError}");

return $"{desease}{delimiter}{metrics.MeanSquaredError}";
}
}

class Program
{
    const string delimiter = "\\t"; static void Main(string[] args)

    {
        StringBuilder sb = new StringBuilder();
        sb.AppendLine($"Disease{delimiter}Interval{delimiter}ROC");

        sb.AppendLine(TrainECG("PQ_gypotermia"));
        sb.AppendLine(TrainECG("QS_gypotermia")); sb.AppendLine(TrainECG("ST_gypotermia"));
    }
}

```

```

        sb.AppendLine(TrainECG("PQ_infarktMiocarda"));
sb.AppendLine(TrainECG("QS_infarktMiocarda"));
sb.AppendLine(TrainECG("ST_infarktMiocarda"));

        sb.AppendLine(TrainECG("PQ_synoatrialBlock"));
sb.AppendLine(TrainECG("QS_synoatrialBlock"));
sb.AppendLine(TrainECG("ST_synoatrialBlock"));

        sb.AppendLine(TrainECG("PQ_trepetPeredserd"));
sb.AppendLine(TrainECG("QS_trepetPeredserd"));
sb.AppendLine(TrainECG("ST_trepetPeredserd"));

File.WriteAllText($"C:\\Words_ROC_values.tsv", sb.ToString());
}

staticstringTrainECG(stringdesease)
{
    varmlContext = newMLContext(seed: 1);

    IDataViewdataView =
mlContext.Data.LoadFromTextFile<ECGWord>($"C:\\Words_ECGDataset_{desease}_withGood.
tsv", hasHeader: true);

    TrainTestDatatrainTestSplit = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);
IDataViewtrainingData = trainTestSplit.TrainSet;
IDataViewtestData = trainTestSplit.TestSet;

    vardataProcessPipeline =
mlContext.Transforms.Text.FeaturizeText(outputColumnName: "Features", inputColumnName:
nameof(ECGWord.Text));

    var trainer =
mlContext.BinaryClassification.Trainers.SdcaLogisticRegression(labelColumnName: "Label",

```

```

featureColumnName: "Features", l2Regularization: 0.000006f, l1Regularization: 0.000013f,
maximumNumberOfIterations: 10000);
    var trainingPipeline = dataProcessPipeline.Append(trainer); ITransformer trainedModel =
trainingPipeline.Fit(trainingData);
    var predictions = trainedModel.Transform(testData);
    var metrics = mlContext.BinaryClassification.Evaluate(data: predictions,
labelColumnName: "Label", scoreColumnName: "Score");

    mlContext.Model.Save(trainedModel, trainingData.Schema, $"C:\\{disease}.zip");

    Console.WriteLine($"\\n\\n\\n===== ML FOR :{disease}
=====\\n"); Console.WriteLine($"ROC AUC: {metrics.AreaUnderRocCurve}\\n");
    Console.WriteLine($"ACURACY: {metrics.Accuracy}");
    Console.WriteLine($"NegativePrecision: {metrics.NegativePrecision}");
Console.WriteLine($"NegativeRecall: {metrics.NegativeRecall}");
Console.WriteLine($"PositivePrecision: {metrics.PositivePrecision}");
Console.WriteLine($"PositiveRecall: {metrics.PositiveRecall}");

    return $"{{disease.Split("_")[1]}{delimiter}{{disease.Split("_")[0]}{delimiter}{{metrics.Area
UnderRocCurve}}";
    }
}

```

ДОДАТОК Г
(Обов'язковий)

Додаток Г
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

Автоматизована система аналізу сигналів електрокардіограм

1. Схема нормального запису ЕКГ сегменту;
2. Приклади PQRS;
3. Метод локальних мінімумів та максимумів;
4. Алгоритм поділу на R-інтервали;
5. Алгоритм розбиття ЕКГ на опорні точки інтервалів;
6. Схема структурна класів;
7. Концепція шаблону Model-View-Presenter;
8. Окремі результати роботи програмного забезпечення;
9. Аналіз ефективності та оцінка результатів;
10. Ефективність методів;

Студент групи 2АКІТ-22м



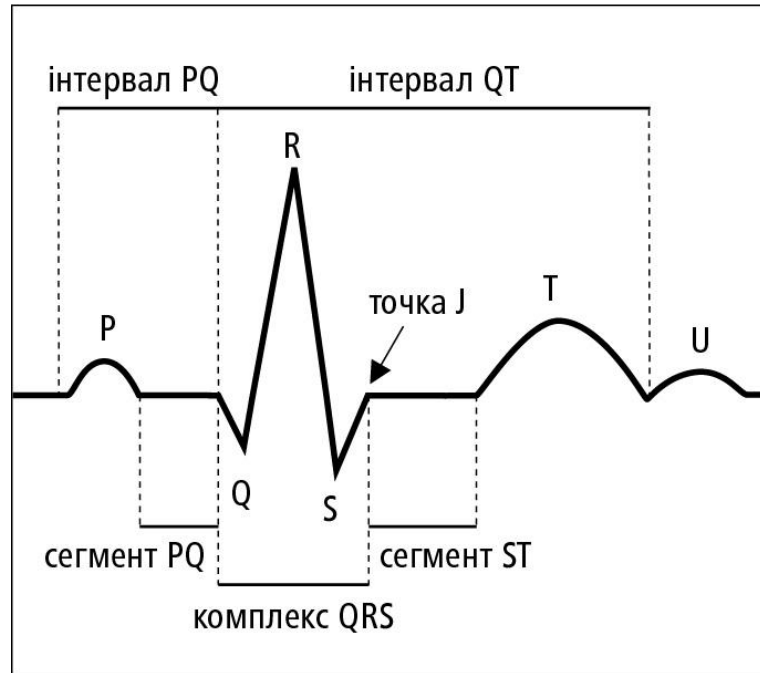
Євгеній СЛОБОДЯНЮК

Керівник к.т.н., доцент кафедри
КСУ



Олена НИКИТЕНКО

Схема нормального запису ЕКГ сегменту



Приклади PQRS

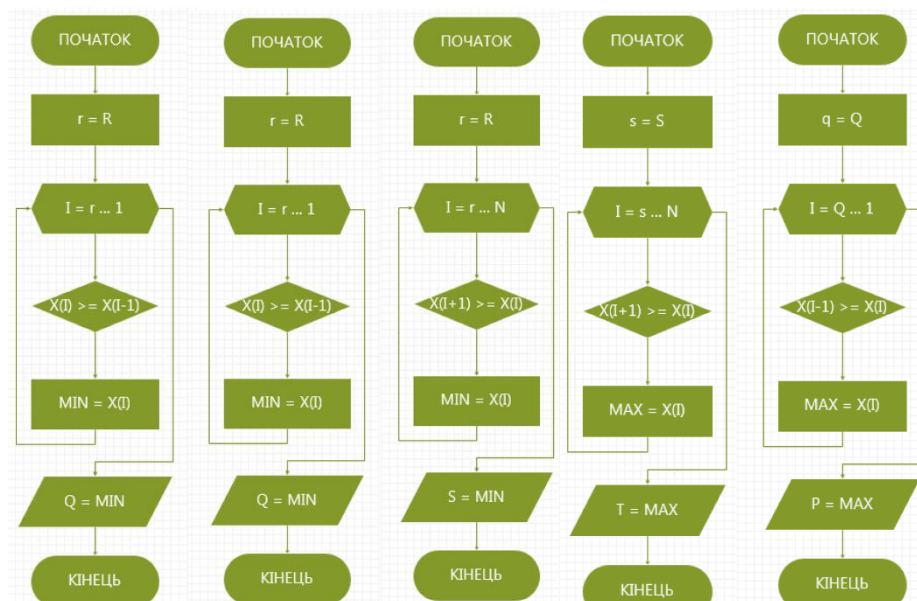


Приклад PQRS здорової людини

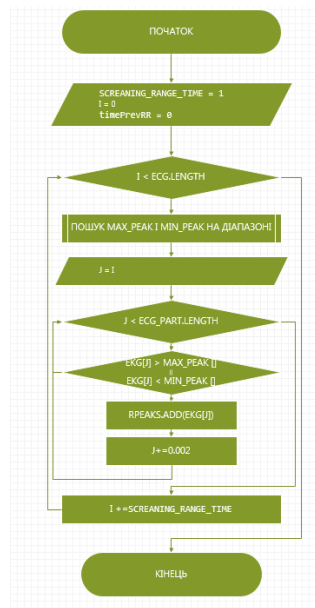


Приклад викривлення PQRS хворої людини

Метод локальних мінімумів та максимумів



Алгоритм поділу на R-інтервали



Алгоритм розбиття ЕКГ на опорні точки інтервалів

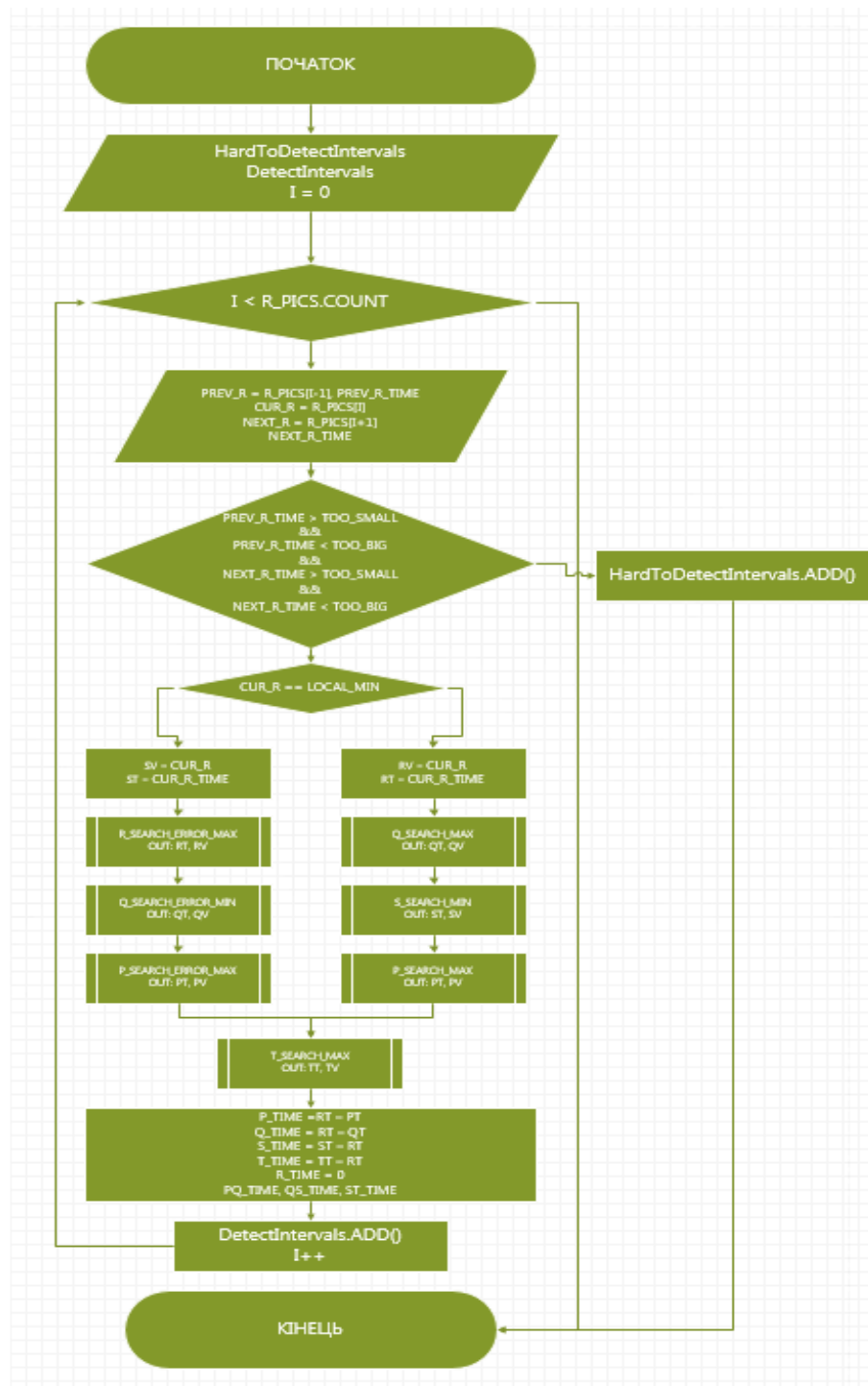
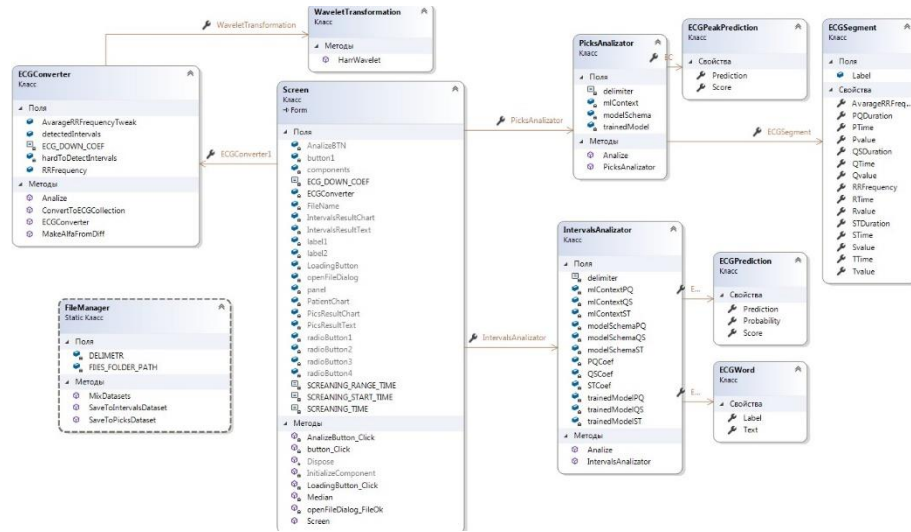
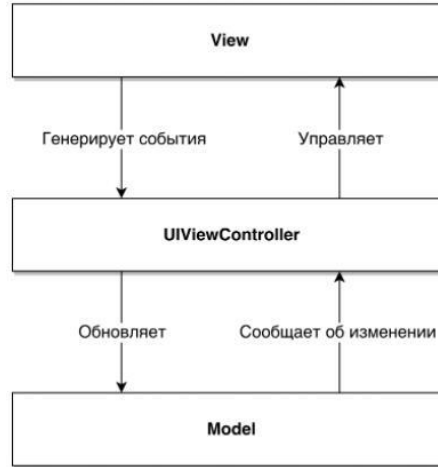


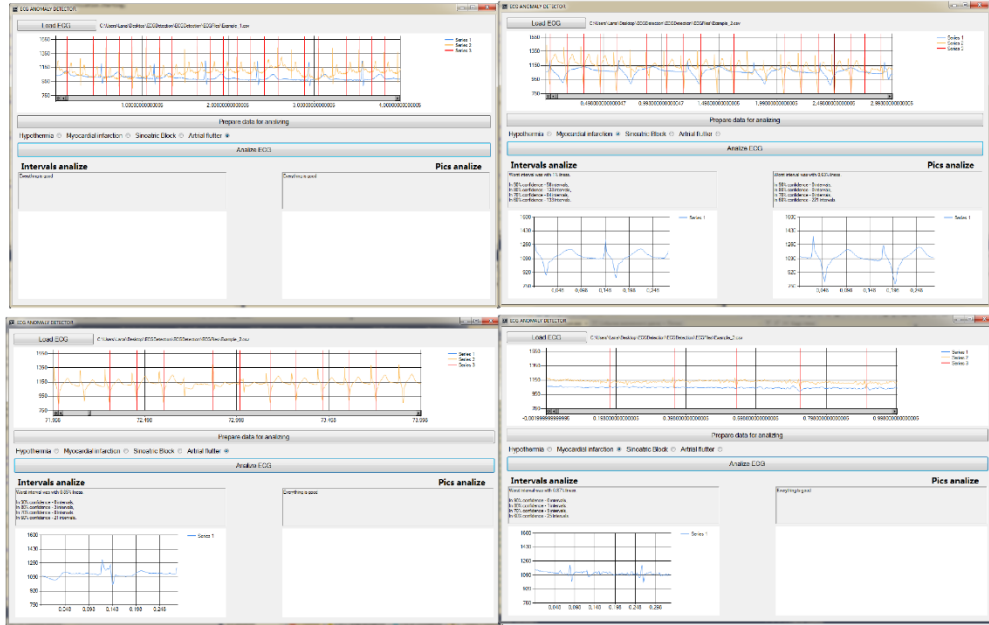
Схема структурна класів



Концепція шаблону Model-View-Presenter

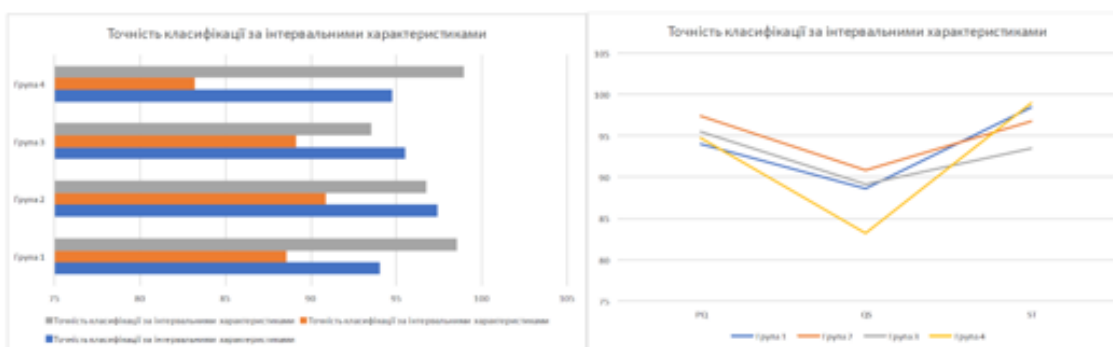
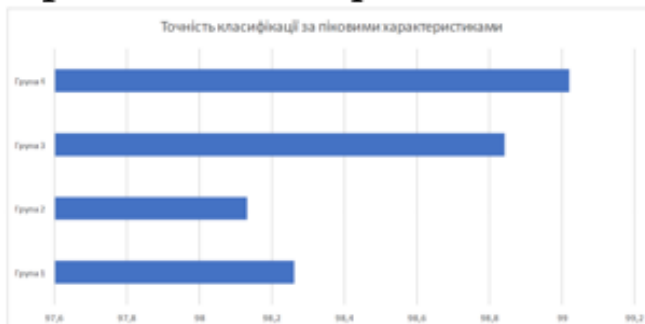


Окремі результати роботи програмного забезпечення



Аналіз ефективності та оцінка результатів

Аналіз ефективності запропонованого підходу



Матриця оцінки результатів алгоритму класифікації

Prediction

		1	0
Actual	1	True Positive (TP)	False Negative (FN)
	0	False Positive (FP)	True Negative (TN)

Ефективність методів

Ефективність пікового методу



Ефективність інтервального методу

