

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису  
УДК 004.896

«До захисту допущено»  
В.О. завідувача кафедри  
ММСА  
\_\_\_\_\_ О.Л. Тимощук  
«\_\_» \_\_\_\_\_ 2020 р.

## Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 122 Комп'ютерні науки  
на тему: «Пацієнт-специфічна система класифікації ЕКГ на основі  
синтезованих зразків»

Виконала:  
студентка II курсу, групи КА-93мп  
Кравченко Влада Андріївна \_\_\_\_\_

Керівник:  
професор кафедри ММСА,  
д. т. н., проф. Данилов В. Я. \_\_\_\_\_

Рецензент:  
професор кафедри інформаційної безпеки  
КПІ ім. Ігоря Сікорського  
д. т. н., проф. Качинський А. Б. \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

Київ  
2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти – другий (магістерський)

Спеціальність (спеціалізація) — 122 «Комп’ютерні науки» («Системи штучного інтелекту»)

ЗАТВЕРДЖУЮ

В.О.Завідувача кафедри

\_\_\_\_\_ О.Л.

Тимощук

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

### ЗАВДАННЯ

на магістерську дисертацію студенту Кравченко Владі Андріївні

1. Тема роботи «Пацієнт-специфічна система класифікації ЕКГ на основі синтезованих зразків», керівник роботи Данилов Валерій Якович, д.т.н, професор, затверджені наказом по університету від «02» листопада 2020 р. № 3182-с
2. Термін подання студентом дисертації: 17 грудня 2020 р.
3. Об’єкт дослідження: аномалії роботи серця та їх відображення в сигналах електрокардіограм
4. Предмет дослідження: методи генерації синтетичних часових сигналів кардіограм та системи їх розпізнавання і класифікації
5. Перелік завдань, що потрібно розробити:
  - 1) провести аналіз існуючих систем та методів кардіограм
  - 2) дослідити засоби генерації синтетичних часових сигналів методами штучного інтелекту
  - 3) виконати проектування системи класифікації сигналів кардіограм
  - 4) реалізувати один з модулів спроектованої системи

5) розробити стартап-проект виведення на ринок результатів дослідження.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

1) Архітектури досліджених нейронних мереж

2) Графіки навчання нейронних мереж

3) Принципи роботи нейронних мереж

7. Орієнтовний перелік публікацій:

Кравченко В. А. Пацієнт-специфічна система класифікації ЕКГ на основі синтезованих зразків // Проблеми інформатизації: наук. техн. конф. (Черкаси-Харків-Баку-Бельсько-Бяла, 26-27 листопада 2020 року). Харків: друкарня ФОП Петров В. В., 2020. с. 6

8. Дата видачі завдання 01.09.2020

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів роботи
1.	Концептуальний вступ дисертації. Формулювання об'єкта, предмета, цілі, завдань, новизни, практичної значущості результатів	01.09.2020-19.09.2020
2.	Перший розділ. Огляд літературно-інформаційних джерел. Понятійно-категоріальний апарат. Характеристика об'єкта	20.09.2020-27.09.2020
3.	Другий розділ. Попередня обробка даних ЕКГ	27.09.2020-05.10.2020
4.	Третій розділ. Застосування генеративно-змагальних мереж в задачах генерації часових сигналів	05.10.2020-21.10.2020
5.	Четвертий розділ. Методи класифікації ЕКГ	21.10.2020-04.11.2020
6.	П'ятий розділ. Опис архітектури системи та аналіз отриманих результатів	04.11.2020-19.11.2020
7.	Шостий розділ. Стартап-проект	19.11.2020-23.11.2020
8.	Концептуальні висновки. Перспективи розвитку отриманих рішень	23.11.2020-30.11.2020

Студент

В. А. Кравченко

Науковий керівник дисертації

В. Я. Данилов

## РЕФЕРАТ

Магістерська дисертація: 107 с., 47 рис., 39 табл., 42 джерела.

Мета роботи – розробка системи класифікації сигналів кардіограм для діагностики захворювань серця на основі розширеного датасету з синтезованими сигналами ЕКГ.

Об'єктом дослідження є набір даних кардіограм пацієнтів з захворюваннями серця.

Предметом дослідження є засоби генерації часових медичних сигналів та методи класифікації на основі штучних нейронних мереж.

В роботі досліджується проблема розпізнавання стану здоров'я людини, шляхом аналізу її електрокардіограми, а також програмні засоби для реалізації поставленої задачі.

Виконано аналіз методів обробки часових сигналів, проведено дослідження засобів генерації штучних сигналів ЕКГ, аналіз математичних структур та сучасних методів класифікації.

Результатом роботи є аналіз сучасних систем та підходів до вирішення задачі класифікації кардіограм, розробка модифікованої системи класифікації зі збільшенням кількості вхідних даних шляхом додавання синтетичних сигналів ЕКГ. Під час роботи програмно реалізовано запропоновану систему з використанням мови програмування Python, що надає широкий спектр бібліотек для обробки та аналізу даних.

КЛАСИФІКАЦІЯ, ГЕНЕРАТИВНО-ЗМАГАЛЬНІ МЕРЕЖІ,  
РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, ЕЛЕКТРОКАРДІОГРАМА, ЗГОРТКОВІ  
НЕЙРОННІ МЕРЕЖІ

## ABSTRACT

Master's thesis: 107 pages, 47 fig., 39 tables, 42 sources.

The purpose of work - development of systems of classification of signals, the cardiogram for diagnosis of heart diseases on the basis of the expanded dataset with the synthesized ECG signals.

The object of the study is a set of cardiogram data of patients with heart disease.

The subject of the study is the means of generating hours of medical signals and methods of classification based on artificial neural networks.

The paper investigates the problem of recognizing the state of human health by analyzing its electrocardiogram, as well as software for this task.

The analysis of methods of processing of time signals is carried out, research of illuminations of generations of control signals of an ECG, the analysis of mathematical structures and modern methods of classification is carried out.

The result is the analysis of modern systems and approaches to solving the problem of classification of the cardiogram, the analysis of modified classification systems by increasing the number of input data by the completion of synthetic signals. During the work, the proposed system is implemented using software using Python programs, allows you to get a wide range of libraries for data processing and analysis.

CLASSIFICATION, GENERATIVE-ADVERSARIAL NETWORKS,  
RECURRENT NEURAL NETWORKS, ELECTROCARDIOGRAPHY,  
CONVOLUTIONAL NEURAL NETWORKS

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ .....	8
ВСТУП.....	9
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ .....	11
1.1 Аналіз вимог до системи розпізнавання ЕКГ. Постановка задачі.....	11
1.2 Засоби обробки сигналів кардіограм.....	12
1.3 Будова та функціонування серця, зв'язок з ЕКГ .....	13
1.4. Висновки до розділу.....	17
РОЗДІЛ 2 НЕЛІНІЙНІ ЗАСОБИ АНАЛІЗУ ЧАСОВИХ РЯДІВ КАРДІОГРАМ .....	18
2.1. Графік Пуанкаре .....	20
2.2. Фрактальна розмірність .....	23
2.3. Застосування фрактальної розмірності в кардіології .....	27
2.4 Висновки до розділу.....	29
РОЗДІЛ 3. ГЕНЕРАЦІЯ ЧАСОВИХ СИГНАЛІВ ЕКГ .....	30
3.1. Пацієнт-специфічне моделювання .....	30
3.2. Генеративно-змагальні мережі .....	31
3.2.1. Генеративно-змагальні мережі часових рядів .....	36
3.2.2. Генеративно-змагальні мережі для синтезу специфічних для класу даних .....	37
3.3. Способи оцінювання синтетичних даних часових рядів .....	40
3.4. Застосування синтезованих даних в навчанні класифікатора .....	41
3.5. Висновки до розділу.....	43

РОЗДІЛ 4. ДОСЛІДЖЕННЯ МЕТОДІВ ПРИЙНЯТТЯ РІШЕНЬ В ЗАДАЧАХ КЛАСИФІКАЦІЇ ЕЛЕКТРОКАРДІОГРАМ .....	45
4.1. Глибинне навчання в задачах класифікації ЕКГ .....	45
4.1.1. Рекурентні нейронні мережі .....	48
4.1.2. Мережі LSTM .....	52
4.1.3. Згорткові нейронні мережі.....	56
4.2. Критерії якості оцінки рішення задачі класифікації .....	59
4.3. Висновки до розділу.....	61
РОЗДІЛ 5. ОПИС ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ .....	63
5.1. Опис вхідних даних.....	63
5.2. Розрахунок фрактальної розмірності .....	65
5.3. Генерація синтетичних зразків .....	67
5.4. Оцінка якості результатів класифікації.....	71
5.5. Висновки до розділу.....	76
РОЗДІЛ 6. РОЗРОБКА СТАРТАП-ПРОЕКТУ .....	77
6.1. Опис ідеї проекту .....	78
6.2. Технологічний аудит ідеї проекту .....	80
6.3. Аналіз ринкових можливостей запуску стартап-проекту.....	81
6.4. Аналіз ринкової стратегії проекту .....	89
6.5. Розроблення маркетингової програми стартап-проекту.....	91
6.6. Висновки до розділу.....	96
ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ ЩОДО ПОДАЛЬШИХ ДОСЛІДЖЕНЬ.....	97
ПЕРЕЛІК ПОСИЛАНЬ .....	98
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ .....	103

## ПЕРЕЛІК СКОРОЧЕНЬ

ВП – вейвлет-перетворення

ГЗМ – генеративно-змагальні мережі

ЕКГ - електрокардіограма

ЗНМ – згорткова нейронна мережа

ПСМ – пацієнт-специфічне моделювання

РНМ – рекурентна нейронна мережа

СППР – система підтримки прийняття рішень

ЦОС – цифрова обробка сигналів

ФР – фрактальна розмірність

ШІ – штучний інтелект



## ВСТУП

За офіційними даними ВООЗ у 2016 році найбільша кількість людських смертей, а саме третина - 17,9 млн випадків, трапляється через патологію серця та кровоносних судин. Ці хвороби виділяють у певних клас. До нього належать інфаркти та інсульты, гіпертонія, хвороба периферичних артерій, ревматична та вроджена вади серця і серцева недостатність. Державна служба статистики України стверджує, що показник смертності українців через серцево-судинні хвороби складає 199905 осіб за період січня-вересня 2020 року, що складає 46,86% ві загальної кількоісті смертей за цей період.

З високою точністю дипломовані фахівці та спеціалісти-кардіологи, опираючись на сучасні методи діагностики, можуть визначити у людини наявні хвороби серця на судин та їх ступінь важкості. Для аналізу та лікування захворювань пацієнта використовують різноманітні методи обстеження роботи серця. Найбільш поширений та інформативний спосіб - ЕКГ. Електрокардіограма - це запис електричної активності роботи серця. Високий відсоток неправильного діагностування трапляється через людський фактор, адже існує велике різноманіття підходів до діагностики, а також через дефекти запису роботи серця, які прийнято називати артефактами. Тож нагальною постає потреба у вивченні та дослідженні нових методів обробки електрокардіограми.

Обробка ЕКГ людиною досить трудоємкий процес та займає багато часу, тому був запатентований новий спосіб - цифрова обробка. Для визначення хвороб серця використовують такі параметри, як міжзубцеві інтервали, частоти та інші. Набір значень, що отримують лікарі, можна представити у вигляді певної залежності (функції). Аналіз цього електричного сигналу та виявлення відхилень за допомогою спеціальних програмних продуктів і доробок новітніх технологій значно пришвидшує обробку даних, що в свою чергу підвищує ефективність дігностики, визначення хвороби та подальшого лікування фахівцем в області кардіології.

Мета цієї роботи полягає в розробці системи розпізнавання даних, отриманих за допомогою електрокардіограми, та їх аналітики, тобто визначення наявності або відсутності хвороб серця, а також для класифікації п'яти видів патології серцебиття.

## РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Аналіз вимог до системи розпізнавання ЕКГ. Постановка задачі

Метою дипломної роботи є розробка системи класифікації ЕКГ на основі синтезованих зразків. Програма має класифікувати захворювання серця у разі їхньої наявності. Окремий програмний модуль використовується для генерації синтетичних сигналів ЕКГ для збільшення набору даних з метою підвищення точності класифікації.

Дана система може бути використана у галузі кардіології для покращення ефективності обробки сигналу ЕКГ та для діагностики на основі даних ЕКГ такої патології серця, як аритмія. Аритмія - це будь-яке відхилення від нормального синусового ритму. Програмний продукт дозволяє зменшити затрати часу на встановлення діагнозу, що є, безсумнівно, дуже значущим чинником у вирішенні проблем, що стосуються серцево-судинної системи.

Необхідно розв'язати наступні задачі для досягнення мети:

- Дослідити зв'язок особливостей будови та функціонування серця з сигналами електрокардіограм.
- Дослідити методи генерації медичних часових сигналів.
- Дослідити методи прийняття рішень.
- Реалізувати блок генерації синтетичних сигналів.
- Реалізувати блок прийняття рішень.
- Провести навчання системи.
- Виконати аналіз одержаних результатів.

## 1.2 Засоби обробки сигналів кардіограм

Різноманіття способів аналітики електрокардіограми налічує численну кількість. Можна окреслити основні: безпосередній аналіз форми сигналу (визначення PQRST – комплексу, визначення RR-інтервалів тощо), використання нейронних мереж, застосування функціональних перетворень, що є чутливими до змін локальних властивостей сигналу (швидке перетворення Фур'є, вейвлет-перетворення).

Аналіз ЕКГ з використанням відповідних програмних засобів є достатньо поширеним способом на сьогоднішній день. Його можна розділити на три основних етапи: цифрова обробка сигналу, виділення ознак, аналіз одержаних параметрів.

Перший етап включає у себе переведення сигналу ЕКГ у цифрову форму, фільтрацію (видалення шумів та артефактів) і нормування сигналу. Шумом вважаються високочастотні компоненти кардіосигналу, його видалення призводить до згладжування. Другий - розпізнавання характерних ділянок ЕКГ (зубці, сегменти, QRS-комплекс). Останній етап являє собою аналіз параметрів і взаємозв'язку розпізнаних ділянок і формування підсумкового висновку.

Застосування автоматичних алгоритмів розпізнавання дозволяють отримувати якісні результати.

У таблиці 1.1 наведено список програм для аналізу та інтерпретації сигналу ЕКГ від провідних світових компаній-виробників електрокардіографів з зазначенням їх ключових можливостей.

Таблиця 1.1 - Програми для аналізу ЕКГ

Назва	Виробник	Можливості
Heart Rate Variability Software	Schiller (Швейцарія)	Аналіз варіабельності серцевого ритму (ЕКГ)

Продовження таблиці 1.1

Marquette 12SL	GE Healthcare (США)	Аналіз ЕКГ, унікальні критерії оцінки головних сегментів ЕКГ, автоматичне визначення аритмій
Cardiosoft	GE Healthcare (США)	Вимірювання інтервалів ЕКГ, аналіз основних зубців, аналіз аритмій
FP-804	Fukuda (Японія)	Основні виміри: ЧСС, RR, PR, QRS, час QT, QTc, електрична ось, SV, RV5(6); 120 типів кодів інтерпретації, 130 типів кодів Мінесоти

Серед проблем аналітики електрокардіограми можна виділити основні, такі як: топологія захворювань пацієнта не враховуються при цифровій обробці даних, проте лікар-діагност бере їх до уваги; діагностична похибка; некомфортний інтерфейс програмного продукту або технічного засобу. За деякою інформацією результати комп'ютерного аналізу даних в 5-20% випадків різняться с заключенням спеціаліста-кардіолога.

### 1.3 Будова та функціонування серця, зв'язок з ЕКГ

У 1903 році нідерландський спеціаліст у сфері фізіології та медицини Віллем Ейнтговен сконструював пристрій, що дозволяє визначити показники

роботи серця завдяки запису електричної активності. Він запатентував електрокардіограф, за що у 1924 році удостоївся Нобелівської премії. Електрокардіографія - це реєстрація різниці коливань електричного поля серцевого м'язу (міокарду). Зазвичай електрокардіограф використовує 6 відведень, що накладаються на кінцівки та грудну клітину людини, в розширеному випадку - 12 (6 додаткових електродів закріплюються в області грудей). В результаті маємо запис на стрічці кардіограми, за допомогою якої діагностується наявність або відсутність патологій роботи серця.

За технічними властивостями серце - це насос, що містить чотири камери: праве передсердя і шлуночок, що формують праве серце, та ліве передсердя і шлуночок, що формують ліве. Насичена киснем кров з легень поступає в ліве передсердя, звідси в лівий шлуночок, а потім в аорту. Далі слідує по венам до правого передсердя, звідти потрапляє в правий шлуночок і далі артерією прямує до легенів, де начисується киснем і знову надходить до лівого передсердя. На електрокардіограмі скорочення шлуночків реєструються як відхилення від ізолінії. Її іменували зубцями и позначають літерами P, Q, R, S, T.

На рисунку 1.1 сформовано порядок утворення зубців ЕКГ за рахунок скорочень серця.

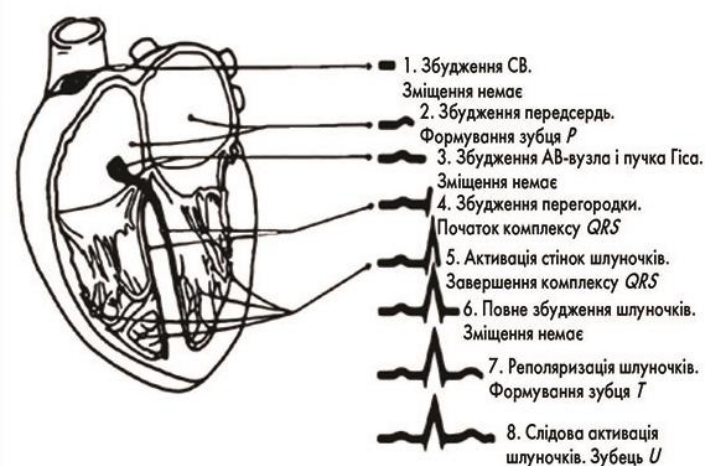


Рисунок 1.1 - Схема поширення збудження провідною системою серця та схематичне зображення кривої ЕКГ

До серця струм потрапляє через синусовий вузол, саме тому серцевий ритм у нормі називається синусовим. Вузол знаходиться у правому передсерді, через що першим збуджується саме воно, надалі імпульс переходить до лівого передсердя. Сигнали від збудження кожного передсердя фіксуються і внаслідок накладання вони утворюють Р зубець.

Сигнал, зафіксований на міжшлунковій перегородці формує зубець Q, реєструючи на електрокардіограммі значення зі знаком мінус через те, що вектор збудження напрямлений вбік від датчика. Потім в стан збудження приходить верхівка серця, вектор якої напрямлений прямо до електрода, фіксуючи зубець R, який є найбільшим. Останнім формується пік S на основі серця, вектор якого напрямлений проти датчика, отже має від'ємне значення. Ці три зубці разом складають QRS-комплекс.

Після цього можемо фіксувати ST-інтервал, що можемо побачити на рисунку 1.2. На цьому інтервалі серце відновлюється і починає цикл знову.

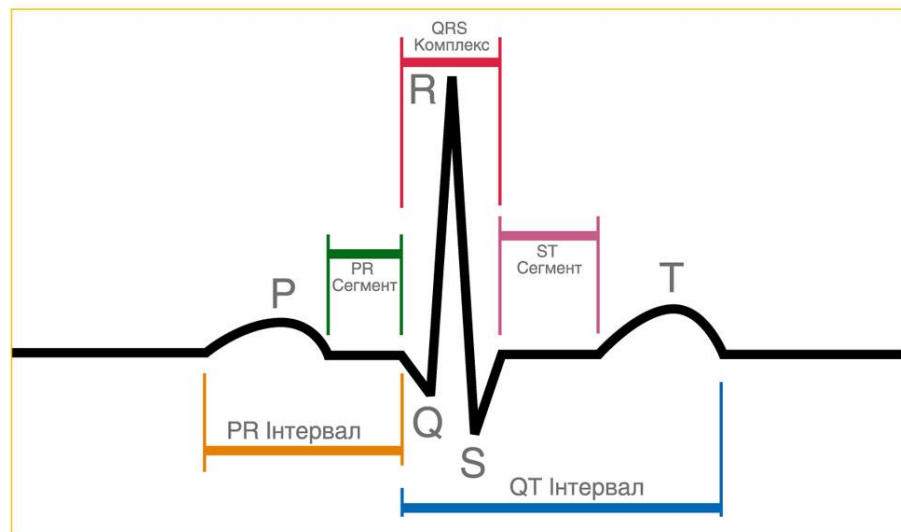


Рисунок 1.2 - Елементи ЕКГ в нормі

Зазвичай певні незвичайні показники, закономірності, аномальні поєднання циклів, хвиль та складових сигналу, які видно на кардіограмі допомагають кардіологу визначити причину захворювання. Діагностика та лікування можуть залежати від оцінки деяких періодів, які визначені

координатами певних точок (піки, зміна P, QRS, T і U-хвиль). Конфігурації хвиль, комплекси сигналів, певні закони, що видно на графіку ЕКГ, можуть оцінюватися лікарем вручну або за допомогою доробок новітніх технологій, зокрема використовуючи різноманітні програми та спеціальне обладнання. Правильність інтерпретації даних корелює з досвідом та ступінню обізнаності фахівця, оскільки холтер, що працює добу, має в собі близько ста тисяч ударів. Обробки такої великої кількості даних людиною є нелегкою задачею.

Аритмія (будь-яке відхилення від нормального синусового ритму) є однією з найбільш поширених хвороб серцево-судинної системи. Нормальна частота, з якою Р-піки з'являються на графіку варіюється в проміжку від 60 до 90 разів за хвилину. Також приділяють особливу увагу часовому поміжку між появою Р-зубців. Якщо дельта між найбільшим і найменшим більша 0,12 секунд, йдеться про аритмію, якщо ж самі інтервали менші 0,12 секунд, то діагностують неправильний синусовий ритм (рис. 1.3).

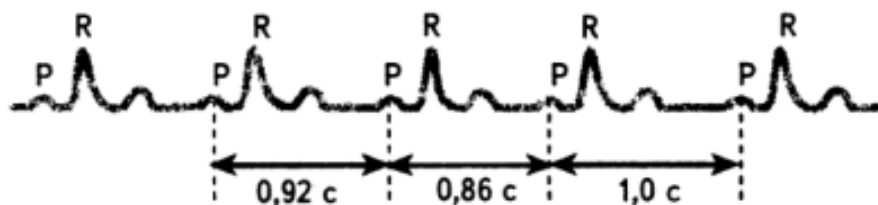


Рисунок 1.3 - Крива ЕКГ хворого на аритмію

Європейським стандартом визначено, що лікар має визначити порушення ритму, його характер та причину і назначити коректне лікування у випадку підозри на таке захворювання серця, як аритмія.



#### 1.4. Висновки до розділу

В даному розділі розглянута предметна область задачі, описані технічні засоби, які допомагають у виявленні та аналізі хвороб серцево-судинної системи, приведений опис програмних продуктів. Також була окреслена робота серця і наведено визначення електрокардіограми як основного інструмента його діагностики

## РОЗДІЛ 2 НЕЛІНІЙНІ ЗАСОБИ АНАЛІЗУ ЧАСОВИХ РЯДІВ КАРДІОГРАМ

Динаміка серцевого ритму є фізіологічним показником, який аналізується чи не найчастіше. Було запропоновано безліч математичних методів оцінки варіабельності серцевого ритму: ці методи успішно застосовувались у дослідженнях серцево-судинної динаміки в здорових та патологічних станах, проте вони все ще є далекими від клінічної практики.

Серед існуючих методів можна виділити нелінійні, а саме ті, що засновані на теорії хаосу, фрактальності та складності: графік Пуанкаре, аналіз графіку повторюваності, фрактальна розмірність та розмірність кореляції, показник Херста тощо.

В науці термін «хаос» стосується математичного підходу до систем, що можна повністю описати, проте ці системи генерують випадкові результати при певних умовах. Теорія хаосу розглядає закономірності часової еволюції нелінійної системи, чутливої до початкових умов. З іншого боку, фрактал – це об'єкт, що складається з субодиноць, які нагадують великомасштабну структуру. Така властивість самоподібності (або масштабної інваріантності) означає, що деталі структур є схожими, але не обов'язково ідентичними при масштабуванні з різною роздільною здатністю.

Фрактальна організація є гнучкою, і порушення цієї масштабної інваріантності може призвести до більш жорсткої та менш адаптивної системи або з випадковою, або з сильно корельованою структурою. Тим не менш визначення фрактала виходить за межі самоподібності як такої, що включає ідею деталізованого малюнка, який повторюється у все менших масштабах. Хоча фрактали нерегулярні, не всі часові ряди є фрактальними. Самоподібність коливання системи можна спостерігати, коли сигнал аналізується в різних часових масштабах.

Концепція, пов'язана з динамічним хаосом та фрактальністю, є концепцією складності. Є багато визначень складності, одне з найпоширеніших – складність

– це властивість кожної системи, яка визначає кількість структурованої інформації. Шеннон продемонстрував, як інформація в сигналі може бути визначена кількісно з абсолютною точністю як кількість неочікуваних даних, що містяться в повідомленні (позначена ентропія). Колмогоров запропонував визначення складності, яке кількісно оцінює інформацію про окремі об'єкти як розмір найменшого представлення.

Дослідження, пов'язані зі складністю, розкидані по різних наукових сферах, таким як фізика, економіка, біологія, психологія тощо. Більшість цих досліджень спрямовано на пошук закономірностей або законів, що керують динамікою набору даних. Багато нелінійних методів, заснованих на концепції хаосу, фрактальності та складності використовувались при оцінюванні варіабельності серцевого ритму для розуміння серцево-судинної динаміки як у здорових, так і в патологічних станах (рис. 2.1).

В останні роки наукова спільнота опублікувала кілька оглядових статей, пов'язаних з нелінійними методами, що застосовуються до часових рядів частоти пульсу. В огляді Фосса (Voss) 2009 року узагальнені деякі індекси варіабельності серцевого ритму, отримані з нелінійної і фрактальної динаміки (фрактал, ентропія, графічне представлення Пуанкаре). Огляд демонструє актуальність методів в клінічних дослідженнях та відображає суттєві аспекти, пов'язані з їхнім практичним застосуванням, хоча і не надає математичного описання алгоритмів. У 2016 Годой (Godoy) зробив огляд літератури про лінійні та нелінійні методи прогнозування серцево-судинних захворювань та прийшов висновку, що нелінійні методи є більш ефективними, ніж лінійні.

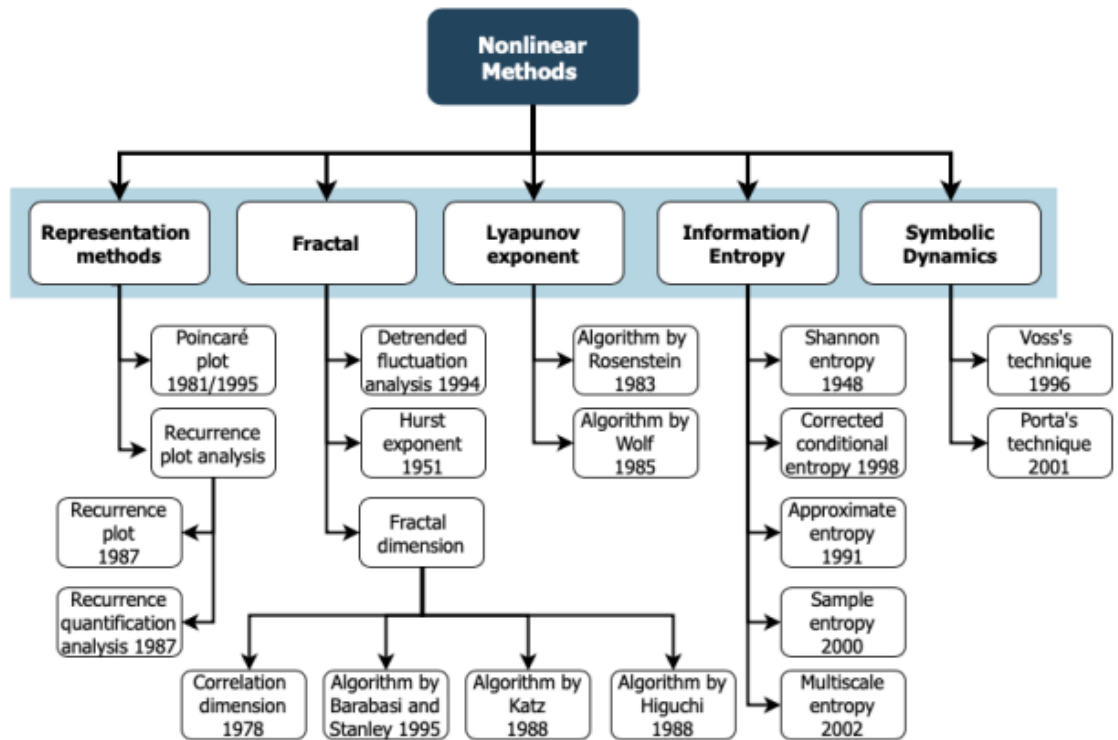


Рисунок 2.1 – Нелінійні методи, що застосовуються до часових рядів сигналу ЕКГ

## 2.1. Графік Пуанкаре

Графік Пуанкаре також відомий як карта повернення або затримки, дозволяє оцінити динаміку серцебиття на основі спрощеного вкладання у фазовий простір. Графік Пуанкаре – це двомірний графік, на якому кожен інтервал RR,  $x(i)$  нанесений як функція попереднього інтервалу RR,  $x(i-1)$ . Кожна пара інтервалів RR буде відображатися в одній точці на графіку. Чим довшим є моніторинг, тим більше точок з'являється на графіку.

Аналіз графіку Пуанкаре – це кількісно-візуальний метод, за якого форма графіку надає зведену інформацію про поведінку серця.

Для здорового серця сукупність точок являє собою форму комети, орієнтовану вздовж лінії ідентичності (бісектриса першої чверті координатної площини). Динаміка серцевої недостатності характеризується сукупністю точок

у формі еліпсу також вздовж лінії ідентичності. У випадку фібриляції передсердь сукупність точно має більш круглу форму, аналогічно до того, що відбувається з часовим рядом білого шуму (рис. 2.2).

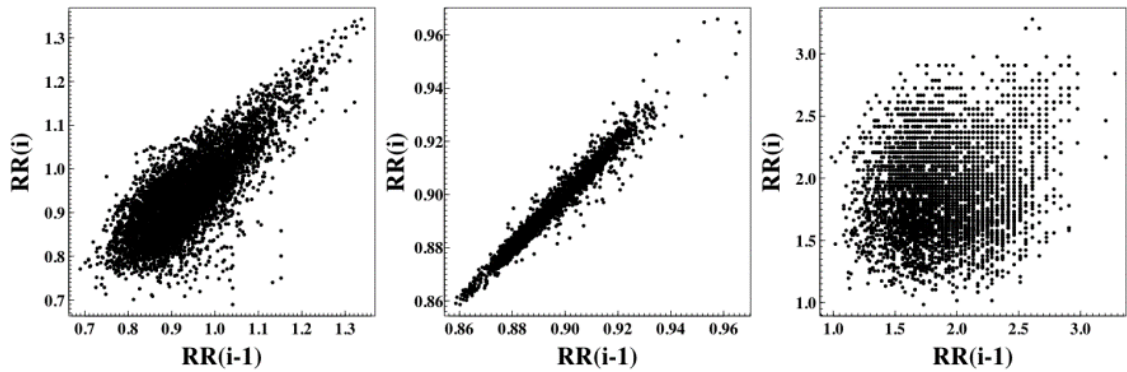


Рисунок 2.2 – Графіки Пуанкаре трьох RR часових рядів: перший відображає нормальний синусоїдальний ритм, другий – серцева недостатність, третій – фібриляція передсердь. Слід зважити, що значення осей відрізняються в трьох випадках

Було розроблено ряд методів для узагальнення геометричних фігур графіка Пуанкаре кількісно. Геометричні дескриптори, такі як техніка апроксимації еліпса, гістограмний метод, коефіцієнт кореляції є найбільш популярними в клінічній літературі та літературі по ВСП (варіабельності серцевого ритму).

Розподіл точок вздовж перпендикуляру до лінії ідентичності відображає рівень короткочасної варіабельності, що виражається показником SD1. З іншого боку, дисперсія точок вздовж лінії ідентичності відображає рівень довгострокової варіабельності та вимірюється стандартним відхиленням, що позначається як SD2.

Стандартне відхилення RR інтервалів, що позначається як SDRR, зазвичай застосовується як міра загальної ВСП. Воно визначається як квадратний корінь дисперсії RR інтервалів:

$$SDRR = \sqrt{E[X^2] - E[X]^2}$$

де  $E[X]$  - середнє значення інтервалу RR.

Стандартне відхилення послідовних різниць інтервалів RR, що позначаються SDSD, є важливим показником короткочасного ВСР. Воно визначається як квадратний корінь дисперсії послідовності  $\Delta X(i) = x(i) - x(i + 1)$ :

$$SDSD = \sqrt{E[\Delta X^2] - E[\Delta X]^2}$$

Варто відмітити, що  $E[\Delta X] = 0$  для стаціонарних інтервалів; отже, SDSD еквівалентно кореню середньоквадратичного послідовних різниць, що позначається RMSSD.

Геометричні показники, отримані шляхом підгонки еліпса до графіка Пуанкаре, залежать від загальних індексів ВСР у часовій області. Ширина графіка Пуанкаре - це лінійне масштабування найбільш поширених статистичних даних, що використовується для вимірювання короткочасного ВСР, індексу SDSD. Насправді ширина графіка Пуанкаре корелює з іншими вимірами короткочасного ВСР.

$$SD1^2 = Var\left(\frac{1}{\sqrt{2}}X(i) - \frac{1}{\sqrt{2}}X(i+1)\right) = \frac{1}{2}Var(\Delta X) = \frac{1}{2}SDSD^2$$

де Var – відхилення:

$$SD2^2 = 2SDRR^2 = \frac{1}{2}SDSD^2$$

Два простих узагальнення графіка Пуанкаре – графіки Пуанкаре з відставаннями (лагами) і графіки Пуанкаре вищого порядку - також можна зустріти в літературі. На графіках Пуанкаре з відставаннями (lag - c),  $x(i)$  наноситься замість  $x(i + c)$ , де c - деяке невелике додатне ціле значення. Загалом,

графік все ще кластеризований навколо лінії ідентичності. Однак довжина та ширина ділянки змінюються у міру збільшення відставання.

З іншого боку, вважаючи стандартний графік Пуанкаре першим порядком, графік Пуанкаре другого порядку є тривимірним графіком розсіювання трійок  $(x(i), x(i+1), x(i+2))$ . Існує три ортогональні види форми цього графіку, що призводить до 2D-проекцій на кожен з координатних площин  $(x(i), x(i+1))$ ,  $(x(i+1), x(i+2))$  та  $(x(i), x(i+2))$ . Перші два подання еквівалентні стандартному графіку Пуанкаре, а третій - lag-2 графік. Цю ідею можна поширити у вищі виміри, за допомогою проєкцій графіку на координатні площини, що відстають від графіків Пуанкаре. Отже, порядок  $s$  графіка Пуанкаре описується геометрично за допомогою набору графіків Пуанкаре з відставанням до lag- $s$  включно.

Графік Пуанкаре - це потужний інструмент не тільки для графічного представлення зведеної статистики, але і також для структури "від удару до удару".

## 2.2. Фрактальна розмірність

Фрактальна розмірність (FD) - це статистичний показник того, як деталі образу змінюються з масштабом, в якому він вимірюється. FD дозволяє визначити, скільки місця займає об'єкт між евклідовими розмірностями. FD сигналу є потужним інструментом для виявлення перехідних процесів. Чим вище FD, тим більше сигнал нерегулярний, тобто тим більше сигнал самоподібний.

З кількох доступних алгоритмів для обчислення FD часового ряду, найпоширеніші чотири - розмірність кореляції, розмірність box-counting та алгоритми, запропоновані вченими Кац (Katz) та Гігучі (Higuchi).

### 1. Розмірність кореляції:

Розмірність кореляції (CD), одна з найбільш широко використовуваних мір фрактальної розмірності, може розглядатися як міра кількості незалежних змінних, необхідних для визначення загальної системи у фазовому просторі. Для аналізу складності системи, як правило, здійснюється перехід від часової області до фазового простору.

Кожному вектору  $X_m^1(i)$  відповідає певна кількість векторів  $X_m^1(j)$ , для яких  $d(X_m^1(i), X_m^1(j)) \leq r$ , де  $r$  – порогове допустиме значення, що обчислюється наступним чином:

$$C_m^r(i) = \frac{1}{N-m+1} \sum_{j=i+1}^{N-m+1} RM_{ij}$$

де  $RM_{ij}$  – елементи матриці рекурентності (RM) розміром  $K \times K$ , які визначаються як:

$$RM_{ij} = \Theta(r - d(X_m^\tau(i), X_m^\tau(j)))$$

де  $i, j = 1, \dots, K$ ,

$r$  – порогова відстань,

$d$  – Евклідова відстань,

$\Theta(\cdot)$  – функція Хевісайда,

$\tau$  - вкладання відставання.

Евклідова відстань обчислюється так:

$$d(X_m^1(i), X_m^1(j)) = \sqrt{\sum_{k=1}^m (X_m^1(i, k) - X_m^1(j, k))^2}$$



де  $X_m^1(i, k)$  та  $X_m^1(j, k)$  відповідають  $k$ -тим елементам послідовностей  $X_m^1(i)$  та  $X_m^1(j)$  відповідно. Якщо двофазні просторові вектори  $X_m^r(i)$  та  $X_m^r(j)$  досить близько один до одного, тоді  $RM_i, j = 1$ ; інакше  $RM_i, j = 0$ .

Ймовірність того, що дві обрані точки близькі одна до одної на відстань, меншу ніж  $r$ , обчислюється середнім  $C_m^r(i)$  по  $i$ :

$$C_m^r = \frac{2}{N-m} \sum_{i=1}^{N-m+1} C_m^r(i)$$

Індекс  $C_m^r(i)$  обчислюється для зростаючих значень вбудованих розмірностей  $m$  (зазвичай тут вбудовані розмірності коливаються від 2 до 30). Grassberger та Procaccia показали, що CD можна отримати з наступного:

$$CD = \lim_{r \rightarrow 0} \lim_{N \rightarrow \infty} \frac{\log(C_m^r)}{\log(r)}$$

Градієнти кривих  $\log(r)$  та  $\log(C_m^r)$  визначаються з послідовності  $d(m)$ . По мірі збільшення  $m$ ,  $d(m)$  прямує до константної величини насичення, що є величиною CD. На практиці це граничне значення апроксимується нахилом кривої регресії  $(\log(r), \log(C_m^r))$ .

## 2. Алгоритм Барабасі та Стенлі (Barabasi and Stanley)

Головна особливість методу box-counting полягає в аналізі складних образів шляхом розбиття сигналу на все менші та менші частини, зазвичай обмежені формою квадрату (ящику, box-shaped), та аналізу частин в кожному масштабі. Мінімальна кількість елементів заданого розміру ( $\varepsilon$ ) необхідна для повного покриття прямої ( $S$ ) розмірності  $d$  обчислюється як  $N_\varepsilon$  :

$$N_\varepsilon(S) \sim \frac{1}{\varepsilon^d} \text{ при } \varepsilon \rightarrow 0$$

Оскільки розмір елемента наближається до нуля, загальна площа, яку охоплюють елементи площі, буде сходитись до міри кривої. Таким чином,  $FD_b$  можна оцінити за допомогою алгоритму підрахунку ящиків, як пропонують Барабасі та Стенлі, наступним чином:

$$FD_b = -\lim_{\varepsilon \rightarrow 0} \frac{\ln(N_\varepsilon(S))}{\ln(\varepsilon)}$$

### 3. Алгоритм Каца

Фрактальна розмірність  $FD_k$  часового сигналу за методом Каца обчислюється наступним чином:

$$FD_k = \frac{\log(L)}{\log(d)}$$

де  $L$  - загальна довжина кривої, обчислена як сума відстані між послідовними точками даних, а  $d$  - діаметр або плоска протяжність кривої, оцінена як відстань між першою точкою та точкою в послідовності, що дає найдальшу відстань. Для сигналів, які не перехрещуються, це можна виразити як:

$$d = \max_{i=2, \dots, N} |x(1) - x(i)|$$

### 4. Алгоритм Гігучі

Метод Гігучі є дуже ефективним алгоритмом для обчислення  $FD$  кривої, і він все частіше використовується для аналізу часових рядів. Для часового ряду, вираженого  $x(i)$ ,  $i = 1, \dots, N$ , нові послідовності  $Y_r^m$  отримують наступним чином:

$$Y_r^m = \left( x(m), x(m+r), x(m+2r), \dots, x\left(m + \left\lfloor \frac{N-m}{r} \right\rfloor r\right) \right), m = 1, \dots, r$$

де  $[.]$  – код Гауса - позначення для математичних вузлів. Він створюється шляхом перерахування та класифікації перетинів вкладання вузла в площину. Код Гауса представляє вузол із послідовністю цілих чисел. Наприклад, вузол трилисника в коді Гауса може бути поданий як: 1, -2,3, -1,2, -3. Де  $m$  та  $r$  є цілими числами, що позначають початковий час і інтервал часу відповідно.

Довжина нового ряду  $Y_r^m, L_m(r)$  обчислюється наступним чином:

$$L_m(r) = \left[ \left( \sum_{i=1}^{\lfloor \frac{N-m}{r} \rfloor} |x(m+ir) - x(m+(i-1)r)| \right) \times \frac{N-1}{\lfloor \frac{N-m}{r} \rfloor} \right] \frac{1}{r}$$

Довжина  $L(r)$  для інтервалу часу  $r$  отримується шляхом усереднення всіх довжин підряду  $L_m(r)$ , які були отримані для заданого значення  $r$ .

Якщо  $L(r)$  пропорційна  $r^{-D}$ , крива, що описує форму, є фракталоподібною з розмірністю  $D$ . Таким чином, якщо  $L(r)$  побудовано проти  $r$ , у подвійному логарифмічному масштабі ( $\ln(\frac{1}{r}), \ln(L(r))$ ), точки повинні падати на пряму лінію з нахилом, рівним  $-D$ . Коефіцієнт лінійної регресії графіку береться як оцінка  $FD$  епохи. Застосування зазначеного співвідношення передбачає правильний вибір максимального значення  $r$ , для якого зв'язок між  $L(r)$  і  $r^{-D}$  є приблизно лінійним.

### 2.3. Застосування фрактальної розмірності в кардіології

З часу коли було запропоновано перший алгоритм розрахунку фрактальної розмірності, понад 120 робіт було опубліковано, застосовуючи ці алгоритми до часових рядів серцебиття. Бекерс та ін. (Beckers et al.) мали на меті порівняти вплив статі, віку та зміни дня і ночі застосовуючи кілька нелінійних індексів ( $FD$

— алгоритм Каца, DFA, CD, LE та ApEn) до інтервалів часових рядів RR від здорового населення.

Їх висновки були наступними: значення були  $FD = 1,27 \pm 0,09$  (день для чоловіків);  $FD = 1,20 \pm 0,08$  (ніч для чоловіків);  $FD = 1,28 \pm 0,08$  (день для жінок); і  $FD = 1,22 \pm 0,08$  (ніч для жінок). CD обчислювали з використанням алгоритму, описаного раніше. Отримані значення були  $CD = 3,97 \pm 0,72$  (день для чоловіків);  $CD = 4,37 \pm 1,30$  (ніч для чоловіків);  $CD = 4,15 \pm 0,75$  (день для жінок); і  $CD = 4,41 \pm 1,29$  (ніч для жінок).

Автори встановили, що (1) всі нелінійні індекси представляють зміну день-ніч, за винятком CD у жіночої популяції; (2) гендерні відмінності існують лише в ApEn, DFA та LE; (3) всі нелінійні показники суттєво корелюють з віком в денний час. Протягом ночі зв'язок з віком зникав у деяких індексах, таких як DFA. (4) FD не був пов'язаний з лінійними показниками BCP, але корелював з усіма іншими нелінійними індексами, крім CD; та (5) CD лише помірно корелював із часткою високої частоти потужності, а також показав позитивні кореляції з ApEn і негативні кореляції з DFA.

Овіс та ін. (Owis et al.) успішно використовували CD в інтервалах RR для часових рядів з п'яти різних типів сигналів від MIT-BIH (Массачусетський технологічний інститут - Бостонська лікарня Бет Ізраїль). База даних аритмії: норма, шлуночковий куплет, шлуночкова тахікардія, шлуночковий бігемін і фібриляція шлуночків. CD (розрахований за алгоритмом Грассбергера та Прокаччі) був значно вищим у нормальній групі ( $CD = 3,27 \pm 0,42$ ) у порівнянні з кожною з патологічних.

Накамура та ін. (Nakamura et al.) вивчали зміну FD під час фізичних навантажень, зондуючи RR часовий ряд 10 суб'єктів. Індекс FD обчислювали з використанням степенної складової  $\beta$ . Більшість розрахованих значень для  $\beta$  були від 1 до 3 (що вказує на фрактальну динаміку); отже, індекс FD було розраховано як  $FD = 1 / (\beta - 1)$  для  $1 < \beta \leq 3$ . Зі збільшенням інтенсивності вправ зменшується парасимпатичний показник, що супроводжувалося зменшенням

FD. Під час легких фізичних навантажень значення були близько 3, які зменшились до рівня нижче 2.

Вон та ін. (Vaughn et al.) аналізували BCP у здорових дорослих під час сну за допомогою індексу FD Каца. Вони виявили, що FD суттєво відрізняється залежно від фази сну зі значеннями  $FD = 2,04 \pm 0,22$  (не сплячий);  $FD = 2,87 \pm 0,80$  (стадія 2);  $FD = 2,16 \pm 0,59$  (стадія 3/4); і  $FD = 2,43 \pm 0,40$  (швидкий рух очей).

Нарешті, Turcott et al. порівняли FD обчислені за допомогою алгоритму підрахунку ящиків через 15 інтервалів RR часових рядів здорових суб'єктів та 15 часових рядів пацієнтів із серцевою недостатністю. Вони показали, що у здорових пацієнтів значення FD становить  $2,75 \pm 0,20$  і що у пацієнтів із серцевою недостатністю показники були значно нижчими ( $FD = 2,17 \pm 0,29$ ).

## 2.4 Висновки до розділу

У другому розділі було розглянуто засоби нелінійного аналізу часових сигналів, до яких належать сигнали ЕКГ, а саме аналіз графіку Пуанкаре та розрахунок фрактальної розмірності. Було розглянуто найчастіше використовувані алгоритми розрахунку фрактальної розмірності, такі як розмірність кореляції, алгоритм Барабасі та Стенлі (box-counting), алгоритм Каца, алгоритм Гігучі.

Виконано огляд останніх досліджень в галузі кардіології, в яких застосовувалися методи розрахунку фрактальної розмірності кардіограм.

## РОЗДІЛ 3. ГЕНЕРАЦІЯ ЧАСОВИХ СИГНАЛІВ ЕКГ

### 3.1. Пацієнт-специфічне моделювання

В медичній діагностиці використовуються показники, отримані в результаті клінічних випробувань. Такі показники є усередненими та грубими, а отже, вони можуть бути непридатними до діагностики конкретного стану пацієнта (Kent and Hayward 2007).

Останні кілька років у дослідженнях комп'ютерних методів діагностики розглядаються пацієнт-специфічні методи моделювання, оскільки вони мають потенціал оптимізувати діагностику та підвищити її точність. Пацієнт-специфічне моделювання (Patient-specific modelling (PSM)) - це розробка обчислювальних моделей патофізіології людини, які індивідуалізовані до пацієнт-специфічних даних (Neal and Kerckhoffs 2010). Кількість досліджень, орієнтованих на пацієнта, зростає з експоненційною швидкістю. Так, за даними огляду Sack and Davies, 2016, кількість наукових публікацій, що присвячені дослідженню серцевих захворювань та містять «пацієнт-специфічний» у назві, в 2015 році у два рази перевищувала кількість досліджень, що базувались на узагальнених даних (рис. 3.1).

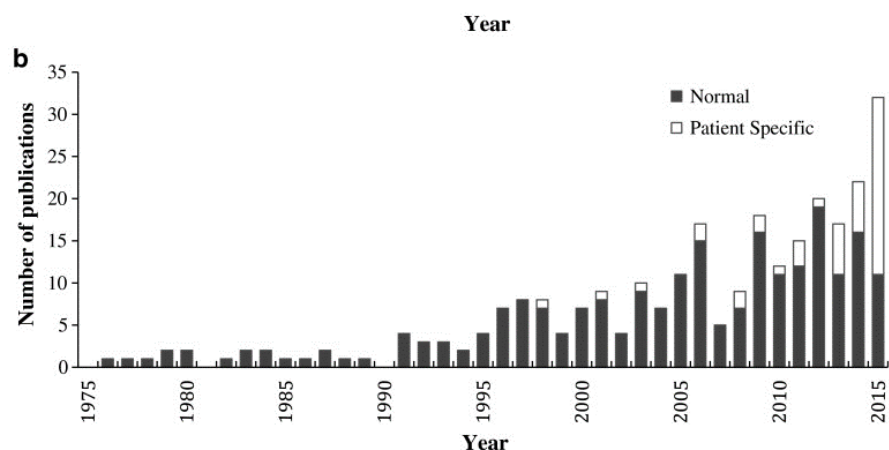


Рисунок 3.1 – Щорічна кількість публікацій, присвячених дослідженню серцевих захворювань, що базуються на а) узагальнених, б) пацієнт-специфічних даних

Сьогодні більшість методів класифікації ЕКГ зосереджені на застосуванні класичних контрольованих методів машинного навчання (supervised machine learning) - (Chazal and Reilly 2007; Ye, Kumar, and Coimbra 2012; Escalona-Moran et al. 2014) та ручних методах витягування ознак (manual feature engineering). Останнім часом моделі глибокого навчання (Al Rahhal et al. 2016) були успішно застосовані та досягли високих результатів, зменшивши потребу в ручних методах витягування ознак. Однак на практиці жоден із цих методів не зміг масштабувати різні ЕКГ різних типів пацієнтів (Kiranyaz, Ince та Gabbouj 2016). Характер динаміки ЕКГ сигналу та морфологічні характеристики суттєво відрізняються між пацієнтами і сильно залежать від фізичного стану пацієнта. Враховуючи високу варіативність між пацієнтами та внутрішню варіативність класифікації серцебиття для тих самих пацієнтів, побудова моделей глибокого навчання для розпізнавання ЕКГ, які можуть бути використані на практиці, є кількісно обмеженою.

Для масштабування глибокого навчання з метою персоналізованої класифікації ЕКГ є потреба у об'єктно-специфічних розмічених прикладах. Проте маркування достатньої кількості зразків ЕКГ для кожного пацієнта - незручне завдання. Для подолання проблеми розрідженості даних пропонується синтетично генерувати персоналізовані ЕКГ-сигнали різних аритмій, які мають аналогічні морфологічні характеристики до предмету класифікації. Надалі вони використовуються для навчання моделей глибокого навчання, які краще адаптуються до предмету класифікації.

### 3.2. Генеративно-змагальні мережі

Для вивчення пацієнт-специфічних сигналів застосовуються персоналізовані генеративно-змагальні мережі (PGAN). PGAN вчать генерувати персоналізовані ЕКГ-сигнали, характерні для аритмії, без потреби

предметного маркування ЕКГ. Це досягається шляхом навчання мережі генерувати ЕКГ, які імітують схеми деполяризації передсердь та шлуночків пацієнта та реполяризації під час специфічної аритмії: застосовується навчання без вчителя на кількахвилинному записі нерозміченого сигналу ЕКГ пацієнта.

Генеративно-змагальні мережі були представлені у 2014 році Яном Гудфеллоу та ін. (Ian J. Goodfellow et. al) у статті «Генеративно-змагальні мережі» (“Generative Adversarial Nets”).

Розглянемо архітектуру традиційної генеративно-змагальної мережі (рис. 3.2): вона складається з двох мереж, що змагаються – генератора  $G$  та дискримінатора  $D$ .  $G$  приймає вектор випадкових значень  $z$  в якості вхідних даних та генерує синтетичні дані, дискримінатор приймає реальні та згенеровані дані в якості вхідних даних та намагається їх відрізнити.

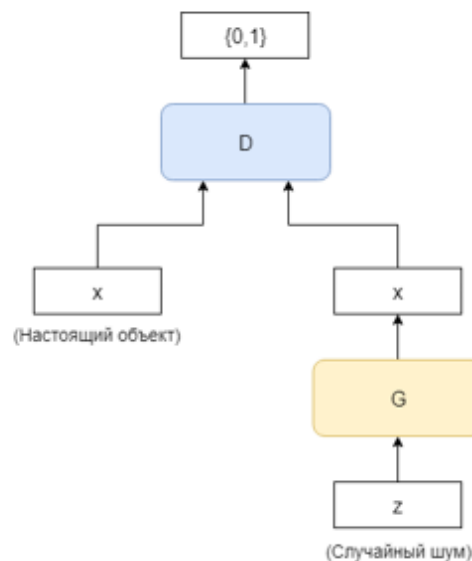


Рисунок 3.2 – Архітектура традиційної генеративно-змагальної мережі

Навчання  $G$  і  $D$  відбувається одночасно та спирається на припущення, що на кожній ітерації  $D$  підвищує якість розпізнавання синтетичних об'єктів, а  $G$  в свою чергу вимушений підвищувати якість генерованих об'єктів. Формально процес навчання можна описати як антагоністичну (некооперативну) гру з функцією виграшу  $V(D,G)$ :



$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{real}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z)))]$$

де  $p_{real}$  – розподіл реальних об'єктів, а  $p_z$  – розподіл випадкового шуму.

Дискримінатор в GAN є класифікатором. Він намагається відрізнити реальні дані від тих, що були створені генератором. Генератор може використовувати будь-яку мережеву архітектуру, що буде підходити до типу даних, які класифікуються.

Є два джерела тренувальних даних дискримінатора:

- Реальні дані, такі як сигнали ЕКГ пацієнтів. Дискримінатор використовує такі дані в якості позитивних прикладів під час тренування.
- Штучні дані, створені генератором. Дискримінатор використовує такі дані як негативні приклади під час тренування.

На рис. 3.3 блоки, позначені як Sample (зразки, приклади) представляють два джерела даних, які подаються на вхід дискримінатора. В той час, коли відбувається тренування дискримінатора, генератор не навчається. Його ваги залишаються константами, доки він створює приклади для навчання дискримінатора.

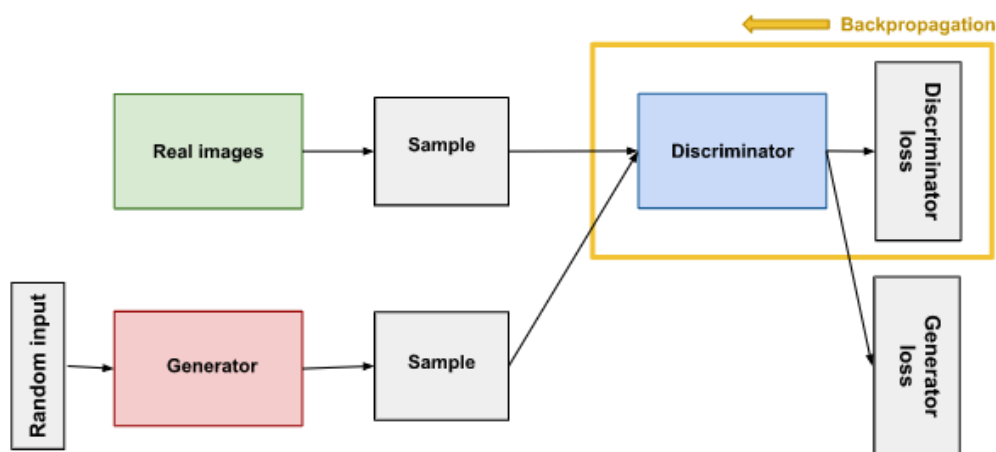


Рисунок 3.3 – Зворотнє поширення помилки під час тренування дискримінатора

Дискримінатор пов'язаний з двома функціями втрат. Під час навчання дискримінатор ігнорує функцію втрат генератора і використовує лише функцію

втрат дискримінатора. Функція втрат генератора використовується під час тренування генератора.

Отже, під час навчання дискримінатора:

1. Дискримінатор класифікує реальні та штучні дані з генератора.
2. Функція втрат дискримінатора штрафувє дискримінатор за неправильну класифікацію реальних даних як штучних та навпаки.
3. Дискримінатор оновлює ваги за алгоритмом зворотнього поширення помилки з функції втрат дискримінатора через мережу дискримінатора.

Генератор в GAN навчається створювати штучні дані шляхом включення зворотного зв'язку від дискримінатора. Він навчається змушувати дискримінатор класифікувати свої вихідні дані як реальні.

На рис. 3.4 зображено схему зворотного розповсюдження помилки при навчанні генератора.

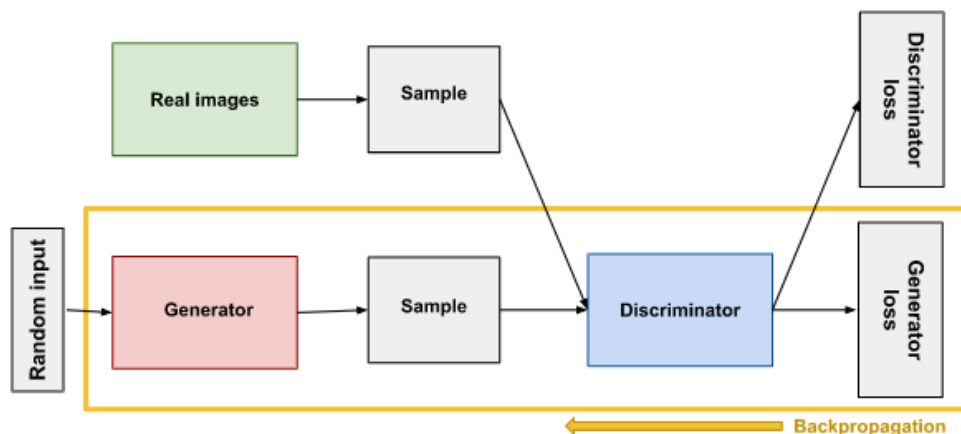


Рисунок 3.4 – Зворотнє поширення помилки під час тренування генератора

Навчання генератора потребує більш тісної інтеграції між генератором і класифікатором, ніж цього вимагає дискримінатор. Частина GAN, яка тренує генератор, включає:

- Випадковий вхід.
- Мережу генератора, яка трансформує випадковий вхід у приклад (зразок) даних.

- Мережу дискримінатора, яка класифікує згенеровані дані.
- Вихід дискримінатора.
- Функцію втрат генератора, яка штрафує генератор за невдалу спробу обдурити дискримінатор.

Випадковий вхід. Нейронні мережі потребують вхідні дані в деякому форматі. Зазвичай ми подаємо на вхід дані, які хочемо обробити, як наприклад дані, які хочемо класифікувати або зробити прогнозування. Проте генератор GAN отримує на вхід випадковий шум, після чого трансформує його значущі дані на виході. Вводячи шум, ми можемо змусити GAN генерувати широкий спектр даних з вибіркою з різних частин цільового розподілу.

Експерименти демонструють, що розподіл шуму не має великого значення, тому ми можемо обрати такий розподіл, з якого буде легко зробити вибірку, наприклад, рівномірний розподіл. Для зручності простір, з якого формується вибірка шуму, зазвичай має меншу розмірність, ніж розмірність вихідного простору.

Використання дискримінатора для навчання генератора. Для того, щоб навчити нейронну мережу, ми коригуємо її ваги, щоб зменшити помилку або функцію втрат її виходу. В мережі GAN генератор не має прямого зв'язку з функцією втрат, на яку ми хочемо впливати. Генератор подає сигнал в мережу дискримінатора, а дискримінатор надає вихідні дані, на які ми намагаємось впливати. Функція втрат генератора штрафує його за створення зразків даних, які дискримінатор класифікує як штучні.

Ця додаткова частина мережі має бути включена у зворотне поширення. Зворотне поширення коригує кожен вагу в правильному напрямку, обчислюючи вплив ваги на результат – як зміниться результат, якщо зміниться вага. Проте вплив ваги генератора залежить від ваги дискримінатора, на який він впливає. Таким чином, зворотне поширення сигналу починається на виході та повертається через дискримінатор до генератора. В той самий час, ми хочемо, щоб дискримінатор не змінювався під час тренування генератора.

Тренування генератора відбувається наступним чином:

1. Зразок випадкового шуму.
2. Створення виходу генератора зі зразка випадкового шуму.
3. Отримання класифікації дискримінатора (реальний/штучний сигнал) для виходу генератора.
4. Розрахунок функції втрат по класифікації дискримінатора.
5. Зворотне поширення через дискримінатор та генератор для отримання градієнтів.
6. Використання градієнтів для зміни вагів лише генератора.

### 3.2.1. Генеративно-змагальні мережі часових рядів

Генеративна мережа для синтезу часового сигналу, яким є і сигнал ЕКГ, має певні особливості. Так, задачею моделі є не лише фіксувати розподіл ознак в кожен момент часу, проте вона також має відображати потенційно складну динаміку цих змінних в часі. При моделюванні багатомірних послідовностей даних  $x_{1:T} = (x_1, \dots, x_T)$  ми прагнемо точно захопити умовний розподіл  $p(x_t|x_{1:t-1})$  переходів, залежних від часу.

Багато досліджень зосереджено на вдосконаленні часової динаміки авторегресійних моделей для прогнозування послідовностей. Вони в першу чергу вирішують проблему складання помилки під час багатокрокового створення зразків, вводячи різні модифікації часу навчання для більш точного відображення умов часу тестування. Авторегресійні моделі явно враховують розподіл послідовності у добуток умовних  $\prod_t p(x_t|x_{1:t-1})$ . Однак, хоча це корисно в контексті прогнозування, цей підхід є принципово детермінованим і не є справді генеративним у тому сенсі, що з його допомогою можна випадково відбирати нові послідовності без зовнішніх умов. З іншого боку, окремий напрямок роботи зосереджений на безпосередньому застосуванні генеративно-

змагальних мереж (GAN) для послідовних даних, насамперед шляхом створення рекурентних мереж в якості генератора та дискримінатора.

Розглянемо загальний підхід до генерації часових даних, де кожен екземпляр сигналу складається зі статичних ознак (що не змінюються з часом, напр. стать) та ознак, що залежать від часу. Нехай  $\hat{S}$  - векторний простір статичних ознак,  $\dot{X}$  - залежних від часу, і нехай  $S \in \hat{S}$ ,  $X \in \dot{X}$  випадкові вектори, які можуть набувати конкретних значень, позначених  $s$  та  $x$ . Розглянемо послідовності  $(S, X_{1:T})$  з деяким спільним розподілом  $p$ . Довжина  $T$  кожної послідовності також є випадковою змінною, розподіл якої - для зручності позначення - включено в  $p$ . В тренувальних даних, нехай окремі зразки індексуються  $n \in \{1, \dots, N\}$ , навчальний набір даних позначимо як  $D = \{(s_n, x_{n,1:T_n})\}_{n=1}^N$

Мета - використовувати навчальні дані  $D$  для вивчення щільності  $\hat{p}(S, X_{1:T})$ , яка найкраще наближається до  $p(S, X_{1:T})$ . Це мета високого рівня, і залежно від довжини, розмірності та розподілу дані може бути важко оптимізувати в стандартній системі GAN. Тому додатково пропонується використовувати авторегресійну декомпозицію:

$$p(S, X_{1:T}) = p(S) \prod_t p(X_t | S, X_{1:t-1}),$$

Даючи додаткову і простішу мету вивчення щільності  $\hat{p}(X_t | S, X_{1:t-1})$ , яка найкраще наближається до  $p(X_t | S, X_{1:t-1})$  у будь-який момент часу  $t$ .

### 3.2.2. Генеративно-змагальні мережі для синтезу специфічних для класу даних

Важливість створення достатньо різноманітних специфічних для класу даних спонукала адаптувати введений термін винагороди, зробивши його

специфічним для класу. Класи можуть бути визначені довільно і можуть залежати від використовуваного набору даних. У контексті аналізу електрокардіограм, наприклад, класи представляють різні захворювання серця.

$$\mathcal{L}_{DS} = -\mathbb{E}_c \left[ \mathbb{E}_{z_1, z_2} \left[ \frac{\|G(z_2|c) - G(z_1|c)\|}{\|z_2 - z_1\|} \right] \right]$$

де зовнішнє сподівання стосується всіх класів,

$G$  являє собою генераторну мережу,

$z_1$  і  $z_2$  представляють будь-які два вхідні вектори шуму, що належать до одного класу  $c$ .

Інтуїтивно ця умова винагороджує генератор відповідно до того, наскільки він чутливий до зміни вхідних даних. Екстремальний режим згортання, наприклад, призводить до нульової чутливості, оскільки однакові вихідні дані будуть генеруватися для двох різних вхідних сигналів шуму  $G(z_2|c) = G(z_1|c)$ . Таким чином повертається нульове значення винагороди. Така умова включається у моделі CGAN (Conditional Generative Adversarial Nets) з метою заохочення внутрішньокласової диверсифікації (рис. 3.5).

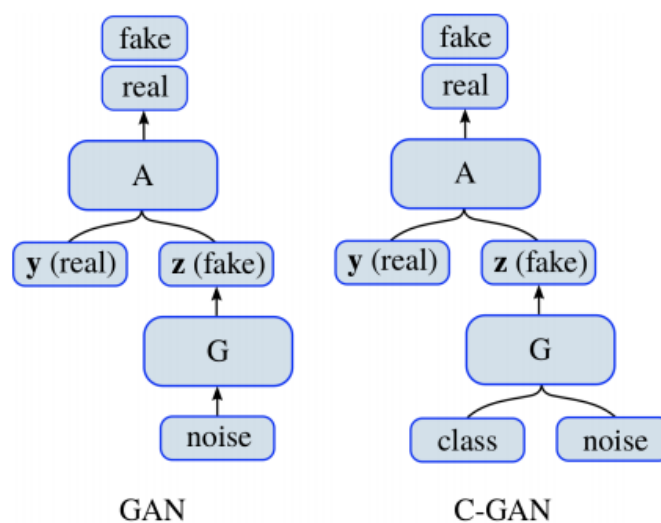


Рисунок 3.5 – Порівняння класичної генеративно-змагальної мережі (GAN) та умовної генеративно-змагальної мережі (CGAN)

Vanilla CGAN із чутливістю до специфічності: “Vanilla CGAN” включає умовну змінну в будь-яку точку всередині мережі генератора  $G$  та / або дискримінатора  $D$ . В роботі Kiyasseh et al., 2019 генератор навчався, використовуючи функції збитків, які складаються з трьох умов:

1. Функція втрат Дженсена-Шеннона  $\mathcal{L}_{JS}$  (Jensen-Shannon loss), яка штрафує мережу за генерування нереалістичних синтетичних даних  $x^\wedge$ .
2. Допоміжна функція втрат перехресної ентропії, яка штрафує мережу за генерування даних, які не можуть бути правильно класифіковані як істина  $k$ .
3. Клас-специфічна функція втрат чутливості до різноманітності, що штрафує мережу за генерування НЕ різноманітних синтетичних даних

$$\mathcal{L}_G = \mathcal{L}_{JS} - \mathbb{E}_{x^\wedge \sim P_g} [\log(p(y = k|x^\wedge))] + \lambda_{div} \mathcal{L}_{DS}$$

$$\mathcal{L}_{JS} = - \mathbb{E}_{x^\wedge \sim P_g} [\log(D(x^\wedge))]$$

де  $P_g$  – розподіл синтетичних даних,

$\lambda_{div}$  – гіперпараметр, який визначає рівень різноманітності чутливості.

Незалежно від генератора, дискримінатор навчається з використанням функції втрат, яка також складається з трьох умов:

1. Функція втрат Вассерштейна (Wasserstein loss function), яка штрафує мережу за класифікацію синтетичних даних як реальних та реальних даних як синтетичних.
2. Градієнтний штраф нуля для поліпшення стабільності тренувань.
3. Допоміжна функція втрат перехресної ентропії, яка штрафує мережу за неправильну класифікацію реальних даних:

$$L_D = \mathbb{E}_{x^\wedge \sim P_g} [D(x^\wedge)] + \mathbb{E}_{x \sim P_r} [D(x)] + \mathbb{E}_{x^*} \left[ \left\| \nabla_{x^*} D(x^*) \right\|^2 \right] - \mathbb{E}_{x \sim P_r} [\log(p(y=k|x))]$$

де  $P_r$  – розподіл реальних даних,

$\nabla$  - оператор градієнту,

$x^* = \alpha x + (1 - \alpha) \hat{x}$  - лінійна комбінація реальних та синтетичних даних,

$\alpha \sim U(0,1)$ .

### 3.3. Способи оцінювання синтетичних даних часових рядів

Після генерації синтетичних часових сигналів постає задача оцінки їх репрезентативності. Загальною проблемою для такого класу задач є режим згортання, коли генератор не може створити різноманітні зразки, тобто існує відображення типу багато-до-одного або багато-до-мало випадкової величини  $z$  на синтетичне зображення  $\hat{x}$ . Актуальною ця проблема є і в задачах синтезу сигналів ЕКГ, оскільки очікується генерація різноманітних даних.

Розглянемо деякі основні міри оцінки GAN, що прийнятні в задачах синтезу часових сигналів:

1. Репрезентативність синтетичних даних. Для вимірювання цього показника використовується максимальна середня невідповідність ядра (maximum mean discrepancy (MMD) – загальний метод оцінки GAN, за допомогою якого порівнюють подібність синтетичних та реальних даних. Кількісно ця подібність визначається функцією ядра  $K$ :

$$K(x, x') = e^{-\|x-x'\|^2}$$

де  $x$  та  $x'$  - два вектора, що порівнюються. Якщо вони ідентичні, то функція ядра оцінюється як одиниця. Чим більше вектори відмінні, тим меншою є величина, нижньою межею якої є нуль.



Оскільки оригінальна метрика MMD не може показати більш клас-специфічні подібності, пропонується застосовувати показник  $MMD_c$ , який дозволяє порівнювати клас-специфічну ефективність різних GAN:

$$MMD_c = \sum_{i \neq i'} K_{ii'} - 2 \sum_{i \neq j} K_{ij} + \sum_{j \neq j'} K_{jj'}$$

де  $K$  – функція ядра, що вимірює подібність між вхідними даними,

$$K_{ii'} = K(x_i^c, x_{i'}^c),$$

$$K_{ij} = K(x_i^c, x_j^c),$$

$c$  – певний клас,

$x$  та  $x^c$  - синтетичні та реальні дані відповідно.

2. Різноманітність в класах. Важливо виявити варіацію згенерованих даних в межах одного класу та між різними класами, оскільки вона допомагає оцінити умовну складову CGAN (Умовна Генеративно- Змагальна Мережа). Цей показник обчислюється через експоновані квадратичні ядра.

### 3.4. Застосування синтезованих даних в навчанні класифікатора

Особливість процесу навчання класифікатора з використанням датасету, розширеного синтетичними прикладами, полягає в тому, що тренування відбувається на змішаному наборі даних, який складається як з реальних, так і з синтезованих прикладів. Проте навчається класифікатор лише на реальних даних. Застосування такого підходу дозволяє проаналізувати ефективність використання синтетичних даних в задачах глибокого навчання.

Визначити правила розширення датасету можна наступним чином:

1. Вибір класу для дисбалансу.

2. Ступінь синтетичного дисбалансу.
3. Співвідношення синтетичних та реальних даних.

Таку перевірку можна виконати за допомогою алгоритмів К-Найближчих Сусідів, Наївний Баєс, лінійного та квадратичного дискримінантного аналізу, логістичної регресії, SVM, дерево прийняття рішень, випадковий ліс, багат шаровий перцептрон.

Математично для певної політики збільшення датасету та для всіх класифікаційних моделей  $M$  можна обчислити відсоткову зміну в метриці інтересу:

$$\% \Delta_M = \frac{X_{TSRTR} - X_{TRTR}}{X_{TRTR}} * 100$$

де  $X$  може бути будь-якою метрикою, наприклад AUROC,

$X_{TSRTR}$  – показник метрики на тренувальній вибірці, яка включає синтезовані зразки та реальні,

$X_{TRTR}$  - показник метрики на тренувальній вибірці, яка включає лише реальні зразки.

Описаний метод оцінки є обмеженим та просто описує продуктивність окремої моделі класифікації для певної політики розширення датасету. Для того, щоб отримати цілісну оцінку усіх моделей класифікації для всіх політик збільшення вибірки даних, а отже й оцінити більш реалістично будь-яку GAN, пропонується застосовувати криву синтетичного узагальнення (Synthetic Generalization Curve). Така метрика дозволяє обчислити, наскільки всі моделі класифікації  $M$  мають вищу чи нижчу продуктивність порівняно з середнім значенням. Точку на кривій, яка називається синтетичним узагальненням, можна обчислити наступним чином:

$$\forall \varepsilon SG(X, \varepsilon) = \frac{1}{M} \sum_{m=1}^M \delta(X_{TSRTR} \geq (1 - \varepsilon)X_{TRTR})$$

де  $\delta$  – дельта-функція Кронекера, яка оцінює одиницею аргумент, що є правдою, та нулем у випадку, якщо аргумент – неправда,

$SG$  – обчислюється для конкретної політики розширення набору даних  $p_i$  з множини політик  $P$ ,

$\varepsilon \neq 1$  – задає порівняння моделей класифікації, що тренувалися на розширеному датасеті  $X_{TSRTR}$ , з середнім показником  $(1 - \varepsilon)X_{TRTR}$ , та змінюється в залежності від вимоги користувача.

Наприклад, коли  $\varepsilon < 0$ , показник  $SG$  є відсотком класифікаційних моделей на розширеному датасеті, що переходять за середню лінію хоча б на  $-\varepsilon * 100$  відсотків.

З цієї кривої отримуємо нову метрику, що називається Площа Під Кривою Синтетичного Узагальнення (Area Under the Synthetic Generalization Curve or AUSG). Дану криву можна усереднити за багатьма політикамо розширення датасету, що дозволяє давати більш реалістичну оцінку продуктивності різних типів генеративно-змагальних мереж.

### 3.5. Висновки до розділу

У третьому розділі було розглянуто підхід пацієнт-специфічного моделювання в задачах медичної діагностики. Проведено аналіз актуальних досліджень даної теми, що проводились в останні роки, на основі чого зроблено висновок про ефективність використання специфічних даних для розв'язання задач класифікації електрокардіограм.

Розглянуто такий засіб генерації часових сигналів, як генеративно-змагальні мережі та їх застосування в задачах пацієнт-специфічного моделювання. Так, в розділі описано класичну архітектуру GAN, а саме архітектури генераторів, класифікаторів та алгоритм навчання мережі.

Наведено детальний опис архітектур генеративно-змагальних мереж, що використовуються в задачах синтезу часових рядів, в тому числі і сигналів ЕКГ.

Описано міри оцінки якості синтезованих зразків часових сигналів та особливості застосування згенерованих даних при навчанні моделей класифікації.

## РОЗДІЛ 4. ДОСЛІДЖЕННЯ МЕТОДІВ ПРИЙНЯТТЯ РІШЕНЬ В ЗАДАЧАХ КЛАСИФІКАЦІЇ ЕЛЕКТРОКАРДІОГРАМ

### 4.1. Глибинне навчання в задачах класифікації ЕКГ

У багатьох випадках традиційна діагностика, заснована на правилах, неефективна через обробку великої кількості різнорідних даних і вимагає значного аналізу та медичної експертизи для досягнення належної точності діагностики. Проблема стає більш вираженою в місцях, де бракує медичних експертів та клінічного обладнання, особливо в країнах, що розвиваються. Це породжує вимогу до надійної, автоматичної та дешевої системи моніторингу та діагностики. Ця вимога стає все більш затребуваною у постачальників медичних послуг, так що відповідні медичні оцінки можуть бути пов'язані з використанням комп'ютерних систем діагностики з автоматизованою діагностикою (CADS). Такі системи складаються з автоматичних процедур моніторингу стану здоров'я, що працюють на основі аналізу фізіологічних сигналів для моніторингу та оцінки функціональності відповідного органу. Вони надають людям портативні та прості рішення для інформування їх про їхні захворювання.

Отже, через постійний дефіцит підготовлених кардіологів та при широкому використанні домашніх ЕКГ для моніторингу пацієнтів із серцевими вадами, виникла необхідність розробки повністю автоматизованих механізмів класифікації ЕКГ.

Класифікація відноситься до задач машинного навчання та полягає в тому, що маючи набір розмічених даних, тобто таких, де відома належність кожного об'єкту до певного класу, можна навчити алгоритм, який буде визначати клас для об'єктів, які не мають попередньо визначеної мітки класу.

Формально алгоритм класифікації можна описати наступним чином. Дано:

1. Множина об'єктів  $T = \{t_1, t_2, \dots, t_n\}$  – навчальна вибірка,
2.  $t_i \rightarrow \{x_1, x_2, \dots, x_m, y\}$ ,
3.  $X = \{x_1, x_2, \dots, x_m\}$  – незалежні змінні (атрибути),

4.  $C_h = \{c_{h1}, c_{h2}, \dots, c_{hq_h}\}$  - значення, які приймає  $x_h$ ,
5.  $y$  – залежна змінна,
6.  $V = \{v_1, v_2, \dots, v_s\}$  – значення, які приймає  $y$ .

Знайти: значення залежної змінної  $y$  при умові, що незалежні змінні набуватимуть нових значень.

В даній роботі постановка задачі класифікації може бути визначена так: навчальну вибірку визначити як множину векторів ознак кардіограм пацієнтів з різними захворюваннями серця, де класами є види серцевих вад. Побудувати класифікатор, що визначатиме, якому класу захворювання належить сигнал кардіограми, поданої на вхід.

Глибинні нейронні мережі (DNN) широко використовуються для класифікації та прогнозування в різних сферах. Останнім часом DNN почали стрімко розвиватись, суттєво впливаючи на точність класифікації для широкого кола медичних завдань. Сучасні системи CADs використовують DNN для виявлення аритмії захопленого ЕКГ-сигналу, що веде до зниження вартості постійного моніторингу серця та поліпшення якості прогнозів.

Однак автоматизована класифікація аритмії на основі ЕКГ, як правило, стикається з декількома важливими проблемами:

1. Симптоми аритмії можуть не спостерігатися протягом періоду захоплення ЕКГ-сигналу (Ceylan & Özbay, 2007).
2. Властивості ЕКГ-сигналу (такі як період та амплітуда) відрізняються від людини до людини і залежать від різних факторів, таких як вік, стать, фізичні умови та спосіб життя. Знайти узагальнену структуру разом із відповідними стандартами для функціонування для загальної популяції проблематично (Ceylan, Özbay, 2007, Joshi, Chandran, Jayaraman, Kulkarni, 2009).
3. Морфологія ЕКГ-сигналу часто не є стаціонарною навіть для одного пацієнта, з урахуванням фізичного стану (біг, ходьба та сон).
4. Обсяг даних, що враховуються для аналізу сигналів ЕКГ, великий. Отже, є більша ймовірність помилкового діагнозу аритмія.

5. Шум, артефакти та перешкоди можуть призвести до морфологічних варіацій та розбіжностей у захопленому сигналі ЕКГ (Adams, Choi, 2012, Dinakarrao, Jantsch, Shafique, 2019).

До задач глибинного навчання відносяться інтелектуальне прийняття рішень, прогнозування або інші засоби розпізнавання складних образів, використовуючи набори тестових даних. Порівняно з традиційними техніками навчання, мережі глибинного навчання є більш масштабованими, оскільки більша точність зазвичай досягається збільшенням розміру мережі або навчального набору даних. Моделі неглибокого навчання, такі як дерева рішень та підтримка векторних машин (SVM), неефективні для багатьох сучасних застосувань, це означає, що вони вимагають великої кількості спостережень для досягнення узагальнення та накладення значної людської праці для уточнення попередніх знань у моделі (Goodfellow, Bengio, Courville, 2016, Loni, Sinaei, Zoljodi, Daneshtalab, Sjödin, 2020).

В останні роки було запропоновано кілька моделей глибокого навчання (DL) для підвищення точності різних навчальних завдань, включаючи багат шаровий перцептрон (MLP), згорткову нейронну мережу (CNN), рекурентну нейронну мережу (RNN), довгу короткочасну пам'ять (LSTM) та Deep Belief Network (DBN).

У таблиці 4.1 наведено порівняння моделей класифікаторів глибинного навчання, які було реалізовано за останні кілька років, та які мають одні з найвищих показників точності та чутливості розпізнавання.

Таблиця 4.1 – Порівняння існуючих систем класифікації ЕКГ

Роботи	Рік	Класи	Методи	ACC, %	P, %	S, %
Oh et al.	2018	5 типів ударів серця	CNN-LSTM	98.10	97.50	98.70

Продовження таблиці 4.1

Pandey et al.	2019	5 типів ударів серця	CNN	98.30	95.51	86.06
Yildirim et al.	2019	5 типів ударів серця	LSTM	99.23	99.00	99.00
Gao et al.	2019	8 типів ударів серця	LSTM, FL	99.26	99.26	99.14
Zhang et al.	2019	18 типів ударів серця	CNN	98.27	60.93	99.95

Можна зробити висновок, що найкращу точність в дослідженнях показали саме мережі з довгою короткочасною пам'яттю (Long Short Term Memory (LSTM)), з невеликою перевагою показавши кращі результати, ніж згорткові нейронні мережі (Convolutional Neural Networks (CNN)). Отже, розглянемо архітектуру мереж з довгою короткочасною пам'яттю та рекурентних мереж, що передували появі перших, а також архітектуру згорткових мереж.

#### 4.1.1. Рекурентні нейронні мережі

Рекурентні нейронні мережі (Recurrent Neural Networks (RNN)) були представлені у 80-х роках як засіб моделювання часових рядів, оскільки традиційні підходи, засновані на мережах прямого поширення (Feedforward Neural Network (FNN)) не були доречними для розв'язання задач цього класу.



Мережі прямого поширення працюють за припущенням, що кожен елемент вхідних даних є незалежним від інших. Оскільки інформація поширюється лише в прямому напрямку в кожному шарі мережі, вона не враховує інформацію, що міститься в порядку введення.

Останнє має вирішальне значення для таких завдань, як регресія часових рядів, неоднозначність сенсу слова та взагалі будь-яка проблема, де вхідними даними є впорядкована послідовність. Наприклад, якщо ми хочемо передбачити ціну акцій Netfilx у січні 2021 року, потрібно взяти до уваги послідовність курсів акцій у попередні місяці: грудень 2020, листопад 2020 тощо (часова залежність). Якщо нам потрібно розрізнити значення слова, необхідно оцінити позицію цього слова у фразі (просторова залежність). У конкретному випадку виявлення аритмії потрібно зафіксувати просторову залежність серед зразків, щоб виявити такі особливості, як комплекс QRS, сегмент ST.

Ті самі залежності, які має на меті охопити РНМ, характеризують і навчальний процес людини. Так, ми вивчаємо послідовність, а не лише символи: для нас нескладно одразу сказати алфавіт від А до Я, але в зворотному напрямку ця задача вже не є простою. Те саме стосується номера нашого телефону. Для моделювання залежностей RNN має внутрішню пам'ять, яка постійно оновлюється через послідовність введення (рис. 4.1).

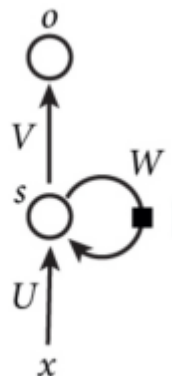


Рисунок 4.1 - Загальна структура рекурентної нейронної мережі

Інтуїтивно чітке визначення залежності послідовності в RNN полягає в тому, щоб подавати кожен шар з попереднім через зворотну силу.

Таким чином, зразок  $x_t \in S = [x_1, \dots, x_T]$ , подається в RNN для отримання вихідних даних  $o_t$  і прихованого стану  $s_t$ , що буде враховано мережею для наступного зразку  $x_{t+1}$ . Цю структуру можна розгорнути у декількох блоках, щоб видалити цикл, як показано на рис. 4.2: таким чином стає зрозуміло, як залежить вихідний результат  $o_t$  від поточного входу  $x_1$  і попереднього стану:  $s_{t-1}$ .

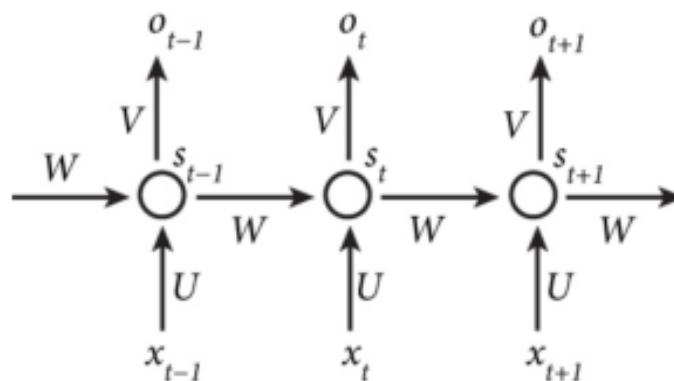


Рисунок 4.2 – Розгорнута структура РНМ

Формально ми маємо:

1.  $x_t$  є  $t$ -м входом послідовності
2.  $s_t$  - це прихований стан мережі, який постійно оновлюється таким чином:  $s_t = \phi(Ux_t + Ws_{t-1})$ , де  $\phi$  - загальна функція активації (ReLU, softmax, tanh). Слід звернути увагу на залежність від попереднього стану  $s_{t-1}$ : саме тут залежність послідовності робиться явною.
3.  $y_t^* = o_t$  кінцевий вихід, обчислений як:  $y_t^* = o_t = \phi(Vs_t)$

Важливо відзначити, що матриці:  $U$ ,  $V$ ,  $W$  однакові для кожного розгорнутого блоку, на відміну від класичної FNN, де кожен шар має свої параметри. Це є позитивним в розрізі меншої кількості параметрів для

нейронної мережі, проте в той же час призводить до певних проблем при обчисленні градієнту, а саме його затухання.

Розглянемо проблему затухання градієнту більш детально. Для того, щоб оцінити якість рекурентної нейронної мережі, визначається функція втрат  $L$  так, що для заданої послідовності розміру  $T$  загальна помилка визначається як сума помилок над усіма передбаченнями в послідовності:

$$E = \sum_{t=0}^T L(y_t^*)$$

Градієнт помилки щодо будь-якого з параметрів матриці, таких як  $W$ :

$$\frac{\partial E}{\partial W} = \sum_{t=0}^T \frac{\partial L(y_t^*)}{\partial W}$$

Розглянемо  $\frac{\partial L(y_t^*)}{\partial W}$ : ми маємо  $y_t^* = f(s_t)$  та  $s_t = g(W, s_{t-1})$ , отже:

$$\frac{\partial L(y_t^*)}{\partial W} = \sum_{k=0}^t \frac{\partial L(y_t^*)}{\partial y_t^*} \frac{\partial y_t^*}{\partial s_t} \frac{\partial s_t}{\partial s_k} \frac{\partial s_k}{\partial W}$$

Далі положимо  $s_j \in D^1$  та  $W \in D^1$ , тобто РНМ має розмірність 1. Проте так само можна розглядати багатовимірний випадок.

Дано:  $s_j = \phi(U_{x_t} + W_{s_{j-1}})$ . Маємо:

$$\frac{\partial s_j}{\partial s_{j-1}} = \frac{\partial(U_{x_t} + W_{s_{j-1}})}{\partial s_{j-1}} \frac{\partial(\phi(U_{x_t} + W_{s_{j-1}}))}{\partial(U_{x_t} + W_{s_{j-1}})} = W \phi'(U_{x_t} + W_{s_{j-1}})$$

Умову  $\frac{\partial s_t}{\partial s_k}$  можна розписати за правилом ланцюжка як:

$$\frac{\partial s_t}{\partial s_k} = \prod_{j=k+1}^t \frac{\partial s_j}{\partial s_{j-1}} = W^{t-k} \prod_{j=k+1}^t \phi'(U_{x_t} + W_{s_{j-1}})$$

Саме тут виникає проблема затухаючого градієнту.

При умові  $|W| > 1$  або  $|W| < 1$ ,  $W^{t-k}$  буде експоненційно вибухати або затухати. Так само  $|\phi'(U_{x_t} + W_{s_{j-1}})| \leq 1$  у випадку ReLU, tanh, softmax.

Ці два останні міркування дозволяють зробити висновок, що ще два віддалені вище стани  $t - k$  і  $\frac{\partial s_t}{\partial s_k}$  стануть надзвичайно великими (вибухаючий градієнт) або невеликими (затихаючий градієнт).

У першому випадку ваги стануть настільки великими, що призведуть до переповнення та значень NaN в результаті, модель буде нестабільною і не зможе узагальнити що-небудь з даних.

У другому випадку внесок попереднього стану для поточного буде неактуальним: модель не може навчитися довгостроковій залежності.

#### 4.1.2. Мережі LSTM

Мережі з довго та короткочасовою пам'яттю були винайдені Hochreiter і Schmidhuber в 1997 році для розв'язання проблеми затухаючого градієнту і встановили рекорди з точності у вирішенні серії задач, що належать до різних областей застосування.

Вони складаються з клітинного стану  $c_t$ , який оновлюється за допомогою структур, які називають воротами. Останні складаються з сигмоїди, яка визначає обсяг інформації, яка передається в клітину: нульові значення означають нічого, одиничні - все. Загалом, LSTM містить троє воріт в такому порядку: забуття, вхідні та вихідні ворота. На додаток до цього, кожна розгорнута комірка LSTM

бере як вхід фактичний зразок  $x_t$ , попередній стан комірки  $c_{t-1}$  та прихований стан  $h_{t-1}$ .

Спочатку ворота забуття вирішують, яку частину  $x_t$  і  $h_{t-1}$  нам потрібно відкинути для оновлення стану комірки. Зокрема, обчислюється  $f_t$  - вектор значень в діапазоні:  $[0,1]$ , що вказує на величину кількості інформації, що стосується попереднього прихованого стану та поточного входу, що не враховується (рис. 4.3):

$$f_t = \sigma(W_f[h_{t-1}; x_t] + b_f)$$

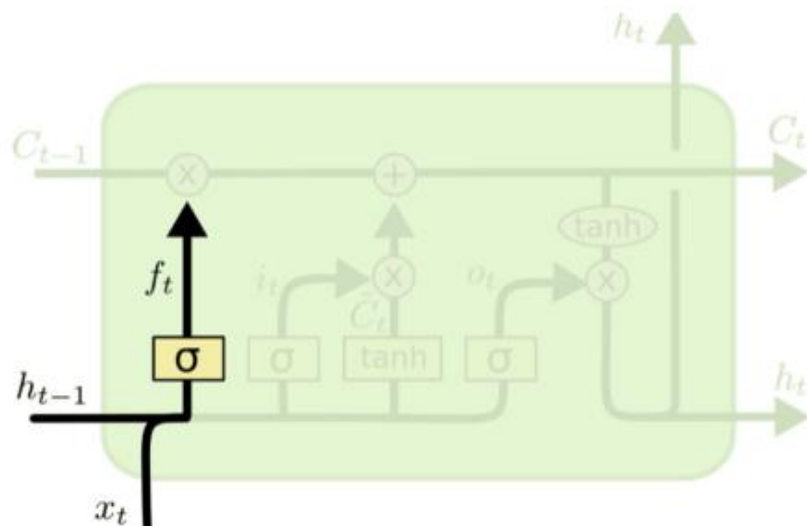


Рисунок 4.3 – Шар забування в LSTM

Симетричним способом вхідні ворота визначають значення об'єднаного вектора для зберігання:

$$i_t = \sigma(W_i[h_{t-1}; x_t] + b_i)$$

По-друге, завжди на цьому рівні будується стан комірки-кандидата (Рис. 4.4):

$$\hat{C}_t = \tanh(W_c[h_{t-1}; x_t] + b_c)$$

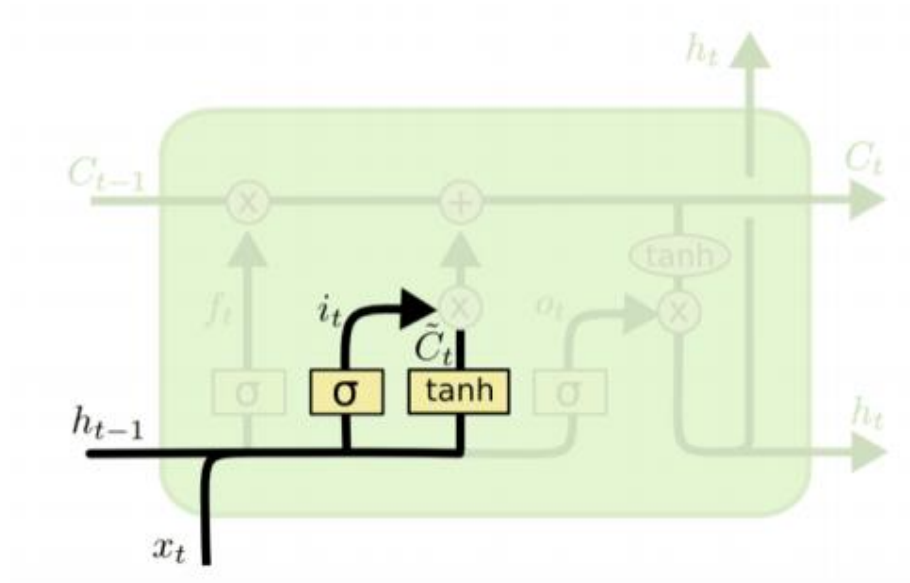


Рисунок 4.4 – Вхідний шар в LSTM

Тепер зрозуміло, як оновити стан комірки (рис. 4.12):

$$C_t = f_t * C_{t-1} + C_t * i_t$$

Нарешті, вихідні ворота видають результат:  $o_t = \sigma(W_o[h_{t-1}; x_t] + b_o)$  і прихований стан  $h_t = o_t * \tanh(C_t)$  (рис. 4.5):

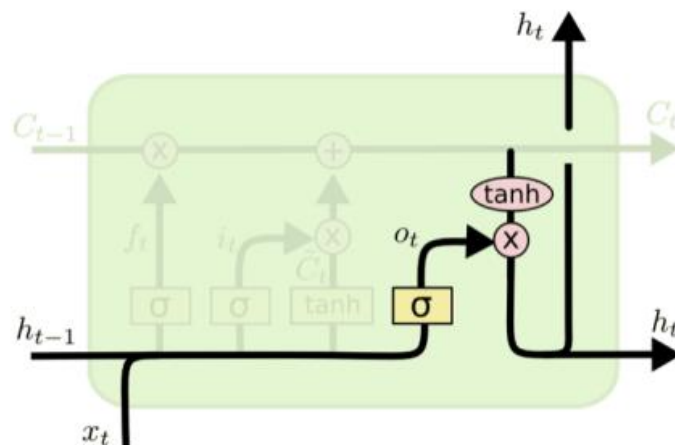


Рисунок 4.5 – Прихований стан та вихід в LSTM

Обговорювана структура є розв'язком проблеми затухаючого градієнту. Насправді, якщо ми обчислимо зворотній градієнт у цьому випадку маємо:

$$\frac{\partial C_t}{\partial C_{t-1}} = f_t + K$$

де  $K$  – результат взяття інших часткових похідних, які для нашого випадку не представляють інтересу.

Найважливішим моментом тут є те, що навіть у випадку довгострокової залежності, де ми множимо похідну, раніше обчислену кілька разів,  $f_t$  може завжди приймати будь-які значення в діапазоні  $[0,1]$ , і це не експоненційно збігається до нуля.

На додаток до цього наявність воріт з внутрішніми параметрами, а не простими функціями активації, дозволяє моделі адаптивно встановлювати відповідні параметри відносно поточного вхідного та прихованого стану. Таким чином ворота можуть активно оновлювати свій стан, коли градієнт занадто низький або занадто високий.

Варто зауважити, що існує кілька різновидів описаних архітектур. У 2002 році було запропоновано модель LSTM, де всі троє воріт:  $G_1$ ,  $G_2$ ,  $G_3$  зумовлені станом клітини, а також прихованим станом:

$$G_i = \sigma(W_i[C_{t-1}; h_{t-1}; x_t] + b_i)$$

Ще однією варіацією, яка нещодавно набула популярності, є воротний рекурентний блок (Gated Recurrent Unit (GRU)). В ньому вхідний затвор також використовується як ворота забування:

$$f_t = 1 - i_t$$

На додаток до цього клітинні та приховані стани об'єднуються разом.

Загалом, ми маємо:

1.  $i_t = \sigma(W_f[h_{t-1}; x_t])$ , що запам'ятовує попередній прихований стан та вхід
2.  $r_t = \sigma(W_r[h_{t-1}; x_t])$ , що залишається від попереднього прихованого шару
3.  $\hat{C}_t = \tanh(W[r_t * h_{t-1}; x_t])$  – кандидат прихованого стану (відмінний від комірки-кандидата в звичайній LSTM)
4.  $h_t = (1 - i_t) * h_{t-1} + i_t * \hat{C}_t$  - прихований стан, що будується шляхом забуттям про попередній прихований стан та додаванням того, що слід пам'ятати про кандидата прихованого стану.

#### 4.1.3. Згорткові нейронні мережі

**Згорткові нейронні мережі** (англ. *convolutional neural network, CNN*) можуть бути застосовані в задачах розпізнавання та класифікації. Структура мережі - односпрямована (без зворотних зв'язків), принципово багатощарова.

ЗНМ включає в себе такі види шарів: згорткові (convolutional) шари, шари субдискретизації (subsampling, підвибірка), а також шари «звичайної» нейронної мережі - персептрона (рис. 4.6).

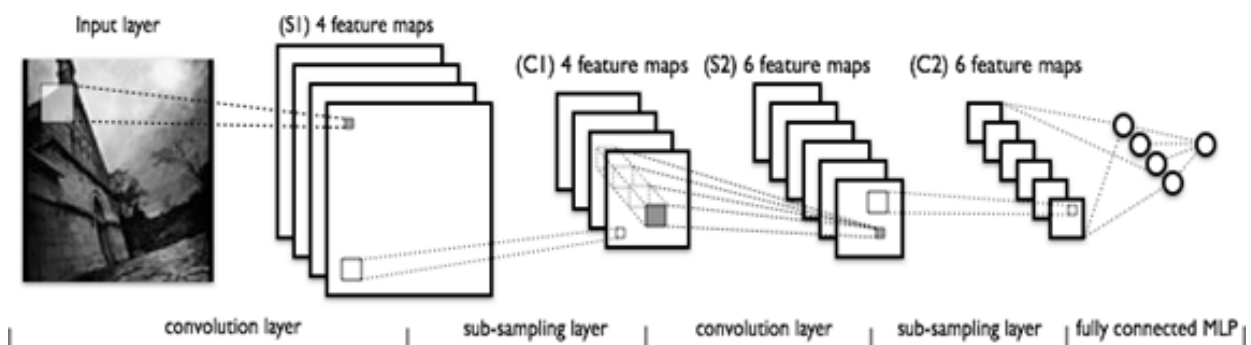


Рисунок 4.6 - Будова згорткової нейронної мережі

Вхідними даними є зображення. У випадку, якщо вони кольорові, то кожне представляється трьома каналами кольорів: червоний, синій, зелений.



Вхідні дані кожного значення пікселя нормалізуються від 0 до 1, для чого може бути використана формула:

$$f(p, min, max) = \frac{p - min}{max - min}$$

де  $f$  – функція нормалізації,

$p$  – відповідає значенню певного кольору пікселя від 0 до 255,

$min$  – мінімальне значення пікселя, що відповідає нулю,

$max$  – максимальне значення пікселя – 255.

Розмірність карт на згортковому шарі є однаковими та можуть бути обчислені за формулою:

$$(w, h) = (mW - kW + 1, mH - kH + 1),$$

де  $(w, h)$  – розмір згорткової карти, який обчислюється,

$mW$  – ширина попередньої карти,

$mH$  – висота попередньої карти,

$kW$  – ширина ядра,

$kH$  – висота ядра.

Згортковий шар може бути виражений формулою:

$$x^l = f(x^{l-1} * k^l + b^l)$$

де  $x^l$  – вихід шару  $l$ ,

$f()$  – активаційна функція,

$b^l$  – коефіцієнт зсуву для  $l$ -го шару,

$*$  – операція згортки входу  $x$  зі значенням ядра  $k$ .

Розмір вихідних матриць зменшується за рахунок крайових ефектів, що формульно може бути представлено так:

$$x_j^l = f\left(\sum_i x_j^{l-1} * k_j^l + b_j^l\right)$$

де  $x_j^l$  – карта ознак  $j$  (вихід шару  $l$ ),

$f()$  – активаційна функція,

$b_j^l$  – коефіцієнт зсуву шару  $l$  для карти ознак  $j$ ,

$k_j^l$  – ядро згортки  $j$  карти шару  $l$ ,

\* - операція згортки входу  $x$  з ядром  $k$ .

Pooling шар можна описати наступною формулою:

$$x^l = f(a^l * \text{subsample}(x^{l-1}) + b^l)$$

де  $x^l$  - вихід шару  $l$ ,

$f()$  – функція активації,

$a^l, b^l$  – коефіцієнти зсуву шару  $l$ ,

$\text{subsample}()$  - операція вибірки локальних максимальних значень.

Два типи шарів (convolutional, subsampling), чергуючись між собою, формують вхідний вектор ознак для багат шарового перцептрона.

Звичайний багат шаровий перцептрон призначений для класифікації, він моделює складну нелінійну функцію. Якість розпізнавання може бути підвищена шляхом оптимізації цієї функції.

Нейрони кожної карти попереднього підвибіркового шару пов'язані з одним нейроном прихованого шару.

Таким чином кількість нейронів прихованого шару дорівнює кількості карт підвибіркового шару, але зв'язки можуть бути не обов'язково такими.

Обчислення значень нейрона можна описати формулою:

$$x_j^l = f\left(\sum_i x_i^{l-1} * w_{i,j}^{l-1} + b_j^{l-1}\right)$$

де  $x_j^l$  – карта ознак  $j$  (вихід шару  $l$ ),

$f()$  – функція активації,

$b_j^l$  – коефіцієнт зсуву шару  $l$  для карти ознак  $j$ .

На сьогоднішній день кращі результати в розпізнаванні зображень отримують з допомогою ЗНМ. В середньому точність розпізнавання таких мереж перевершує звичайні ШНМ на 10-15%. ЗНМ - це ключова технологія Deep Learning.

#### 4.2. Критерії якості оцінки рішення задачі класифікації

Розглянемо основні методи оцінки якості роботи класифікаційних моделей: чисельна оцінка якості, точність та повнота, матриця неточностей.

Чисельна оцінка якості алгоритму (Accuracy) представляє відсоток прикладів, для яких класифікатор визначив правильний клас:

$$Accuracy = \frac{P}{N}$$

де  $P$  - кількість прикладів, що розпізнані правильно;

$N$  - розмір навчальної вибірки.

Особливістю цієї метрики є те, що вона присвоює всім прикладам однакову вагу. У випадку якщо розподіл прикладів у вибірці зміщений до певного класу, дана метрика не буде коректною, оскільки класифікатор матиме більше інформації саме по цих класах, і лише для них прийматиме більш правильні рішення.

Одним зі способів подолання цієї проблеми є балансування вибірки, на якій навчається класифікатор. Інший спосіб полягає в зміні підходу до формальної оцінки якості.

Точність і повнота:

Точність системи в межах класу - це частка прикладів, що дійсно належать даному класу щодо всіх прикладів, віднесених системою до цього класу.

Повнота системи - це частка визначених класифікатором прикладів, які належать певному класу відносно всіх прикладів даного класу в тестовому наборі даних.

Для обчислення мір точності та повноти необхідні наступні величини:

- TP - істино-позитивне рішення;
- TN - істино-негативне рішення;
- FP - хибно-позитивного рішення;
- FN - хибно-негативне рішення.

Тоді, точність і повнота калькуються так:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Значення точності і повноти зручно розраховувати за допомогою матриці неточностей (confusion matrix). Матриця неточностей - це матриця розміру N на N, де N - це кількість класів. Стовпці цієї матриці відповідають експертним рішеннями, а рядки - рішеннями класифікатора. Під час класифікації прикладів з тестової вибірки числа, що стоїть на перетині рядка класу, який повернув класифікатор і стовбця класу до якого дійсно відноситься документ, сумуються. На рисунку 4.7 зображено структуру матриці неточностей.

		Predicted class	
		<i>P</i>	<i>N</i>
Actual Class	<i>P</i>	True Positives (TP)	False Negatives (FN)
	<i>N</i>	False Positives (FP)	True Negatives (TN)

Рисунок 4.7 - Структура матриці неточностей

За допомогою такої матриці легко розраховується точність і повнота для кожного класу. Точність дорівнює відношенню відповідного діагонального елемента матриці і суми всього рядка класу. Повнота - відношенню діагонального елемента матриці і суми всього стовпчика класу.

Результуюча точність класифікатора розраховується як арифметичне середнє його точності по всіх класах. Те ж саме з повнотою. Технічно цей підхід називається *macro-averaging*.

### 4.3. Висновки до розділу

У другому розділі було розглянуто застосування глибинних нейронних мереж в задачах розпізнавання ЕКГ. Виконано огляд та порівняльний аналіз існуючих систем класифікації, описано основні проблеми автоматизованої класифікації аритмії на основі ЕКГ.

Було розглянуто такі засоби класифікації, як рекурентні нейронні мережі, LSTM мережі та згорткові нейронні мережі. Детально описано архітектури кожного типу мереж.

Було розглянуто і наведено основні критерії оцінки якості розв'язку поставленої задачі розпізнавання кардіограм, серед яких чисельна оцінка якості алгоритму, точність і повнота, матриця неточностей.

Як результат аналізу розглянутих мереж, було обрано мережі LSTM як класифікатор в даній роботі, з огляду на особливості архітектури таких мереж, що враховують специфіку розв'язуваної задачі.

## РОЗДІЛ 5. ОПИС ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

### 5.1. Опис вхідних даних

База даних, яка розглядається в цьому дослідженні, являє собою колекцію анотованих записів ЕКГ, отриманих лабораторією аритмії бостонської лікарні Бет-Ізраїль, що відома як база даних аритмії МІТ-ВІН. Вона містить 48 півгодинних двоканальних амбулаторних записів ЕКГ, отриманих від 47 досліджуваних пацієнтів у лабораторії аритмії ВІН між 1975 і 1979 рр.

Записи мають частоту 360 Гц: це означає, що кожен запис приблизно складається з  $30 \cdot 360 = 10800$  зразків. Кардіологи визначили загальну кількість у 110 000 анотацій, що входять до бази даних (анотація розміщується відповідно до піку R одиничного удару). Таким чином неявно вирішена для цього конкретного випадку проблема виявлення ударів.

Згідно до стандарту Асоціації удосконалення медичного обладнання (Association for the Advancement of Medical Instrumentation (AAMI)) для тестування та звітності про результати роботи алгоритмів вимірювання серцевого ритму та сегментації серцевих ударів, оцінка алгоритму детектора аритмії повинна бути складена на основі класифікаційних характеристик за п'ятьма основними категоріями серцебиття:

1. N: включає удари, що походять із синусового вузла (нормальні удари).
2. S: включає надшлуночкові ектопічні удари (SVEB).
3. V: включає шлуночкові ектопічні удари (VEB).
4. F: включає удари, які є результатом злиття нормальних та VEB.
5. Q: включає будь-яке серцебиття, що не входить до категорій N, S, V або F, по суті невизначені серцебиття, а також удари в результаті використання кардіостимулятора. На рисунку 5.1 зображено зразки кожного типу ударів серця.

Отже, в наборі даних містяться як нормальні сигнали ЕКГ, так і ряд сигналів ЕКГ пацієнтів, що мають певні відхилення в роботі серця.

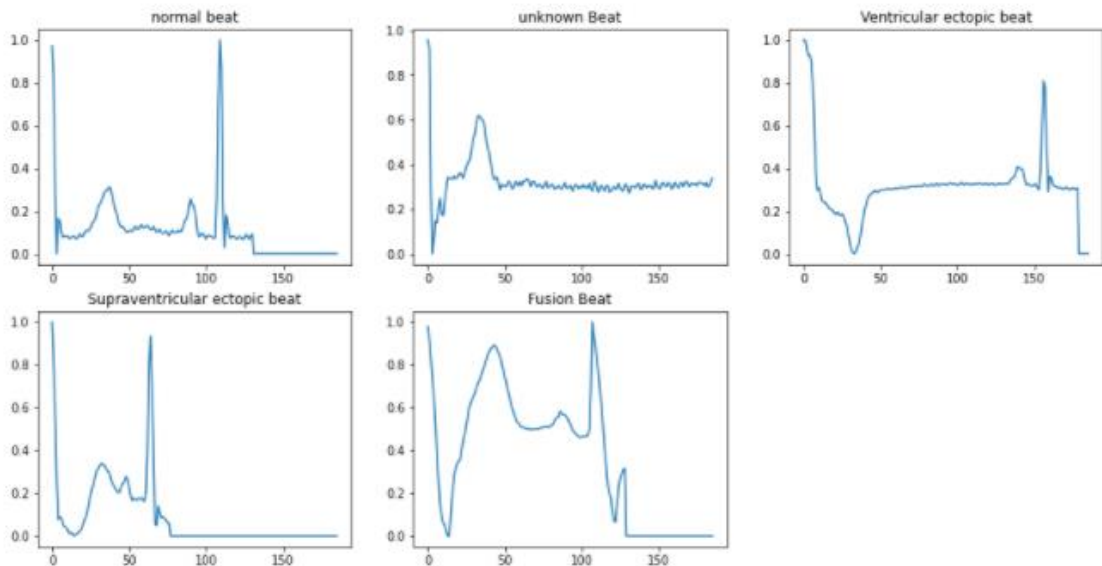


Рисунок 5.1 - Зразки типів ударів серця

Вибірка даних є незбалансованою за класами. Так, значно переважає кількість зразків сигналів нормального серцевого ритму – 72471 приклад, в той час як найменшим за кількістю елементів є клас, що представляє удари-злиття нормальних та шлуночкових ектопічних ударів – 641 приклад.

У таблиці 5.1 наведено кількісний розподіл прикладів між класами серцебиття.

Таблиця 5.1 - Кількісний розподіл прикладів між класами серцебиття

Клас серцебиття	Кількість прикладів
N: удари, що походять із синусового вузла (нормальні удари)	72471
S: надшлуночкові ектопічні удари (SVEB)	2223
V: шлуночкові ектопічні удари (VEB)	5788
F: удари, які є результатом злиття нормальних та VEB	641
Q: будь-яке серцебиття, що не входить до категорій N, S, V або F	6431



На рисунку 5.2 наведено діаграму відсоткового розподілу ударів між описаними вище класами.

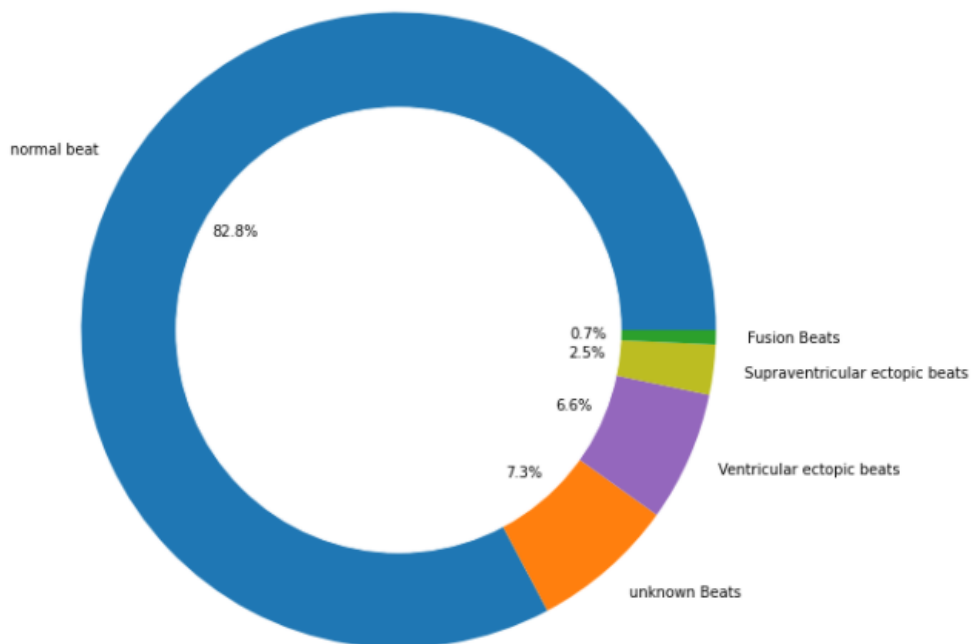


Рисунок 5.2 – Діаграма розподілу ударів між п'ятьма класами

Отже, для розв'язання задачі класифікації набір даних необхідно привести до збалансованого виду, аби уникнути зміщення в сторону класу, представленого більшою кількістю прикладів.

Підхід балансування вибірки синтетичними зразками буде розглянуто в підрозділі 5.3.

## 5.2. Розрахунок фрактальної розмірності

Для сигналів ЕКГ були розраховані фрактальні розмірності. За результатами дослідження встановлено, що більш складні серцеві захворювання мають меншу фрактальну розмірність. Так, на рисунку 5.3 наведено графіки та показники фрактальних розмірностей (а) для нормальних ударів та (б) шлуночкових ектопічних ударів.

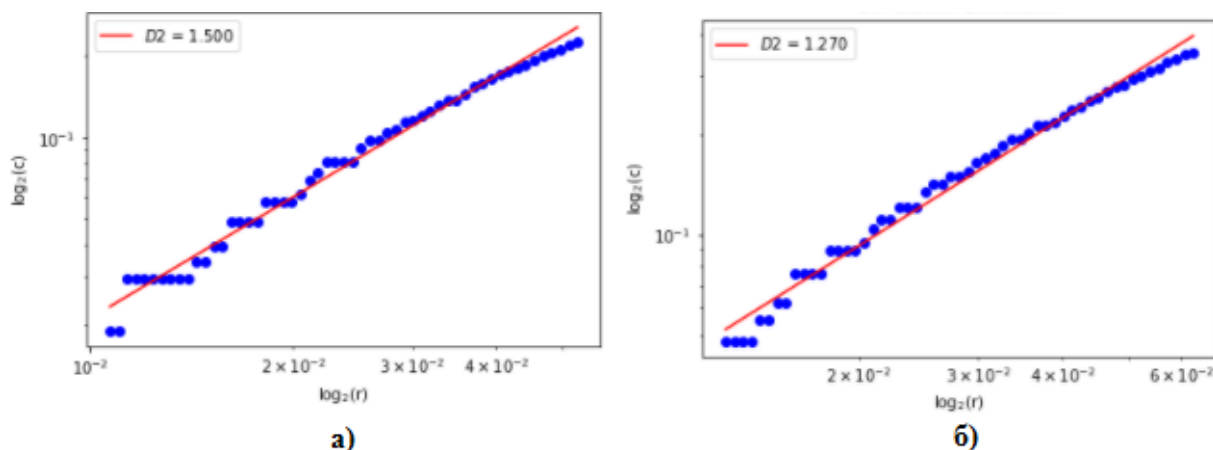


Рисунок 5.3 - Графіки та показники фрактальних розмірностей (а) для нормальних ударів та (б) шлуночкових ектопічних ударів

У таблиці 5.2 наведено середні показники фрактальних розмірностей кожного класу захворювань. Найнижчий показник демонструє клас надшлуночкових ектопічних ударів (SVEB), що складає 1,27. Найвищий показник, як було згадано вище, мають нормальні удари синусового вузла, що складає 1,53.

Таблиця 5.2 - Середні показники фрактальних розмірностей кожного класу

Клас серцебиття	Середній показник фрактальної розмірності
N: удари, що походять із синусового вузла (нормальні удари)	1,53
S: надшлуночкові ектопічні удари (SVEB)	1,27
V: шлуночкові ектопічні удари (VEB)	1,31
F: удари, які є результатом злиття нормальних та VEB	1,42
Q: будь-яке серцебиття, що не входить до категорій N, S, V або F	1,39

Отже, такі показники свідчать про те, що серце людини є складною нелінійною структурою та з появою порушень серцевого ритму самоподібність ударів стає меншою, що свідчить про їх нерегулярність.

### 5.3. Генерація синтетичних зразків

З метою покращення результатів класифікації було проведено дослідження з генерації штучних зразків ЕКГ для збільшення датасету. Для цього запропоновано використання генеративно-змагальної мережі, архітектуру якої наведено на рисунку 5.4.

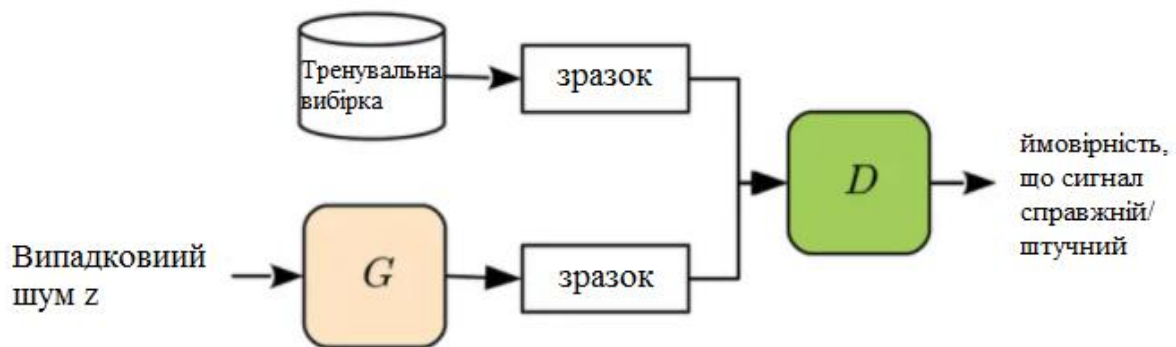


Рисунок 5.4 – Архітектура генеративно-змагальної мережі

Розглянемо окремо архітектуру генератора та дискримінатора мережі.

#### 1. Генератор:

На вхід генератор отримує випадковий шум розмірності  $100 \times 1$ , який передається у двонаправлений шар короткочасної довгострокової пам'яті BiLSTM (рис. 5.5, 5.6). Використання двонаправленого шару дозволяє надати мережі додатковий контекст сигналу. На прихованих згорткових шарах використовується функція активації LeakyReLU. На вихідному шарі використовується функція активації Tanh.

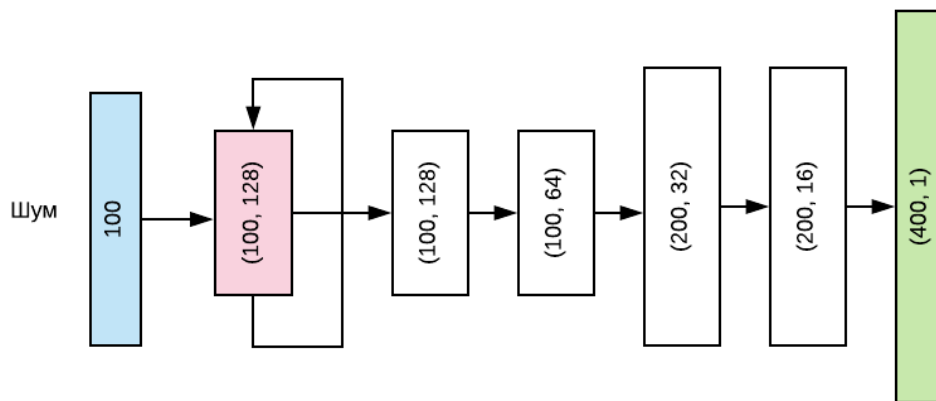


Рисунок 5.5 – Структурна схема архітектури генератора мережі

Layer (type)	Output Shape	Param #
bidirectional_17 (Bidirectio	(None, 100, 128)	39424
conv1d_155 (Conv1D)	(None, 100, 128)	262272
leaky_re_lu_138 (LeakyReLU)	(None, 100, 128)	0
conv1d_156 (Conv1D)	(None, 100, 64)	131136
leaky_re_lu_139 (LeakyReLU)	(None, 100, 64)	0
up_sampling1d_35 (UpSampling	(None, 200, 64)	0
conv1d_157 (Conv1D)	(None, 200, 32)	32800
leaky_re_lu_140 (LeakyReLU)	(None, 200, 32)	0
conv1d_158 (Conv1D)	(None, 200, 16)	8208
leaky_re_lu_141 (LeakyReLU)	(None, 200, 16)	0
up_sampling1d_36 (UpSampling	(None, 400, 16)	0
conv1d_159 (Conv1D)	(None, 400, 1)	257
permute_34 (Permute)	(None, 1, 400)	0

Рисунок 5.6 - Архітектура генератора мережі, реалізована мовою Python

## 2. Дискримінатор:

Дискримінатор отримує на вхід зразок, синтезований генератором, на прихованих згорткових шарах використовується функція активації LeakyReLU. Повнозв'язний вихідний шар має вихід розмірності 1 (рис. 5.7, 5.8).

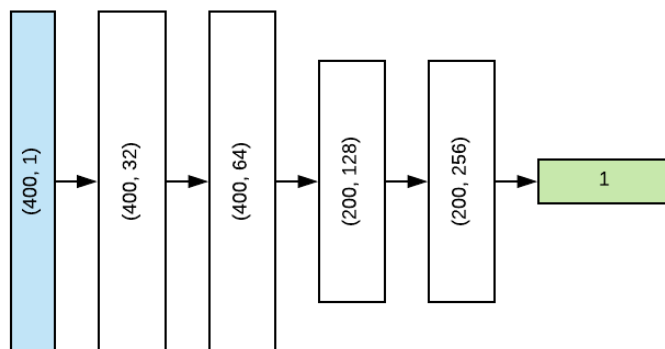


Рисунок 5.7 – Структурна схема архітектури дискримінатора мережі

Layer (type)	Output Shape	Param #
permute_35 (Permute)	(None, 400, 1)	0
conv1d_160 (Conv1D)	(None, 400, 32)	544
leaky_re_lu_142 (LeakyReLU)	(None, 400, 32)	0
conv1d_161 (Conv1D)	(None, 400, 64)	32832
leaky_re_lu_143 (LeakyReLU)	(None, 400, 64)	0
max_pooling1d_35 (MaxPooling)	(None, 200, 64)	0
conv1d_162 (Conv1D)	(None, 200, 128)	131200
leaky_re_lu_144 (LeakyReLU)	(None, 200, 128)	0
conv1d_163 (Conv1D)	(None, 200, 256)	524544
leaky_re_lu_145 (LeakyReLU)	(None, 200, 256)	0
max_pooling1d_36 (MaxPooling)	(None, 100, 256)	0
flatten_17 (Flatten)	(None, 25600)	0
dense_17 (Dense)	(None, 1)	25601

Рисунок 5.8 - Архітектура дискримінатора мережі, реалізована мовою Python

В якості функції оптимізації для обох мереж було обрано Adam з параметром  $\text{learning rate} = 0,0002$ . Кількість епох: 80.

В результаті було одержано синтетичні зразки ЕКГ для різних класів серцебиття. Слід зауважити, що для класу нормальних ударів додаткові зразки не генерувалися, адже він представлений достатньою кількістю прикладів. На рисунку 5.9 наведено приклад згенерованого удару, який є результатом злиття нормального та шлуночкового ектопічного ударів.

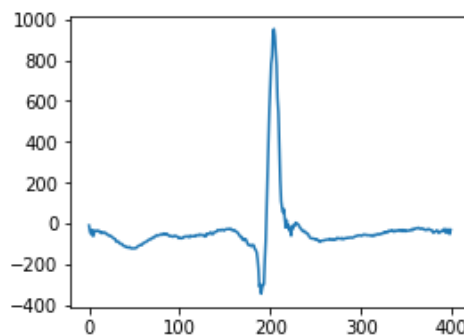


Рисунок 5.9 - Приклад згенерованого удару, який є результатом злиття нормального та шлуночкового ектопічного ударів

На рисунку 5.10 наведено графіки кривих функції втрат дискримінатора і генератора. Так, на графіку функції втрат червона крива відповідає втратам дискримінатора, синя – втратам генератора. Як бачимо, після 60 епохи функції втрат обох мереж перейшли до стану затухання та коливались в діапазоні значень від 0,5 до 2. Вважаємо ці значення такими, що задовольняють розв'язувану задачу.

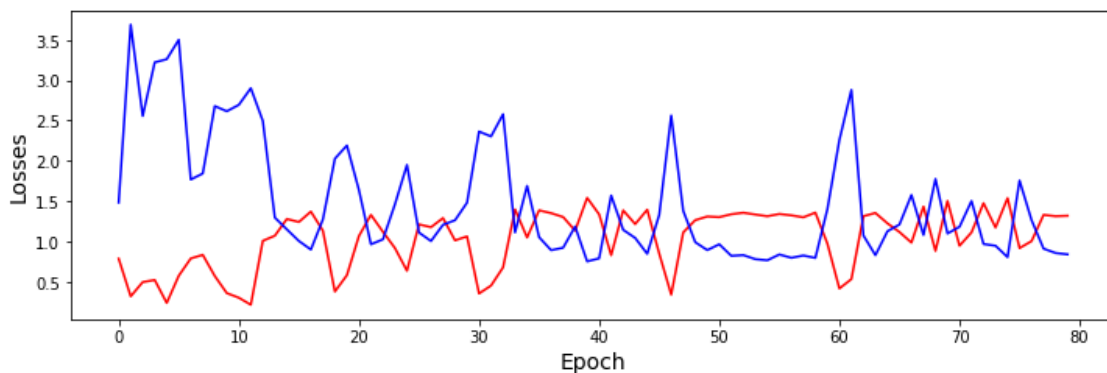


Рисунок 5.10 - Графіки кривих функції втрат дискримінатора і генератора

Для оцінки якості генеративно-змагальної мережі було обрано такі показники, як корінь середньоквадратичної помилки моделі (RMSE) та максимальна середня невідповідність ядра (MMD), а саме середні, мінімальні та максимальні значення (табл. 5.3). Так, середні показники середньоквадратичної помилки та максимальної середньої невідповідності ядра прямують до нуля, що свідчить про те, що згенеровані зразки ЕКГ є подібними до реальних даних.

Таблиця 5.3 – Показники якості генеративно-змагальної мережі

	Mean	Min	Max
RMSE	0,078	0,021	0,169
MMD	$9 \cdot 10^{-4}$	$-6 \cdot 10^{-5}$	$1 \cdot 10^{-2}$

#### 5.4. Оцінка якості результатів класифікації

Для оцінки ефективності доповнення набору реальних даних синтетичними з метою підвищення результатів класифікації ЕКГ порівняємо результати навчання:

1. Мережі, що навчалась на стандартному датасеті реальних даних ЕКГ

- Мережі, що навчалась на змішаному датасеті, що складається з реальних та синтезованих зразків.

Розглянемо архітектуру побудованого класифікатора (рис 5.11). Мережа складається з чотирьох шарів LSTM, шарів дропауту з ранжуванням відповідного параметру в діапазоні 0,2 – 0,3 та повнозв'язного шару. Вихідний шар має вихід розмірності 5, що відповідає кількості класів серцевих ударів. В якості функції активації обрано softmax.

Як функцію втрат було обрано binary crossentropy, оптимізатор – Adam. Навчання мережі проводилось протягом 20 епох.

Layer (type)	Output Shape	Param #
lstm_24 (LSTM)	(None, 400, 100)	40800
dropout_29 (Dropout)	(None, 400, 100)	0
lstm_25 (LSTM)	(None, 400, 200)	240800
dropout_30 (Dropout)	(None, 400, 200)	0
lstm_26 (LSTM)	(None, 400, 200)	320800
dropout_31 (Dropout)	(None, 400, 200)	0
lstm_27 (LSTM)	(None, 200)	320800
dropout_32 (Dropout)	(None, 200)	0
dense_15 (Dense)	(None, 200)	40200
dense_16 (Dense)	(None, 5)	1005

Рисунок 5.11 - Архітектура мережі класифікатора, реалізована мовою Python

Отримано такі результати навчання мережі:

- Стандартний набір даних

Для оцінки якості класифікації ударів серцебиття моделлю, що навчалась на стандартному наборі даних, проаналізуємо показники точності, чутливості та показник метрики F-1 score (таблиця 5.4).

Як бачимо, найбільш точною модель виявилась для класу нормального серцебиття, який представлений найбільшою кількістю прикладів. Показники



чутливості та F-1 score також є наймищими для цього класу. Нескладно помітити закономірне зростання показників між класами відповідно до кількості прикладів у них. Так, найгірші результати показав клас ударів, які є результатом злиття нормальних та VEB, які підтримуються 162 прикладами на тестовій вибірці. Точність класифікатора склала 0,97.

Таблиця 5.4 – Метрики оцінки якості мережі (стандартний датасет)

Клас	Precision	Recall	F-1 score	Support
0	0,97	0,99	0,98	18118
1	0,83	0,55	0,67	556
2	0,93	0,85	0,89	1448
3	0,61	0,58	0,59	162
4	0,95	0,96	0,95	1608

На рисунку 5.12 наведено графіки кривих функцій точності та втрат класифікатора, який навчався на стандартному наборі даних. Як бачимо, до 14 епохи коливання між кривими валідаційної та тренувальної вибірки були незначними, криві точності зростали, відповідно криві функції втрат спадали. Проте на 14 епосі відбулось різке падіння точності та зростання функції втрат, що свідчить про зниження точності класифікатора. Затухаючі коливання спостерігались також протягом наступних епох.

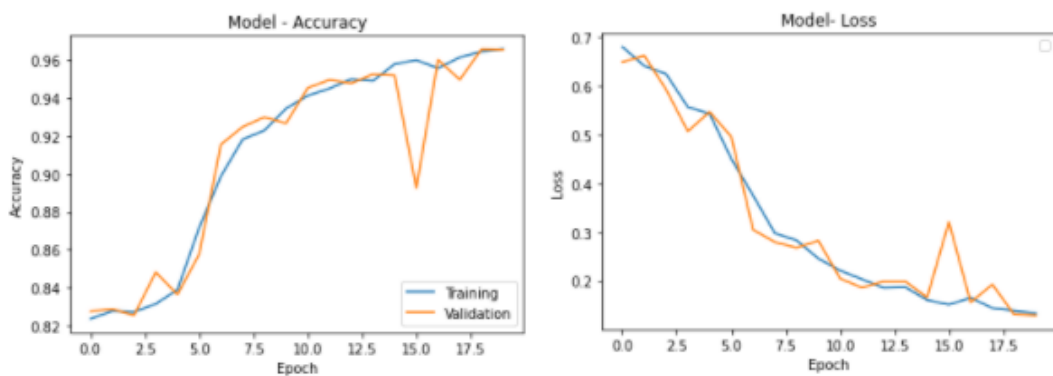


Рисунок 5.12 - Графіки кривих функцій точності та втрат класифікатора, який навчався на стандартному наборі даних

Отже, можна зробити висновок, що модель загалом показала високу точність, проте точність класифікації за класами значно відрізняється в залежності від кількості представлених у них прикладів.

## 2. Змішаний набір даних

Для оцінки якості класифікації ударів серцебиття моделлю, що навчалась на змішаному наборі даних, проаналізуємо показники точності, чутливості та показник метрики F-1 score (таблиця 5.5).

Доповнені згенерованими зразками були усі класи серцебиття окрім класу нормального синусового ритму, оскільки він представлений достатньою кількістю прикладів.

Таблиця 5.5 – Метрики оцінки якості мережі (змішаний датасет)

Клас	Precision	Recall	F-1 score	Support
0	0,97	0,99	0,99	18118
1	0,86	0,74	0,79	656 (556 + 100)
2	0,94	0,94	0,94	1498 (1448 + 50)
3	0,64	0,61	0,63	312 (162 + 150)
4	0,95	0,93	0,94	1658 (1608 + 50)

Як бачимо, аналізовані показники покращились для класів 1, 2 та 3, що були розширені синтетичними даними. Точність класифікатора склала 0,98.

На рисунку 5.13 наведено графіки кривих функцій точності та втрат класифікатора, який навчався на змішаному наборі даних. Як бачимо, до 4 епохи спостерігаються значні коливання функцій точності та втрат, що свідчить про нестабільну поведінку класифікатора. Проте після 4 епохи відбувається постійне зростання точності та спадання функції втрат. Коливання між кривими валідаційної та тренувальної вибірки є незначними, що також свідчить про високу якість класифікації мережі.

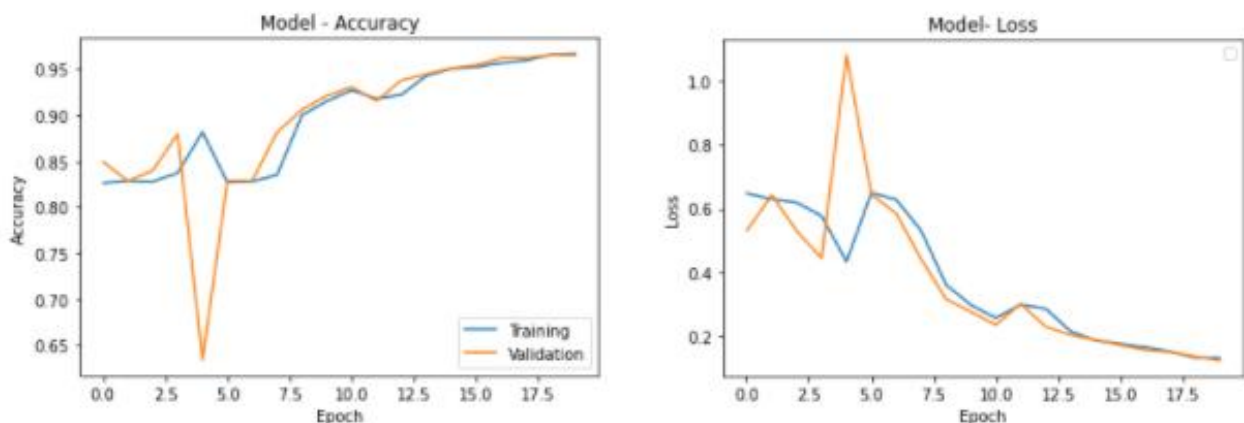


Рисунок 5.13 - Графіки кривих функцій точності та втрат класифікатора, який навчався на змішаному наборі даних

Для того, щоб оцінити ефективність доповнення набору даних синтетичними зразками, обчислимо відсоткову зміну в метриці точності  $\% \Delta_p$  для кожного з класів, що доповнювались синтетичними зразками (таблиця 5.6).

Таблиця 5.6 - Відсоткова зміна в метриці точності  $\% \Delta_A$  для доповнених класів серцебиття

Клас серцебиття	Відсоткова зміна в метриці точності $\% \Delta_A$
1	3,61
2	1,07
3	4,91
4	0

Відсоткова зміна точності є додатньою для всіх класів 1, 2, 3 та нульовою для класу 4, отже, загалом можна зробити висновок про ефективність застосованого підходу до розширення вибірки даних.

## 5.5. Висновки до розділу

У п'ятому розділі було описано та проведено аналіз набору даних ЕКГ, що використовуються в роботі: детально описано класи серцевих ударів, що містяться в датасеті, визначено кількісний та відсотковий розподіл прикладів між класами. Також проведено процедуру балансування даних, в результаті якої вдалося досягти однакового розподілу кількості прикладів між класами.

Проведено розрахунок та аналіз фрактальних розмірностей сигналів ЕКГ за типом серцебиття. Так, було встановлено, що сигнали, які відповідають більш складним типам ударів, тобто певним захворюванням, мають меншу фрактальну розмірність.

В розділі описано архітектуру та основні параметри генеративно-змагальної мережі, що використовується для генерації синтетичних зразків ЕКГ, виконано аналіз одержаних результатів.

Було описано архітектуру класифікатора та результати експериментів з різними типами наборів даних кардіограм.

## РОЗДІЛ 6. РОЗРОБКА СТАРТАП-ПРОЕКТУ

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого. Головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Далі наведено маркетинговий аналіз стартап проекту. В межах цього етапу:

- а) розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- б) аналізуються ринкові можливості щодо його реалізації;
- в) на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

## 6.1. Опис ідеї проекту

В межах підпункту було проаналізовано і подано у вигляді таблиць:

- а) зміст ідеї (що пропонується);
- б) можливі напрямки застосування;
- в) основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- г) чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці (таблиця 6.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 6.1 - Опис ідеї стартап-проекту

<b>Зміст ідеї</b>	<b>Напрямки застосування</b>	<b>Вигоди для користувача</b>
Дана комплексна система дозволяє розв'язати проблему відслідковування хвороб серця у населення. На основі результатів класифікації відбувається розв'язання цієї проблеми – вчасне визначення хвороби на початковій стадії	1. Визначення стану здоров'я людини та наявності серцевих захворювань	Визначення захворювання на початковій стадії, вчасне лікування та нелінійності, що приводить до необхідності
	2. Аналіз стану здоров'я людини в динаміці	Можливість відстежування стану пацієнта

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

- а) визначення переліку техніко-економічних властивостей та характеристик ідеї;
- б) визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;
- в) проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (табл. 6.2).

Таблиця 6.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	FORecast 4u			
1.	Точність прогнозування	Застосування кращої моделі	Відсутнє прогнозування			+
2.	Ризики невірною прогнозу	Існують, через велику кількість факторів	Відсутні через відсутність прогнозу		+	
3.	Доступність, зручність	Обмежені функції: побудова моделі і прогнозування	Власний інтерфейс		+	

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

## 6.2 Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 6.3):

- 1) за якою технологією буде виготовлено товар згідно ідеї проекту?
- 2) чи існують такі технології, чи їх потрібно розробити/додати?
- 3) чи доступні такі технології авторам проекту?

Таблиця 6.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Створення системи класифікації кардіограм	Використання мови програмування Python	Наявна	Доступна
		Tensorflow	Наявна	Доступна
		Використання мови програмування R	Наявна	Доступна
		Використання мови програмування Java	Наявна	Наявна
Обрана технологія реалізації ідеї проекту: мова програмування Python. Бібліотека Tensorflow				

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було



обрано такі технології, як Python 3.4 та Tensorflow через їх доступність та безкоштовність.

### 6.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час 77 ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 6.4).

Таблиця 6.4 - Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	15000000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Ліцензії, регуляторні
5	Специфічні вимоги до стандартизації та сертифікації	+
6	Середня норма рентабельності в галузі (або по ринку), %	350

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект.

За результатами аналізу таблиці 6.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та сформовано орієнтовний перелік вимог до товару для кожної групи (табл. 6.5).

Таблиця 6.5 - Характеристика потенційних клієнтів стартап-проекту

<b>№ п/п</b>	<b>Потреба, що формує ринок</b>	<b>Цільова аудиторія (цільові сегменти ринку)</b>	<b>Відмінності у поведінці різних потенційних цільових груп клієнтів</b>	<b>Вимоги споживачів до товару</b>
1.	Потреба в відокремленні сигналу від шуму побудови прогнозів.	Аналітики, аналітичні відділи, лікарі	Велика кількість даних.	Простота використання, висока точність
2.	Створення якісного прогнозу та визначення наявності захворювання	Користувачі додатку	Цікавить простота у використанні, низька ціна підтримки системи	Швидкість обробки, низька ціна
3.	Створення точного довгострокового прогнозу, відстеження стану здоров'я користувача в часі	Великі медичні компанії, користувачі додатку	Цікавить передусім точність довгострокових прогнозів	Висока якість прогнозу

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 6.6, 6.7).

Таблиця 6.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Вихід на ринок продуктів з кращими характеристиками	<p>Передбачити додаткові переваги власного програмного продукту (ПП) для того, щоб повідомити про них саме після виходу на ринок конкурентів.</p> <p>Вдосконалення технічних моментів власного продукту.</p> <p>Обрати нову цільову аудиторію і зосередитися на ній: зниження цін.</p>
2	Зміна потреб користувачів	Користувачам необхідне програмне забезпечення з іншим функціоналом	Передбачити можливість додавання нового функціоналу до створюваного ПП

Таблиця 6.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Конкуренція	Відсутність аналогічного продукту для вітчизняного користувача.	Адаптація програмного продукту до вітчизняних особливостей.
2	Поява нових методів прогнозування	З'являться нові методи, що будуть швидше та ефективніше прогнозувати показники	Покращити ПП додаванням нового функціоналу, розширення можливостей

## Продовження таблиці 6.7

3	Поява нових методів моделювання	З'являться нові методи, що будуть швидше, та більш точно моделювати процеси	Покращити ПП додаванням нового функціоналу, розширення можливостей
---	---------------------------------	---	--

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 6.8).

Таблиця 6.8 - Ступеневий аналіз конкуренції на ринку

<b>Особливості конкурентного середовища</b>	<b>В чому проявляється дана характеристика</b>	<b>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</b>
1. Вказати тип конкуренції - монополія	На ринку присутні декілька компаній-конкурентів, але їх товар дещо відрізняється між собою.	Підтримка якості продукту та постійні нововведення, вдосконалення.
2. За рівнем конкурентної боротьби - міжнародний	Компанії-конкуренти з інших країн	Створити основу ПП так, щоб можна було легко переробити даний ПП для використання у галузях інших країн.
3. За галузевою ознакою - міжгалузева	Продукт може використовуватись для різних галузей	Постійне вдосконалення продукту, що не має прив'язки до сфери
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між видами ПП, їх особливостями.	Створити ПП, враховуючи недоліки конкурентів

## Продовження таблиці 6.8

5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення ПП, щоб собівартість була нижчою	Удосконалення моделі. Використання більш дешевих технологій для розробки, ніж використовують конкуренти, але тільки якщо ці технології відповідають необхідним вимогам якості.
6. За інтенсивністю - не марочна	Бренд присутній, але його роль незначна	Реклама, участь у конференціях, семінарах.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл. 6.9).

Таблиця 6.9 - Аналіз конкуренції в галузі за М. Портером

<b>Складові аналізу</b>	<b>Прямі конкуренти в галузі</b>	<b>Потенційні конкуренти</b>	<b>Постачальники</b>	<b>Клієнти</b>	<b>Товари-замінники</b>
	<i>Навести перелік прямих конкурентів</i>	<i>Визначити бар'єри входження в ринок</i>	<i>Визначити фактори сили постачальників</i>	<i>Визначити фактори сили споживачів</i>	<i>Фактори загроз з боку замінників</i>
	SAS Matlab	Наявність вже існуючих рішень	-	Контроль якості продукту	Наявність більш широкого функціоналу, зручнішого інтерфейсу та авторитет (перевірена якість)

## Продовження таблиці 6.9

<b>Висновки:</b>	Досить інтенсивна конкурентна боротьба з вже закріпившимися на ринку гравцями	Є можливість виходу на ринок, але є і конкуренти. Строки – 18 місяців.	-	Клієнти диктують умови роботи на ринку: зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для побудови моделей і прогнозів.	Необхідно випускати ПЗ не гірше, ніж у конкурентів та розширяти функціонал.
------------------	---	--	---	--	---

За результатами аналізу табл. 6.9 було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію. Також було зроблено висновок щодо характеристик, які повинен мати проект, щоб бути конкурентоспроможним на ринку.

Цей висновок був врахований при формулюванні переліку факторів конкурентоспроможності у наступному пункті. На основі аналізу конкуренції, проведеного в табл. 6.9, а також із урахуванням характеристик ідеї проекту (табл. 6.2), вимог споживачів до товару (табл. 6.5) та факторів маркетингового середовища (таблиці 6.6, 6.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлено у табл. 6.10

Таблиця 6.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Багатофункціональність	Жоден конкурент не є настільки багатунціональним, не здатен на прогноз, чистку шумів та класифікацію сигналів

## Продовження таблиці 6.10

2	Якість	Можливість використання системи без складних імплементация в існуючу екосистему автомобіля
3	Висока якість прогнозу, велика кількість допоміжних статистичних даних	Робота з клієнтами – великими компаніями та окремими спеціалістами

За визначеними факторами конкурентоспроможності (табл. 6.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 6.11).

Таблиця 6.11 - Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з ... (назва підприємства)						
			-3	-2	-1	0	+1	+2	+3
1	Якість аналізу	15					*		
2	Простота використання	20			*				
3	Орієнтованість на кінцевого споживача	7					*		

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 6.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 6.11). Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого

можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 6.12 - SWOT-аналіз стартап-проекту

<p><b>Сильні сторони:</b> Точність аналізу Простота використання Автономність</p>	<p><b>Слабкі сторони:</b> Потрібен час для навчання системи Інтерфейс користувача</p>
<p><b>Можливості:</b> Аналіз емоцій людини на основі біосигналів Застосування системи для попередження серцевих нападів</p>	<p><b>Загрози:</b> Конкуренція</p>

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. таблицю 6.9, аналіз потенційних конкурентів). Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 6.13).

Таблиця 6.13 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Ліцензування алгоритму і патентів	85%	6 місяців
2	Створення ПП з подальшим з подальшою інтеграцією в систему автомобіля	75%	18 місяців

Після аналізу було обрано альтернативу №1.



## 6.4 Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 6.14).

Таблиця 6.14 - Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Компанії (українські та міжнародні) діяльність яких пов'язана з фінансово-економічною сферами.	Висока	Високий	Сильна	Складно
2	Приватні підприємства міського та міжнародного рівня, діяльність яких пов'язана з фінансово-економічною сферами.	Висока	Високий	Сильна	Складно
3	Приватні підприємства, обласного рівня.	Помірна	Помірний	Помірна	Середня складність
4	Підприємства регіонального характеру	Помірна	Слабкий	Слабка	Просто
5	ФОП, які діють у фінансово-економічній сферах.	Слабка	Слабкий	Слабка	Просто

Які цільові групи обрано:1,2,3.

За результатами аналізу потенційних груп споживачів було обрано цільові групи, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 6.15).

Таблиця 6.15 - Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1		Визначити потреби кожної з груп, розробити відповідно до них стратегії приваблення клієнтів та маркетингової комунікації	Цінова політика, універсальність продукту, орієнтованість на кінцевого споживача	Стратегія диференціації

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 6.16).

Таблиця 6.16 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	«Прешопроходець»	Шукати нових	Ні	Стратегія заняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. таблицю 6.5), а також в залежності від обраної базової стратегії розвитку (таблиця 6.15) та стратегії конкурентної поведінки (таблиця 6.16) розроблено стратегію позиціонування (таблиця 6.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 6.17 - Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Легкість розуміння, зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для побудови моделей і прогнозів.	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Економія часу; Зручність застосування; Практичність та точність результату

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії на ринку.

## 6.5 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього у таблиці 6.18 підсумовано результати попереднього аналізу конкурентоспроможності товару. Концепція товару - письмовий опис фізичних

та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 6.18 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Швидкість отримання результату	Швидка побудова моделі та створення прогнозу	Відсутність необхідності звертатися до сторонньої особи/компанії для побудови моделі та прогнозу. Дані компанії-користувача, якими оперує ПП, не передаються третім особам, чого вимагає політика безпеки багатьох компаній.
2	Зручність застосування	Не потрібно мати глибоких знань, для того щоб побудувати модель та спрогнозувати показники	ПП сам обирає необхідний та оптимальний метод для побудови моделі та прогнозу. Не потрібно мати глибоких знань у прогнозуванні для того, щоб користуватися ПП
3	Практичність та точність результату	Користувач отримує точні (з малою похибкою розбіжності) результати.	Користувач на виході роботи ПП отримує модель та прогноз, котрі відповідають необхідним показникам достовірності та точності.

Розроблено тривірневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 6.19).

1-й рівень При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Таблиця 6.19 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
<i>I. Товар за задумом</i>	Зручність та швидкість отримання практичного результату щодо побудови моделі та прогнозування процесів.		
<i>II. Товар у реальному виконанні</i>	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. функція побудови моделі процесу 2. функція побудови прогнозу		
	Якість: достовірність побудови математичної моделі, достовірність побудови прогнозу		
	Пакування: відсутнє		
	Марка: StatLabs «Forec»		
<i>III. Товар із підкріпленням</i>	До продажу: відсутнє		
	Після продажу: персональна підтримка в обслуговуванні за додаткову платню.		
Вихідний код та математична модель будуть закриті. На ідею зареєстровано патент.			

Після формування маркетингової моделі товару слід відмітити, що проект буде захищено від копіювання за допомогою ноу-хау. Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або

товари субститути, а також аналіз рівня доходів цільової групи споживачів (таблиця 6.20). Аналіз проведено експертним методом.

Таблиця 6.20 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	1800\$	3500\$	У всіх трьох груп високий рівень доходів	Базова покупка 1000\$ Подальша персональна підтримка в обслуговуванні 150\$/місяць

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 6.21)

Таблиця 6.21 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які допоможуть отримати вигоду та покращити дохідність. Вони цікавляться сучасними розробками та інноваційними рішеннями, тому відвідують конференції, інтернет-конференції, семінари.	Встановлення контактів із споживачами і підтримання їх. Формування попиту і стимулювання збуту. Дослідницька робота зі збору маркетингової інформації. Доробка товару, виходячи з потреб конкретного покупця.	Один (від виробника одразу споживачу)	Прямий канал збуту до споживача, мінімізувати збутові витрати розвитку маркетингового спілкування із споживачем

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 6.22).

Таблиця 6.22 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які допоможуть	Конференції, інтернет-конференції, семінари, огляд професійно	Позиція на основі порівняння фірми з товарами	Створення репутації фірми — виробнику чи посереднику; збільшення чистого	Шукаєте вірний шлях для розвитку вашої компанії? Досить
	отримати вигоду та покращити дохідність. Вони цікавляться сучасними розробками та інноваційними рішеннями, тому відвідують конференції, інтернет-конференції, семінари.	літератури, інтернет, періодичні видання у різноманітних (профільних) галузях.	конкурентів; Відмінності особливо споживача	прибутку та рентабельності фірми; – збільшення потоків покупців та обсягів продажу; – стабілізація обсягів продажу в період зменшення попиту та загального спаду ділової активності	даремно гаяти час на вгадування вірної стратегії! Користуйтеся «Fores» і світле майбутнє вашої компанії забезпечено!

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

#### 6.6. Висновки до розділу

В даному розділі було проведено аналіз програмного продукту у якості стартап-проекту. Можна зазначити, що у проекту є можливість комерціалізації, оскільки ринок потребує якісний продукт, що надає оцінки стану серцево-судинної системи.

На ринку наявна монополістична конкуренція, існує декілька фірм-конкурентів, але їх товар дещо відрізняється, тому вихід на ринок не буде легким і потребує грамотної стратегії виходу. Для впровадження ринкової реалізації проекту слід обрати альтернативу, яка передбачає розробку програмного продукту з подальшим розповсюдженням ліцензій та права на використання за певну роялті.



## ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ ЩОДО ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Для задач машинного навчання, в тому числі й задач розпізнавання, важливим аспектом є наявність великого набору даних, на якому проводиться навчання та валідація моделей. При розв'язанні задачі класифікації медичних даних часто постає проблема недостатньо об'ємної вибірки даних, що пов'язано з приватністю медичної інформації, а також потребою у попередній обробці медичними працівниками для розмітки навчальних даних. Як засіб подолання такої проблеми було запропоновано використання генеративно-змагальної мережі для синтезування штучних зразків ЕКГ.

В якості класифікатора було побудовано LSTM мережу та проаналізовано точність її роботи на двох типах виборок: стандартній, а також доповненій синтетичними зразками. Висновок є наступним: розширення вибірки штучними прикладами здатне підвищити точність класифікації.

Також було виконано розрахунки фрактальних розмірностей сигналів ЕКГ для аналізу складності сигналів та дослідження їх нелінійності.

Перспективи розвитку дослідження:

1. Комбінування методів нелінійної динаміки та аналізу часових сигналів ЕКГ для виділення ознак класифікації.
2. Збільшення класів класифікації.
3. Дослідження методів валідації вибірки, сформованої з синтезованих зразків сигналів ЕКГ (зокрема може бути використано розрахунок фрактальних розмірностей).
4. Вдосконалення архітектури класифікатора.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Lee W.C., Hung F.H., Tsang K.F. et al. A Speedy Cardiovascular Diseases Classifier Using Multiple Criteria Decision Analysis. *Sensors*. 2015. Vol. 15. P. 1312-1320.
2. Shin H. S., Lee C., Lee M. Adaptive threshold method for the peak-detection of photoplethysmographic waveform. *Comput. in Biology and Medicine*. 2009. Vol. 39. P. 1145-1152.
3. Butler G. C., Yamamoto Y., Hughson R. L. Heart Rate Variability to Monitor Autonomic Nervous System Activity During Orthostatic Stress. *J. Clin. Pharmacol.* Jun. 1994. Vol. 34, No. 6. P. 558–562.
4. Evans S., Seidman L. C., Tsao J. C. Heart rate variability as a biomarker for autonomic nervous system response differences between children with chronic pain and healthy control children. *J. Pain Res.* Jan. 2013. Vol. 6. P. 449–57.
5. Sotaniemi A., Myllylii V. Comparison of heart rate variability analysis methods in patients with Parkinson's disease. *C. Neurophysiology, B. Engineering, and I. S. R. Hospital.* April, 2002. Vol. 7. P. 408– 414.
6. Gohara T., Mizuta H., Takeuchi I. Heart rate variability change induced by the mental stress: the effect of accumulated fatigue. *Proceedings of the 1996 Fifteenth Southern Biomedical Engineering Conference.* Chicago, USA, Feb 11-13. 1996. P. 367–369.
7. Wyller V. B., Barbieri R., Thaulow E. Enhanced vagal withdrawal during mild orthostatic stress in adolescents with chronic fatigue. *Ann. Noninvasive Electrocardiol.* 2008. Vol. 5. P. 342-354.
8. Bilchick K., Berger R. Heart rate variability. *J. Cardiovasc. Electrophysiol.* 2006. Vol. 17, No 6. P. 691-694.
9. Dishman R., Nakamura Y., Garcia M. Heart rate variability, trait anxiety, and perceived stress among physically fit men and women. *Int. J. Psychophysiol.* 2000. Vol. 37, No 2. P. 121-133.

10. Acharya U., Joseph K., Kannathal N. Heart rate variability: a review. *Med. Biol. Eng. Comput.* 2006. Vol. 44, No. 12. P.1031-1051.
11. Spacic S, Kalauzi A, Grbic G, Martac L, Culic M. Fractal analysis of rat brain activity after injury. *Med & Biol Eng & Comput.* 2005. Vol. 43. P. 345- 348
12. Mulder L.J.M., de Waard D., Brookhuis K.A. Estimating mental effort using heart rate and heart rate variability. *Handbook of Human Factors and Ergonomics Methods.* 2004. Vol. 20, No. 1. P. 20-27.
13. Manzey D. Psychophysiologie mentaler Beanspruchung. In: F. Rösler (Hg): Ergebnisse und Anwendungen der Psychophysiologie. Enzyklopädie der Psychologie Hogrefe, Göttingen 1997. P. 799-864. URL: <https://elib.dlr.de/27581/> (дата звернення: 17.12.2020)
14. Abisse S., Lampert R., Burg M. J. Cardiac repolarization instability during psychological stress in patients with ventricular arrhythmias. *Electrocardiol.* 2011. Vol. 44, No 6. P. 678-683.
15. Reddy G.U., Muralidhar M., Varadarajan S. ECG denoising using improved thresholding base donwavelet transforms. *Int. J. Comput. Sci. Netw. Secur.* 2009. Vol. 9. P. 221–225.
16. Xiong P. ECG signal enhancement based on improved denoising auto-encoder. *Engineering Applications of Artificial Intelligence.* 2016. Vol. 52. P. 194-202.
17. Tompkins W. J. Biomedical Digital Signal Processing C-Language Examples and Laboratory Experiments for the IBM®PC. *New Jersey Prentice Hall.* 2000. P. 245–264. URL: [https://www.academia.edu/42979171/BIOMEDICAL\\_DIGITAL\\_SIGNAL\\_PROCESSING\\_C\\_Language\\_Examples\\_and\\_Laboratory\\_Experiments\\_for\\_the\\_IBM\\_PC](https://www.academia.edu/42979171/BIOMEDICAL_DIGITAL_SIGNAL_PROCESSING_C_Language_Examples_and_Laboratory_Experiments_for_the_IBM_PC) (дата звернення: 17.12.2020)
18. Goldberger A. L., Amaral A. N., Glass L. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research for complex physiologic signals. *Circulation.* 2000 Vol. 101. P. 215-220.

19. Chui K. T., Tsang K. F., Wu C. K. Cardiovascular diseases identification using electrocardiogram health identifier based on multiple criteria decision making. *Expert Systems with Applications*. 2015. Vol. 42. P. 5684-5695.
20. Christinaki E., Giannakakis G., Chiarugi O. Comparison of Blind Source Separation Algorithms for Optical Heart Rate Monitoring. *Wireless Mobile Communication and Healthcare*. Milan, Italy. 2014, P. 339-342. URL: <https://www.semanticscholar.org/paper/Comparison-of-blind-source-separation-algorithms-Christinaki-Giannakakis/e2f96e1b1234a33dd1c63e98ce0b200eae75b0ff> (дата звращения:17.12.2020)
21. Takano C., Ohta Y. Heart rate measurement based on a time-lapse image. *Medical Engineering & Physics*. 2007. Vol.29, No. 8. P. 853-857.
22. Raducanu B., Kumar S. ECG analysis using wavelet transformation. *Computational Statistics & Data Analysis*. 2011. Vol.2, No. 1. P. 57-83.
23. Ghayoumi M. A quick review of deep learning in facial expression. *Journal of Communication and Computer*, 2017. Vol. 1. P. 34–38.
24. Kahou, S.E., Michalski V., Konda K. Recurrent neural networks for ECG classification. *The ACM on International Conference on Multimodal Interaction*, Seattle, WA, USA, 9–13 November 2015, P. 467–474.
25. Zhan C., Li W., Ogunbona P., Safaei F. A real-time cardiosignals recognition system for health care. *International Journal of Computer Technology*, 2008. Vol. 2, No.1. P. 10-20
26. Mehrabian, A. Communication without words. *Psychol. Today*. 1968, Vol. 2. P. 53–56.
27. Dornaika F., Raducanu B. Efficient ECG clustering for human robot interaction. *The 9th International Work-Conference on Artificial Neural Networks 79 on Computational and Ambient Intelligence*, San Sebas Xtián, Spain, 20–22 June 2007, P. 700–708.

28. Bartneck C., Lyons M.J. HCI and the heart: Towards an art of the soluble. *Proceedings of the International Conference on Human-Computer Interaction: Interaction Design and Usability*, Beijing, China, 22–27 July 2007, P. 20–29.
29. Kumar S. ECG recognition: A review. *The National Conference on Cloud Computing and Big Data*, Shanghai, China, 4–6 November 2015, P 159–162.
30. Walecki R., Rudovic O. Deep structured learning for signal processing. *Image Vis. Comput.* 2017. Vol.1. P. 143–154
31. Krizhevsky N., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM.* 2017. Vol. 60. P. 83–90.
32. Guennec A. L., Malinowski S., Tavenard R. Data augmentation for time series classification using convolutional neural networks. *Iarchives-ouvertes.* 2016. Vol. 56. P. 12.
33. Choi E., Bahadori M.T., Sun J., Kulas J., Schuetz A., Stewart W.F. RETAIN: an interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in Neural Information Processing Systems.* 2017. Vol. 29. P. 1995–2004.
34. Arjovsky M., Chintala S., Bottou L. Wasserstein generative adversarial networks. *Journal of Machine Learning.* 2017. Vol. 11. P. 214–223.
35. Borji A. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding.* 2019. Vol. 179. P. 41–65.
36. Gretton A., Borgwardt K. M., Rasch M. J., Scholkopf B., Smola A. A kernel two-sample test. *Journal of Machine Learning Research.* 2012. Vol. 13. P. 723–773.
37. Tran T., Pham T., Carneiro G., Palmer L., Reid I. A bayesian data augmentation approach for learning deep models. *Advances in neural information processing systems.* 2017. Vol. 30. P. 2797–2806.
38. Thwaites G. E., Wills B., Thwaites C. L., Le V. T., van Doorn H. R. Clinical and aetiological study of hand, foot and mouth disease in southern vietnam, 2013–2015: Inpatients and outpatients. *International Journal of Infectious Diseases.* 2019. Vol. 80. P. 1 – 9.

39. Goldberger A. L., Amaral L. A., Glass L., Hausdorff J. M., Ivanov P. C., Mark R. G., Mietus J. E., Moody G. B., Peng C.-K., Stanley H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*. 2000 Vol. 101, No. 23 P. 215–220.
40. Tsang K. Heart rate variability: Standards of measurement, physiological interpretation and clinical use. *T. F. of the European Society of Cardiology, the North American Society of Pacing, and Electrophysiology*. 1996. Vol. 93. P. 1043–1065.
41. Lin Z., Khetan A., Fanti G., Oh S. Pacgan: The power of two samples in generative adversarial networks. *Advances in Neural Information Processing Systems*. 2018. Vol. 31. P. 1498–1507.
42. Кравченко В. А. Система підтримки прийняття рішень розпізнавання кардіограм методами штучного інтелекту : дипломна робота ... бакалавра : 6.050101 Комп'ютерні науки / НТУУ «КПІ ім. Ігоря Сікорського». Київ, 2019. 92 с.

## ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```

import tensorflow as tf
from tensorflow.keras import layers, regulariz
ers
from torch_two_sample.statistics_diff import M
MDStatistic
import torch

import matplotlib.pyplot as plt
import numpy as np
import random

from IPython import display
from tqdm import tqdm
from shutil import copyfile

print(tf.__version__)
# from google.colab import drive
# drive.mount('/drive')

norm_value = 2173

#         Helper functions
def test_show(generator, discriminator):
    noise = tf.random.normal([1, 125, 50])
    generated_ecg = generator(noise, training=
False)
    print(generated_ecg.shape)
    plt.plot(generated_ecg[0, 0, :])
    plt.show()

    decision = discriminator(generated_ecg, tr
aining=False)
    print(decision)

def generate_and_save_ecg(model, epoch, test_i
nput, save):
    predictions = model(test_input, training=F
alse)

    fig = plt.figure(figsize=(4,3))
    plt.plot(predictions[0, 0, :] * norm_value
)
# plt.plot(predictions[0, 0, :])

    if save:
        plt.savefig('./ecg_at_epoch_{:04d}.png
'.format(epoch))

    plt.show()

def prepare_data(dim):
#     copyfile(f"/drive/My Drive/Colab Noteboo
ks/data/fix_signals_400.npy", "./fix_signals.n
py")

    data = np.load('./data/fix_signals_400.npy
')
    data = np.reshape(data, (data.shape[0], 1,
data.shape[1]))
    print('Data shape:', data.shape)

    data = data / norm_value # Normalize
    data = np.array(data, dtype='float32')

    plt.figure(figsize=(4,3))
    plt.plot(data[random.randint(0, data.shape
[0])][0] * norm_value)
    plt.show()

    train_size = int(data.shape[0] * 0.9)
    test_size = data.shape[0] - train_size
    print(train_size, test_size)

# Batch and shuffle the data
    train_dataset = tf.data.Dataset.from_tenso
r_slices(data[:train_size]).shuffle(train_size
).batch(BATCH_SIZE)

    test_dataset = tf.data.Dataset.from_tenso
r_slices(data[train_size:]).shuffle(test_size)
.batch(1)

    seed = tf.random.normal(dim)

    return seed, train_dataset, test_dataset
2.2.0

```

```

def make_generator_model():
    model = tf.keras.Sequential()
    model.add(layers.Input(shape=(100, 12)))

    model.add(layers.Bidirectional(layers.LSTM
(64, return_sequences=True)))

    model.add(layers.Conv1D(filters=128, kernel_size=16,
strides=1, padding='same'))
    model.add(layers.LeakyReLU())

    model.add(layers.Conv1D(filters=64, kernel_size=16,
strides=1, padding='same'))
    model.add(layers.LeakyReLU())

    model.add(layers.UpSampling1D(2))

    model.add(layers.Conv1D(filters=32, kernel_size=16,
strides=1, padding='same'))
    model.add(layers.LeakyReLU())

    model.add(layers.Conv1D(filters=16, kernel_size=16,
strides=1, padding='same'))
    model.add(layers.LeakyReLU())

    model.add(layers.UpSampling1D(2))

    model.add(layers.Conv1D(filters=1, kernel_size=16,
strides=1, padding='same', activation='tanh'))

    model.add(layers.Permute((2, 1)))

    return model

def make_discriminator_model():
    model = tf.keras.Sequential()

    model.add(layers.Input(shape=(1, 400)))
    model.add(layers.Permute((2, 1)))

    model.add(layers.Conv1D(filters=32, kernel_size=16,
strides=1, padding='same'))
    model.add(layers.LeakyReLU())

    # model.add(layers.Dropout(0.4))

    model.add(layers.Conv1D(filters=64, kernel_size=16,
strides=1, padding='same'))
    model.add(layers.LeakyReLU())

    model.add(layers.Conv1D(filters=128, kernel_size=16,
strides=1, padding='same'))
    model.add(layers.LeakyReLU())

    # model.add(layers.Dropout(0.4))

    model.add(layers.Conv1D(filters=256, kernel_size=16,
strides=1, padding='same'))
    model.add(layers.LeakyReLU())

    model.add(layers.MaxPool1D(pool_size=2))

    model.add(layers.Flatten())
    model.add(layers.Dense(1))

    return model

generator = make_generator_model()
discriminator = make_discriminator_model()

generator.summary()
discriminator.summary()

# test_show(generator, discriminator)

# !rm -rf ./logs/

### discriminator loss ###
# disc_log_dir = 'logs/gradient_tape/disc_loss'
# disc_summary_writer = tf.summary.create_file_writer(disc_log_dir)
disc_losses = tf.keras.metrics.Mean('disc_loss', dtype=tf.float32)
disc_losses_list = []

### discriminator accuracy ###
fake_disc_accuracy = tf.keras.metrics.BinaryAccuracy('fake_disc_accuracy')
real_disc_accuracy = tf.keras.metrics.BinaryAccuracy('real_disc_accuracy')
fake_disc_accuracy_list, real_disc_accuracy_list = [], []

### generator loss ###
# gen_log_dir = 'logs/gradient_tape/gen_loss'

```



```

# gen_summary_writer = tf.summary.create_file_
writer(gen_log_dir)
gen_losses = tf.keras.metrics.Mean('gen_loss',
dtype=tf.float32)
gen_losses_list = []

#         Prepare loss functions and optimize
rs

# This method returns a helper function to com
pute cross entropy loss
cross_entropy = tf.keras.losses.BinaryCrossent
ropy(from_logits=True)

def discriminator_loss(real_output, fake_outpu
t):
    real_loss = cross_entropy(tf.ones_like(rea
l_output), real_output)
    fake_loss = cross_entropy(tf.zeros_like(fa
ke_output), fake_output)
    total_loss = real_loss + fake_loss
    return total_loss

def generator_loss(fake_output):
    return cross_entropy(tf.ones_like(fake_out
put), fake_output)

generator_optimizer = tf.keras.optimizers.Adam
(0.0002)
discriminator_optimizer = tf.keras.optimizers.
Adam(0.0002)
disc_steps = 1

# Notice the use of `tf.function`
# This annotation causes the function to be "c
omplied".
@tf.function
def train_step(real_ecg, dim):
    noise = tf.random.normal(dim)

    for i in range(disc_steps):
        with tf.GradientTape() as disc_tape:
            generated_ecg = generator(noise, t
raining=True)

            real_output = discriminator(real_e
cg, training=True)
            fake_output = discriminator(genera
ted_ecg, training=True)

            disc_loss = discriminator_loss(rea
l_output, fake_output)

            gradients_of_discriminator = disc_tape
.gradient(disc_loss, discriminator.trainable_v
ariables)
            discriminator_optimizer.apply_gradient
s(zip(gradients_of_discriminator, discriminato
r.trainable_variables))

            ### for tensorboard ###
            disc_losses.update_state(disc_loss)
            fake_disc_accuracy.update_state(tf.zer
os_like(fake_output), fake_output)
            real_disc_accuracy.update_state(tf.one
s_like(real_output), real_output)
            #####

            with tf.GradientTape() as gen_tape:
                generated_ecg = generator(noise, train
ing=True)
                fake_output = discriminator(generated_
ecg, training=True)

                gen_loss = generator_loss(fake_output)

                gradients_of_generator = gen_tape.gradient
(gen_loss, generator.trainable_variables)
                generator_optimizer.apply_gradients(zip(gr
adients_of_generator, generator.trainable_vari
ables))

            ### for tensorboard ###
            gen_losses.update_state(gen_loss)
            #####

def train(dataset, epochs, dim):
    for epoch in tqdm(range(epochs)):

        for batch in dataset:
            train_step(batch, dim)

            disc_losses_list.append(disc_losses.re
sult().numpy())
            gen_losses_list.append(gen_losses.resu
lt().numpy())

```

```

fake_disc_accuracy_list.append(fake_disc_accuracy.result().numpy())
real_disc_accuracy_list.append(real_disc_accuracy.result().numpy())

### for tensorboard ###

disc_losses.reset_states()
gen_losses.reset_states()

fake_disc_accuracy.reset_states()
real_disc_accuracy.reset_states()
#####

# Save the model every 5 epochs
#     if (epoch + 1) % 5 == 0:
#         generate_and_save_ecg(generator,
epochs, seed, False)
#         checkpoint.save(file_prefix = checkpoint_prefix)

# Generate after the final epoch
display.clear_output(wait=True)
generate_and_save_ecg(generator, epochs, seed, False)

#     generator.save('/content/drive/My Drive/Colab Notebooks/saved_models/generator_1500.h5')
#     discriminator.save('/content/drive/My Drive/Colab Notebooks/saved_models/discriminator_1500.h5')

In [3]:
BATCH_SIZE = 64
noise_dim = [BATCH_SIZE, 100, 12]

seed, train_dataset, test_dataset = prepare_data(noise_dim)
Data shape: (7621, 1, 400)

6858 763

In [0]:
train(train_dataset, 5, noise_dim)

In [0]:
# %reload_ext tensorboard
# %tensorboard --logdir logs/gradient_tape

fig, axes = plt.subplots(2, figsize=(12, 8))
fig.suptitle('Training Metrics')

```

```

axes[0].set_ylabel("Losses", fontsize=14)
axes[0].set_xlabel("Epoch", fontsize=14)
axes[0].plot(disc_losses_list, color='red')
axes[0].plot(gen_losses_list, color='blue')

axes[1].set_ylabel("Accuracy", fontsize=14)
axes[1].set_xlabel("Epoch", fontsize=14)
axes[1].plot(fake_disc_accuracy_list, color='red')
axes[1].plot(real_disc_accuracy_list, color='blue')
plt.show()

seed = tf.random.normal(noise_dim)
ecg = generator(seed, training=False)
answer = discriminator(ecg)
fig = plt.figure(figsize=(4, 3))
plt.plot(ecg[0, 0, :] * norm_value)
plt.show()
print(answer[0])

tf.Tensor([0.7015703], shape=(1,), dtype=float32)

generator.save('/drive/My Drive/Colab Notebooks/saved_models/generator_80e.h5')
discriminator.save('/drive/My Drive/Colab Notebooks/saved_models/discriminator_80e.h5')

model = tf.keras.models.load_model('generator_80e.h5')
WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.

model.summary()

seed = tf.random.normal([1, 100, 12])
ecg = model(seed, training=False)
fig = plt.figure(figsize=(4, 3))
plt.plot(ecg[0, 0, :] * norm_value)
plt.show()

# Testing functions
def rmse(targets, predictions):
    return np.sqrt(np.mean((targets - predictions)**2))

def prd(targets, predictions):

```

```

s1 = np.sum((targets-predictions)**2)
s2 = np.sum(targets**2)
return np.sqrt(s1 / s2 * 100)

def mmd(targets, predictions):
    mmd_stat = MMDStatistic(400, 400)
    sample_target = torch.from_numpy(targets.numpy().reshape((400,1)))
    sample_pred = torch.from_numpy(predictions.numpy().reshape((400,1)))

    stat = mmd_stat(sample_target, sample_pred, [1.])
    return (stat.item())

def testing(test_dataset, model, noise_dim):
    noise = tf.random.normal(noise_dim)
    generated_ecgs = model(noise, training=False)

    mmd_sum, prd_sum, rmse_sum = [], [], []

    for true_ecg, gen_ecg in zip(test_dataset, generated_ecgs):
        prd_sum.append(prd(true_ecg[0][0], gen_ecg[0]))
        rmse_sum.append(rmse(true_ecg[0][0], gen_ecg[0]))
        mmd_sum.append(mmd(true_ecg[0][0], gen_ecg[0]))

    print('mmd :', f'mean={np.mean(mmd_sum):.6f}', f'min={np.min(mmd_sum):.6f}', f'max={np.max(mmd_sum):.6f}')
    print('prd :', f'mean={np.mean(prd_sum):.4f}', f'min={np.min(prd_sum):.4f}', f'max={np.max(prd_sum):.4f}')
    print('rmse:', f'mean={np.mean(rmse_sum):.4f}', f'min={np.min(rmse_sum):.4f}', f'max={np.max(rmse_sum):.4f}')

testing(test_dataset, model, [763, 100, 12])
mmd : mean=0.000944 min=-0.000062 max=0.011455
prd : mean=14.8219 min=4.5564 max=49.1954
rmse: mean=0.0780 min=0.0203 max=0.1678

```