НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Автоматизованих систем обробки інформації і управління
(повна назва кафедри)

«На правах рукопису»
УДК 004.89

До захисту допущено:
В.о. завідувача кафедри
_____ Олександр ПАВЛОВ
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інженерія програмного забезпечення комп'ютеризованих систем»

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: «Програмне забезпечення створення моделі Word2vec на основі сигналу ЕКГ**»**

Виконав:
студент VI курсу, групи ІП-92мп
Терещенко Андрій Сергійович _____

Науковий керівник:
ст. вик., кафедри АСОІУ, Олійник Юрій Олександрович _____

Рецензент:
доц., кафедри ТК, к.т.н., Ткач Михайло Мартинович _____

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

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Автоматизованих систем обробки інформації і управління

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма - «Інженерія програмного забезпечення комп'ютеризованих систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Олександр ПАВЛОВ
«___» грудня 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Терещенку Андрію Сергійовичу

1. Тема дисертації «Програмне забезпечення створення моделі Word2vec на основі сигналу ЕКГ», науковий керівник дисертації старший викладач Олійник Юрій Олександрович затверджені наказом по університету від «26» жовтня 2020 р. №3132-с

2. Термін подання студентом дисертації *2 грудня 2020 р.*

3. Об'єкт дослідження *Аналіз ЕКГ за рахунок створення моделі Word2Vec*

4. Вхідні дані *Запис сигналу ЕКГ*

5. Перелік завдань, які необхідно розробити *виконати огляд існуючих методів та алгоритмів аналізу ЕКГ; розглянути основні етапи створення Word2Vec моделі; провести аналіз існуючих методів кластеризації виділених хвиль з сигнала; спроектувати та розробити власну бібліотеку для створення Word2Vec моделі на основі сигналу ЕКГ*

6. Орієнтований перелік графічного (ілюстративного) матеріалу *Етапи створення Word2Vec моделі; Дослідження ефективності роботи Word2Vec моделі; Екранні форми роботи програмного забезпечення;*

7. Орієнтований перелік публікацій *Тези доповіді у матеріалах III всеукраїнській науково-практичній конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020)*

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Графічний	доц. Ліщук К.І.		

9. Дата видачі завдання 2 вересня 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	*Систематизація результатів огляду літератури*	*09.09.2020*	
2	*Порівняльний аналіз існуючих методів вирішення задачі*	*16.09.2020*	
3	*Формування постановки задачі*	*23.09.2020*	
4	*Модифікація існуючих методів розв'язання задачі*	*25.09.2020*	
5	*Розробка інформаційного та програмного забезпечення*	*01.10.2020*	
6	*Проведення досліджень ефективності роботи алгоритму*	*30.10.2020*	
7	*Оформлення документації*	*11.11.2020*	
8	*Подання роботи на попередній захист*	*27.11.2020*	
9	*Подання роботи на основний захист*	*02.12.2020*	

Студент А.С. Терещенко

Науковий керівник Ю.О. Олійник

РЕФЕРАТ

Магістерська дисертація: 86 с., 17 рис, 25 таб., 2 додатки, 27 джерел.

Актуальність теми: Серцеві захворювання займають значний відсоток серед причин смертності як в Україні так і в більшості країн світу. Для прикладу щороку в Україні понад 68% осіб помирають через серцево-судинні хвороби. Важливим фактором в боротьбі з хворобою є профілактика та виявлення захворювання на ранніх стадіях. Одним із основних методів діагностики серця є електрокардіографія, тому дуже важливо швидко та точно провести аналіз електрокардіограми (ЕКГ).

Мета дослідження: розширення можливостей автоматичного аналізу електрокардіограм за рахунок створення Word2Vec моделі на основі виділених хвиль в ЕКГ.

Об'єкт дослідження: електрокардіограми.

Предмет дослідження: векторні моделі даних та засоби аналізу даних методами NLP.

Методи дослідження: у даній дисертаційній роботі застосовувалися методи обробки природної мови, засновані на правилах, словниках та існуючих лінгвістичних ресурсах, і ймовірнісних тематичних моделях, заснованих на комплексі методів машинного навчання.

Наукова новизна: новий підхід для представлення у векторній структурі серцевого такту, можливість знаходження різниці між сигналами за рахунок обрахунку косинуса подібності та застосування алгоритму TextRank для знаходження ключових тактів, що показує важливість серцебиття.

Практичне значення отриманих результатів визначається тим, що запропонований підхід до аналізу ЕКГ зменшує об'єм даних для аналізу ЕКГ, показує гарну точність аналізу.

Зв'язок роботи з науковими програмами, планами, темами: робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський

політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Методи та технології високопродуктивних обчислень та обробки надвеликих масивів даних». Державний реєстраційний номер 0117U000924.

Апробація: Основні положення роботи доповідались і обговорювались на III всеукраїнській науково-практичній конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020)

Публікації: Наукові положення дисертації опубліковані в тезах конференції «ІНФОРМАТИКА ТА ОБЧИСЛЮВАЛЬНА ТЕХНІКА – ІОТ-2020»

Ключові слова: ЕКГ, WORD2VEC, МЕТОДИ ОБРОБКИ ПРИРОДНОЇ МОВИ, ВЕКТОРНЕ ПРЕДСТАВЛЕННЯ СЛІВ.

ABSTRACT

Master's dissertation consists 86 pages, 17 images, 25 tables, 27 referring sources.

Topicality: Heart disease accounts for a significant percentage of deaths in both Ukraine and most countries. For example, every year in Ukraine more than 68% of people die from cardiovascular disease. An important factor in the fight against the disease is the prevention and detection of the disease in its early stages. One of the main methods of diagnosing the heart is electrocardiography, so it is very important to quickly and accurately analyze the electrocardiogram (ECG).

The purpose of the dissertation research is expanding the capabilities of automatic analysis of electrocardiograms by creating a Word2Vec model based on selected waves in the ECG

Object of study: electrocardiograms.

Subject of research: processing of vector representation of ECG signal by NLP methods.

Research Methods: In this dissertation, natural language processing methods based on rules, dictionaries and existing linguistic resources, and probabilistic thematic models based on a set of machine learning methods have been applied.

Scientific novelty: new approach for representation in the vector structure of the heart rate, the ability to find the difference between the signals by calculating the cosine of similarity and the use of the TextRank algorithm to find key beats, which shows the importance of heartbeat.

The practical value of the obtained results is determined by the fact that the proposed approach to ECG analysis reduces the amount of data for ECG analysis, shows good accuracy of the analysis.

Relationship with working with scientific programs, plans, topics: work was performed at the Department of Automated Information Processing and Management Systems of the National Technical University of Ukraine «Igor Sikorsky Kyiv

Polytechnic Institute» within the topic «Methods and technologies of high-performance computing and processing of large data sets». State Registration Number 0117U000924.

Testing: The main provisions of the work were reported and discussed at the conference "Informatics and Computer Engineering - IOT-2020".

Publications: Theses of the thesis are published in « Informatics and Computer Engineering - IOT-2020».

Keywords: ECG, WORD2VEC, NATURAL LANGUAGE PROCESSING METHODS, WORD EMBEDDING.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕКГ – електрокардіограма.

NLP (Natural Language Processing) – обробка природної мови

T-SNE – стохастичне вкладення сусідів з t-розподілом

CBOW (Continuous Bag of Words) – алгоритм побудови word2vec моделі

PCA (Principal Components Analysis) – метод головних компонент

AFIB (Atrial fibrillation) – миготлива аритмія

ВСТУП

Серцеві захворювання займають значний відсоток серед причин смертності як в Україні так і в більшості країн світу. Для прикладу щороку в Україні понад 68% осіб помирають через серцево-судинні хвороби. Важливим фактором в боротьбі з хворобою є профілактика та виявлення захворювання на ранніх стадіях. Безперечно чим раніше виявляються серцево-судинні захворювання, тим ефективнішим буде їх лікування. Одним із основних методів діагностики серця є електрокардіографія, тому дуже важливо швидко та точно зробити аналіз електрокардіограми (ЕКГ).

Виконання вручну аналізу ЕКГ-сигналів людиною є дуже складним і трудомістким завданням через довгі записи ЕКГ та існування складних закономірностей, пов'язаних з різною аритмією серця. Тому потрібно замінити ручний аналіз сигналів ЕКГ, зосередившись на розробці методів автоматичного аналізу ЕКГ для виконання цього завдання з високою точністю та у режимі реального часу.

З розвитком машинного навчання вдалося автоматизувати досить складні процеси: розпізнавання об’єктів, прогнозування подій, аналіз різноманітних даних. Було б корисно створити модель, яка змогла аналізувати ЕКГ та виявляти чи прогнозувати захворювання.

Аналіз існуючих підходів до вирішення подібних проблем допомагає більш детально дослідити предметну область та спробувати створити удосконалену модель для покращення швидкості та точності виявлення тих чи інших серцевих захворювань. Одним із методів дослідження для аналізу сигналів ЕКГ є структура, яка називається мова обробки ЕКГ. Вона обробляє сигнал ЕКГ подібно до обробки текстового документу методами природної мови.

Подібно до природних мов сигнал ЕКГ складається з послідовностей з трьох або чотирьох різних хвиль, включаючи P-хвилю, комплекс QRS, T-хвилю та U-хвилю. ЕКГ сигнал є послідовністю серцебиття (подібно до речень в природній мові), і кожне серцебиття складається з сукупності хвиль (подібних до слів у

реченні) різної морфології. Аналогічно обробці природної мови (NLP), яка використовується, щоб допомогти комп'ютерам зрозуміти та інтерпретувати природну мову людини, можна розробити методи NLP, що допоможуть комп'ютерам глибше зрозуміти сигнали електрокардіограми. Техніка аналізу ЕКГ, зосереджена на розширенні можливостей комп'ютерів розуміти сигнали ЕКГ в так, як це роблять лікарі.

Дана магістерська дисертація присвячена модернізації та пошуку рішень для можливості аналізу електрокардіограм. Доцільність дослідження обґрунтовано за рахунок необхідності автоматизації аналізу діагностики серцевих захворювань, які є однією з найбільших причин смертності людей.

Тому головною **метою дослідження є** розширення можливостей автоматичного аналізу електрокардіограм за рахунок створення Word2Vec моделі на основі ЕКГ сигналу.

До **об'єкту дослідження** відносяться електрокардіограми, а до **предмету дослідження** відноситься векторні моделі даних та засоби аналізу даних методами NLP.

Науковою новизною є новий підхід для представлення у векторній структурі сигналу ЕКГ та серцевого такту зокрема, можливість знаходження різниці між сигналами за рахунок обрахунку косинуса подібності та застосування алгоритму TextRank для знаходження ключових тактів, що показує важливість серцебиття.

1 ТЕОРЕТИЧНІ ОСНОВИ

1.1 ЕКГ сигнал

Для інструментального дослідження діяльності серцевого м’яза використовують електрокардіографію. Дослідження може проводитися в стані спокою, при фізичному навантаженні і при використанні деяких спеціальних лікарських препаратів – під час ЕКГ визначаються стан серцевого м’яза, ритм серця, кровотік в міокарді.

Електрокардіографія – це метод графічної реєстрації електричних явищ, які виникають у серцевому м’язі під час його діяльності, з поверхні тіла. Криву, яка відображає електричну активність серця, називають електрокардіограмою (ЕКГ). Таким чином, ЕКГ – це запис коливань різниці потенціалів, які виникають у серці під час його збудження.(Рисунок 1.1)

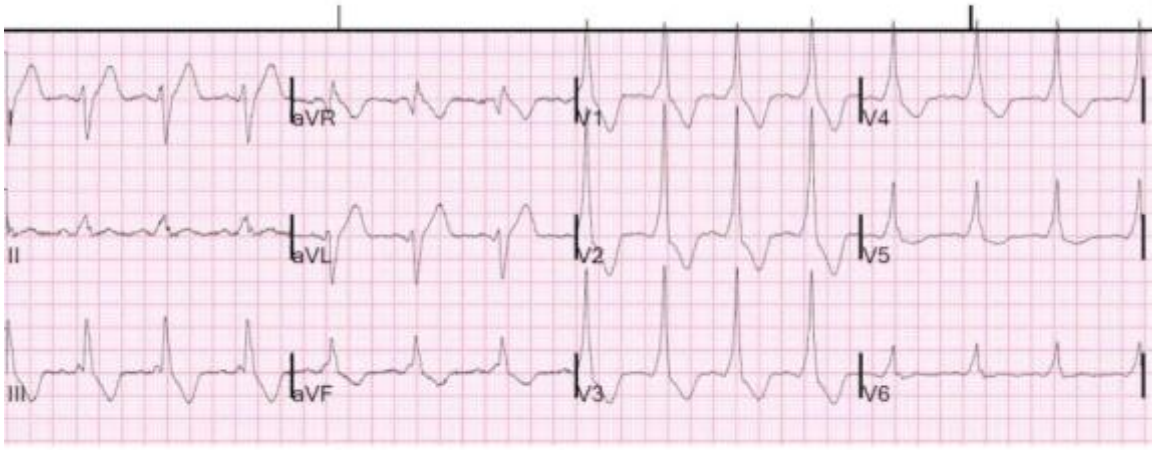


Рисунок 1.1 – Приклад ЕКГ

Електрокардіографія це один з найбільш ефективних способів дослідження серця та діагностики серцево-судинних захворювань. ЕКГ є незамінним у діагностиці порушень ритму і провідності та ішемічної хвороби серця. За допомогою даного методу ми можемо з високою точністю спостерігати та досліджувати зміни міокарда, їх розповсюдженість, глибину і час появи. ЕКГ дозволяє виявити дистрофічні й склеротичні процеси в міокарді, порушення електролітного обміну, що виникають під впливом різних токсичних речовин. ЕКГ широко використовують для функціонального дослідження серцево-судинної

системи. Поєднання електрокардіографічного дослідження з функціональними пробами допомагає виявити приховану коронарну недостатність, перехідні порушення ритму, проводити диференційний діагноз між функціональними та органічними порушеннями роботи серця.[1]

1.2 Лінгвістичний метод представлення сигналу ЕКГ

Для пошуку аномалій було обрано саме метод лінгвістичних ланцюгів тому що він є швидким та призначений для порівняння коротких інтервалів. (На які можна розбити ЕКГ відповідно тому як показано на рисунку 3).

Основною концепцією даного методу [2] є співвідношення числового проміжку до певної літери. Довжина числового проміжку може бути підібрана різною відповідно до розподілу, обраного символьного алфавіту та самого діапазону значень числового ряду.

Приклад відповідного перетворення показаний на рисунку 1.2, де для спрощення відображення було обрано англійський алфавіт та кожній літері відповідав однаковий проміжок.

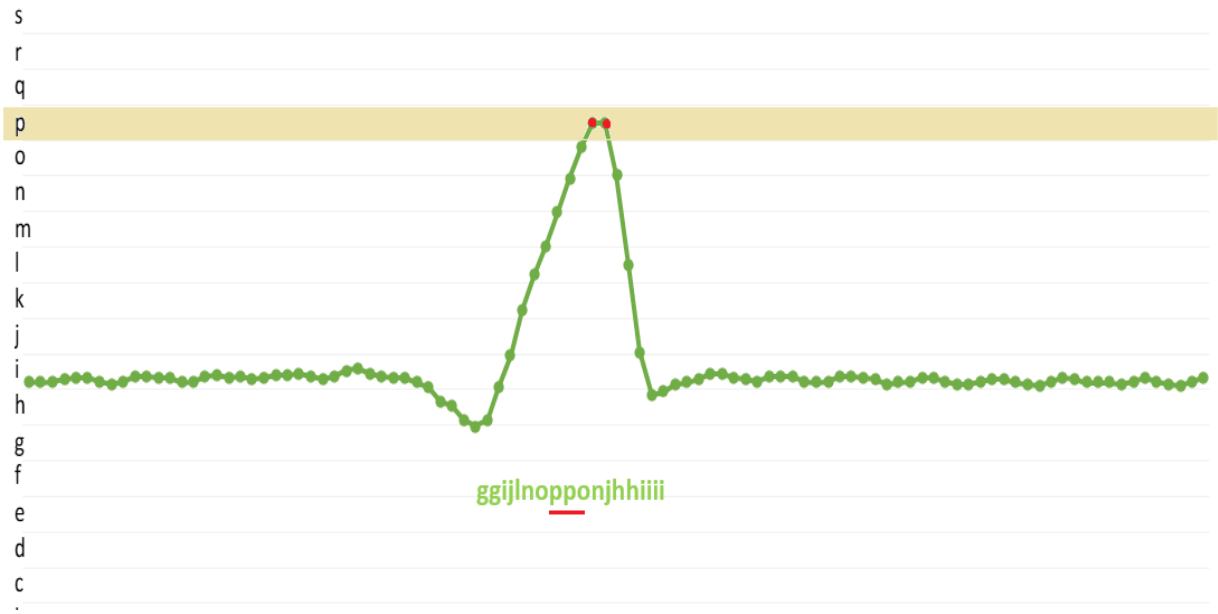


Рисунок 1.2 – Присвоєння числу певної літери

На рисунку 1.2 відображення переведення 2 точок з значенням 1157 ,які відповідають літері p , яка має діапазон значень 1145 по 1168. Таким чином опрацьовуємо всі вхідні дані.

При використанні лінгвістичного підходу варто зазначити необхідність отримання початкового промаркованого набору з аномаліями для їх пошуку в подальшому. При пошуку аномалій в справжніх даних будуть використані алгоритми для пошуку відстані між рядками: відстань Левеншейна, відстань Геммінга, Джаро-Вінклера , Дамерау-Левенштейна.

Недоліком даного методу є представлення хвилі через лінгвістичний ланцюжок. Але так як хвилі одного типу мають різні сигнали то і різні лінгвістичні ланцюжки, що ускладнює їх обробку.

1.3 Алгоритми виявлення QRS комплексу в ЕКГ

1.3.1 Алгоритм Пана-Томпкінса

Алгоритм Пана – Томпкінса [3] зазвичай використовується для виявлення комплексів QRS в електрокардіографічних сигналах. Комплекс QRS являє собою деполяризацію шлуночків обох передсердець, тому вважається основним об’єктом для аналізу ЕКГ-сигналу. Ця особливість робить його особливо придатним для вимірювання пульсу та першого засобу для оцінки стану здоров'я серця. У першій варіації запропонованій Ейнтовеном фізіологічної точки зору на серце, комплекс QRS складається з відхилення вниз (хвиля Q), високого відхилення вгору (хвиля R) і остаточного відхилення вниз (хвиля S).

Алгоритм Пана – Томпкінса застосовує серію фільтрів, щоб виділити частотний вміст швидкої деполяризації серця та усуває фоновий шум. Приклад фільтрації ЕКГ можемо бачити на рисунку 1.3. Потім він підносить в квадрат сигнал для посилення розподілу QRS. Далі він застосовує адаптивні пороги для виявлення піків у відфільтрованому сигналі. Алгоритм був запропонований Джіапу Паном та Уіллісом Дж. Томпкінсом у 1985 р. У журналі “IEEE Transactions on Biomedical Engineering”. Ефективність методу тестували на базі даних анотованих аритмій та оцінювали також за наявності шуму. За результатами

алгоритм Пана і Томпкінса показав, що 99,3 відсотка комплексів QRS було правильно виявлено.

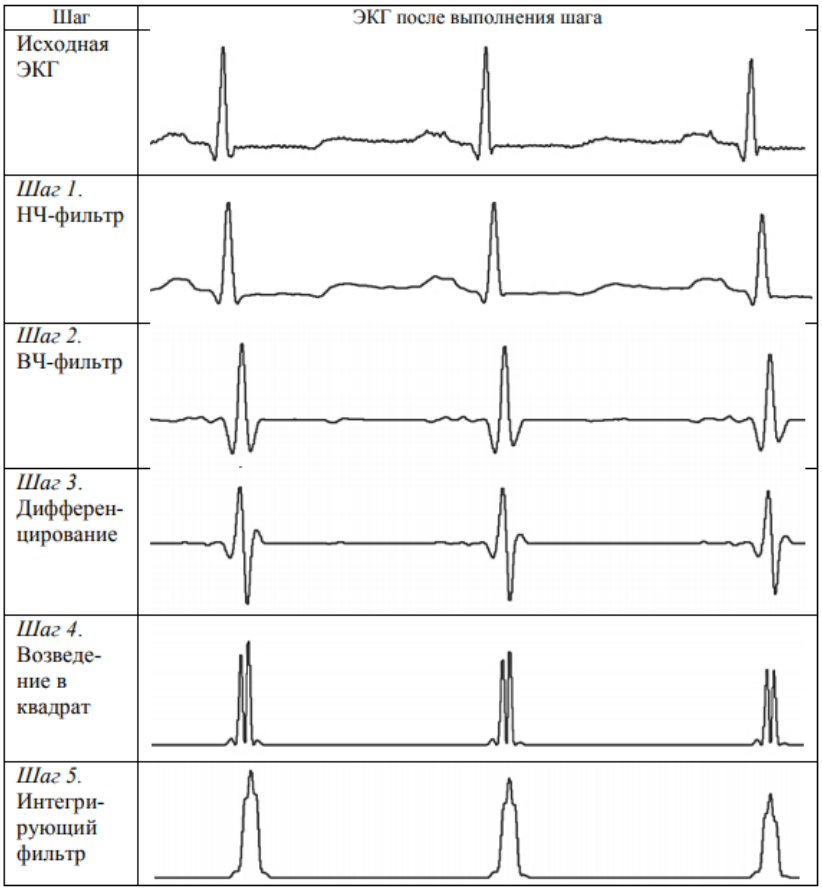


Рисунок 1.3 – Застосування фільтрів до вхідного сигналу ЕКГ

1.3.2 Алгоритм Енгельсе та Зеленберга

Алгоритм Енгельсе та Зеленберга[4] був запропонований у 1979 р. Він використовується для виявлення R піків в сигналі ЕКГ. Спершу до вхідного сигналу застосовується диференціатор та накладаються фільтри низьких частот. Після цього відбувається оцінка в поточному вікні порогового значення для R піка. Перевіряється умова чи є пік максимальним у заданому проміжку. Якщо так то додаємо значення до нашого результату. Порогові значення кожного разу визначаються за допомогою функції максимальної амплітуди сигналу. Після визначення QRS комплексу в поточному вікні переходимо до наступного і проводимо попередньо описанні дії. В результаті отримуємо значення усіх R піків для заданого сигналу ЕКГ.

1.3.3 Алгоритм Гамільтона

У 2002 році Гамільтон запропонував комплексний алгоритм[5] виявлення QRS комплексу, який працює, скануючи сигнал ЕКГ та робить оцінку за наступним алгоритмом:

- ігноруються всі піки, що передують чи слідують за більшими піками на проміжку менше ніж 200 мс;

- якщо пік більший за поріг виявлення, то це називається комплексом QRS, інакше називається шумом;

- якщо з часу останнього виявлення пройшов інтервал, рівний півтора кратному середньому інтервалу між піками, то у межах цього інтервалу був пік, який був більше за половину порога виявлення, і пік слідував за попереднім виявленням принаймні 360 мс, класифікують цей пік як комплекс QRS;

- поріг виявлення є функцією середнього значення шуму та середніх пікових значень QRS;

- середнє значення шуму, середній пік QRS і середні оцінки інтервалу від R до R розраховуються як середнє значення / медіана останніх восьми значень.

1.4 Кластерний аналіз

Кластерний аналіз[6] представляє собою цінний інструмент аналізу даних в сучасних застосуваннях машинного навчання та інтелектуального аналізу даних. У багатьох випадках кластеризація використовується для отримання перших відомостей про дані в процесі аналізу і для вирішення ряду реальних проблем, таких як сегментація клієнтів в маркетингових компаніях, усунення несправностей в моніторингу промислових процесів, моделювання тем в інтелектуальному аналізі тексту, сегментація зображень в комп’ютерному зорі. Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу задач навчання без учителя.

Кластерний аналіз став популярним серед деяких науковців, які займаються дослідженнями машинного навчання без вчителя. Існує багато методів кластерного аналізу, і важко судити про їх відносні переваги та недоліки, оскільки поняття

кластера не є чітко визначеною концепцією. Цілком можливо, що потрібно визначити кілька різних типів кластера, і в цьому випадку будь-хто може ретельно продумати, яке визначення відповідає його вимогам. З цим визначенням можна робити спроби розробки алгоритмів, щоб мати змогу оперувати різними даними. Можна знайти більше одного підходящого алгоритму, але всі вони повинні давати однакові результати, за винятком даних що можуть підходити для декількох кластерів. На практиці процедура зазвичай зворотна, алгоритм неявно визначає кластер.

Кластерний аналіз є досить корисним для багатьох сфер діяльності: його використовують в медицині, хімії, геології, біології, державному управлінні, філології, археології, соціології, та інших дисциплінах.[7]

Основні завдання кластеризації:

- вивчення процесу групування об'єктів в моделі;
- розробка типології або класифікації;
- породження нових гіпотез взаємозв’язку між даними певної структури;
- перевірка гіпотез на рахунок виділених взаємозв’язків між даними.

Етапи для виконання кластеризації:

- створення початкової вибірки для кластеризації;
- визначення ознак за якими буде відбуватися розподіл даних на кластери;
- розрахунок подібності між об’єктами вибірки;
- застосування одного з алгоритмів кластеризації;
- тестування правильності результатів.

1.5 Методи кластеризації

1.5.1 K-середніх

Відмінною рисою даного методу[8] є наявність центроїдів кожного кластера. Центроїдом є точка, яка перебуває посередині кластера. Кожен розглянутий об’єкт буде ставитися до кластеру, центроїд якого знаходиться найближче. На першому етапі центроїди кластерів обираються випадковим чином або за певним правилом

(наприклад вибрати центроїди, що максимізують початкову відстань між кластерами). Наступний етап це віднесення кожного об’єкту до певного кластеру. Для кожного об’єкта рахується відстань до всіх центроїдів, після цього обирається найближчий. Після цього відбувається перерахунок координат центроїдів. Це повторюється на кожному кроці, доки координати центроїдів перестануть змінюватись. Після цього роботу алгоритму можемо вважати завершеною.

Простий приклад кластеризації можемо спостерігати на рисунку 1.4.

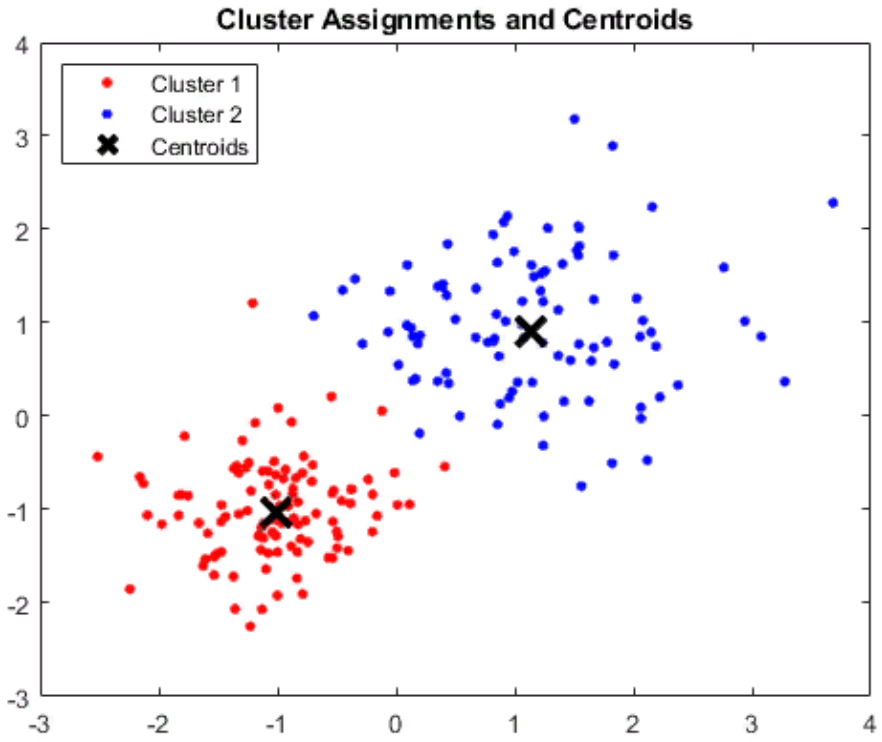


Рисунок 1.4 – Кластеризація методом k-means

Особливості методу k-середніх:

- використання різних метрик відстані в залежності від об’єктів кластеризації;

- якість кластеризації залежить від першочергового вибору центроїдів;

- кількість кластерів не визначається автоматично, а задається дослідником.

1.5.2 Ієрархічна кластеризація

Ієрархічна кластеризація - це сукупність алгоритмів[9] для впорядкування даних, які базуються на створенні ієрархії вкладених кластерів. Виділяють два класи методів ієрархічної кластеризації: агломеративний та розділювальний.

У агломеративного-ієрархічних методах спочатку всі об'єкти розглядаються як окремі, самостійні кластери, що складаються всього лише з одного елемента. Процедура кластеризації полягає в поступовому об'єднанні об'єктів в досить великі кластери, використовуючи деяку міру подібності або відстань між об'єктами. Типовим результатом такої кластеризації є ієрархічне дерево.(Рисунок 1.5)

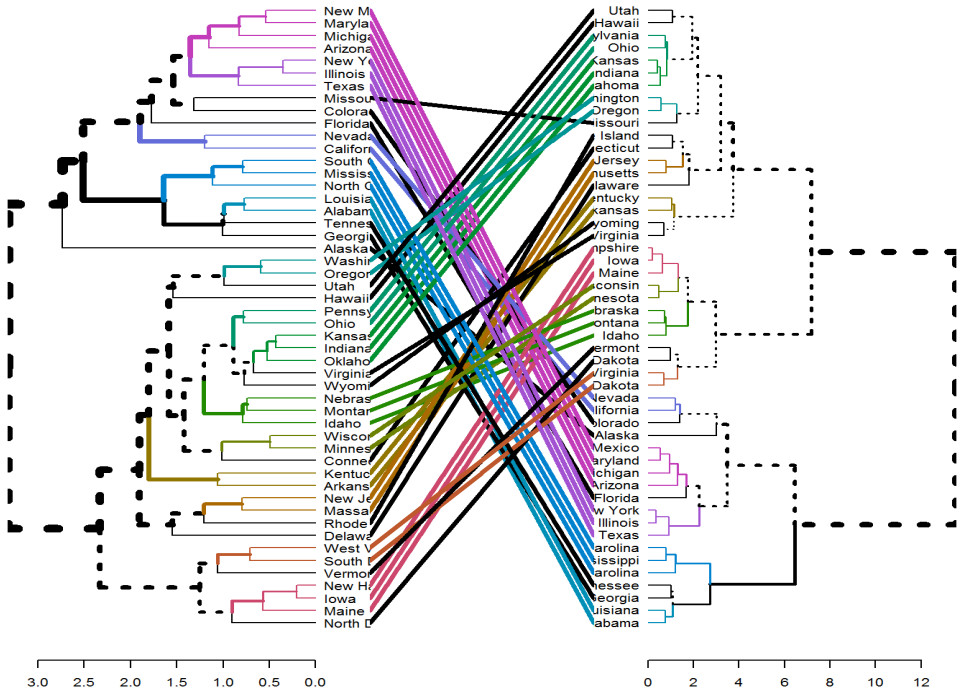


Рисунок 1.5 – Приклад ієрархічного дерева

На першому кроці, коли кожен об'єкт являє собою окремий кластер, відстані між цими об'єктами визначаються обраної мірою. Однак коли зв'язуються разом декілька об'єктів, виникає питання, як слід визначити відстані між кластерами?

Існує багато методів і алгоритмів об'єднання кластерів, перерахуємо деякі з них.

- одиночний зв'язок (метод найближчого сусіда). Відстань між кластерами визначається відстанню між найближчими сусідами в різних кластерах;

- повний зв'язок (метод найбільш віддалених сусідів). Випадок коли відстань між кластерами обчислюється, як найбільша відстань між двома довільними об'єктами в різних кластерах;

- незважене попарне середнє. Тут відстань між двома різними кластерами визначається як середня відстань між парами всіх об'єктів в кластерах;

- зважене попарне середнє. Метод ідентичний методу невиваженого попарного середнього, за винятком того, що при обчисленнях розмір відповідних кластерів використовується в якості вагового коефіцієнта;

- незважений центроїдний метод. У цьому методі відстань між двома кластерами визначається як відстань між їх центрами тяжкості. Зважений центроїдний метод (медіана). Цей варіант ідентичний попередньому, за винятком того, що при обчисленнях використовуються ваги для обліку різниці між розмірами кластерів.

Питання, яке природно виникає в агломеративному методі - коли припинити об'єднання кластерів. Для цього використовують число отриманих кластерів, міжкластерну відстань, максимальний стрибок в зміні міжкластерної відстані. Також використовуються основні статистичні характеристики кластерів, як кількість об'єктів в кластері, середні значення ознак в кожному кластері, дисперсія.

В розділювальному методі на початковому етапі вся вибірка розглядається як єдиний кластер, а потім вже починається процес його розподілу на складові частини. Процес ділення продовжується до тих пір, поки кожне спостереження не перетвориться в окремий кластер. Оскільки дані алгоритми оперують відстанями між спостереженнями, то в деяких програмах передбачена можливість роботи не з вихідної матрицею «об'єкт-ознака», а з симетричною матрицею відстаней між спостереженнями.

Приймаючи рішення про те, який з цих способів вибрати, завжди має сенс спробувати всі варіанти, проте в цілому агломератна кластеризація краще підходить для виявлення невеликих кластерів і використовується більшістю

комп'ютерних програм, а розділювальна кластеризація доцільніше для виявлення великих кластерів.

1.5.3 Спектральна кластеризація

Спектральна кластеризація[10] є одним із найбільш ефективних алгоритмів кластеризації, завдяки своїй здатності розділяти нелінійні дані. Ефективність алгоритму пояснюється тим, що дані з початкового простору відображаються в новий простір в якому їх можна лінійно розділити. Основним недоліком даного алгоритму є кубічна обчислювальна складність.

1.6 Методи пониження розмірності

1.6.1 Аналіз головних компонент

Метод головних компонент (PCA) — метод[11] зменшення розмірності, який використовує ортогональне перетворення множини великого набору змінних у менший, який все ще містить найбільшу частину інформації з попереднього набору даних.

Метод головних компонент — один з основних способів зменшити розмірність даних, втративши найменшу кількість інформації. Винайдений Карлом Пірсоном у 1901 році та доповнений і розширений Гарольдом Готелінґом в 1933 р. Застосовується в багатьох галузях, зокрема, в економетриці, біоінформатиці, обробці зображень, для стиснення даних, у суспільних науках.

Зменшення кількості змінних набору даних, природно, відбувається за рахунок зменшення точності, але фокус у зменшенні розмірності полягає в тому, щоб продати трохи точності для простоти. Оскільки менші набори даних легше досліджувати та візуалізувати, а аналіз даних робиться набагато простішим та швидшим для алгоритмів машинного навчання без сторонніх змінних для обробки.

Задача аналізу головних компонент має щонайменше чотири базових версії:

- апроксимувати дані для зменшення розмірності;

- знайти підпростори меншої розмірності, в ортогональній проекції на які розкид даних;

- знайти підпростори меншої розмірності, в ортогональній проекції на які середньоквадратична відстань між точками максимальна;

- для даної багатовимірної випадкової величини побудувати таке ортогональне перетворення координат, внаслідок якого кореляції між окремими координатами перетворяться в нуль.

1.6.2 T-SNE

Стохастичне вкладення сусідів з t-розподілом (T-distributed Stochastic Neighbor Embedding) - це алгоритм[12] машинного навчання для візуалізації, розроблений Лоренсом ван дер Маатеном і Джеффрі Хінтон. Він є технікою нелінійного зниження розмірності, добре підходить для вкладення даних високої розмірності для візуалізації в простір низької розмірності (дво- або тривимірне). Зокрема, метод моделює кожен об'єкт високої розмірності дво- або тривимірній точкою таким чином, що схожі об'єкти моделюються близько розташованими точками, а несхожі точки моделюються з великою ймовірністю точками, далеко один від одного віддаленими.

Алгоритм t-SNE складається з двох основних етапів. Спочатку t-SNE створює розподіл ймовірностей по парам об'єктів високої розмірності таким чином, що схожі об'єкти будуть обрані з великою ймовірністю, в той час як ймовірність вибору несхожих точок буде мала. Потім t-SNE визначає схоже розподіл ймовірностей по точкам в просторі малої розмірності і мінімізує відстань Кульбака - Лейблера між двома розподілами з урахуванням положення точок. Зауважимо, що вихідний алгоритм використовує евклідова відстань між об'єктами як базу вимірювання подібності, це може бути змінено відповідно до обставин.

Алгоритм t-SNE використовувався для візуалізації широкого ряду додатків, включаючи дослідження комп'ютерної безпеки, музичний аналіз, дослідження по раку біоінформатику і обробку біомедичних сигналів. Алгоритм часто використовується для візуалізації високорівневих уявлень, отриманих зі штучної нейронної мережі.

Оскільки t-SNE відображення часто використовуються для показу кластерів, а на візуалізацію кластерів може мати значний вплив обрана параметризація, остільки необхідно вміння працювати з параметрами алгоритму t-SNE. Для вибору параметрів і перевірки результатів можуть виявитися необхідні інтерактивні [невідомий термін] дослідження. Було продемонстровано, що алгоритм t-SNE часто здатний виявити добре відокремлені один від одного кластери, а при спеціальному виборі параметрів апроксимувати простий вид спектральної кластеризації.

1.7 Обробка природної мови

Обробка природної мови (NLP) - це область досліджень для програмних застосунків, яка вивчає, як комп’ютери можуть бути використані для розуміння та маніпулювання текстом або природною мовою, щоб знаходити з текстових даних деяку корисну інформацію. Дослідники NLP прагнуть зібрати знання про те, як люди розуміють та використовують мову, щоб можна було розробити відповідні інструменти та техніки для розуміння мови комп'ютерними системами та їх навчання маніпулювати даними для виконання бажаних завдань. Основи NLP лежать у низці дисциплін, а саме: комп’ютерні та інформаційні науки, лінгвістика, математика, електротехніка та електронна інженерія, штучний інтелект та робототехніка, психологія тощо. Застосування NLP включає низку областей досліджень, таких як машинний переклад, обробка та узагальнення тексту на природній мові, користувальницькі інтерфейси, багатомовність та перехресний пошук мовної інформації, розпізнавання мови, штучний інтелект та експертні системи.[13]

Одна з важливих областей застосування обробки природної мови, яка є відносно новою і стала досить помітною через поширення всесвітньої павутини та цифрових бібліотек. Вчені вказували на необхідність проведення відповідних досліджень для сприяння пошуку багатомовних чи міжмовних відомостей, включаючи багатомовні системи обробки тексту та багатомовні системи

користувальницького інтерфейсу, з метою повноцінного використання переваг цифрових бібліотек.[14]

Маніпуляція з текстами з метою вилучення деяких знань, для автоматичного індексування та реферування або для перетворення тексту у бажаному форматі визнана важливою сферою досліджень у NLP. Це широко класифікується як область обробки тексту на природній мові, що дозволяє структурувати великі об’єми текстової інформації з метою отримання корисної інформації чи структури знань, яка може бути використана для поставленої мети. Системи автоматичної обробки тексту, як правило, приймають певну форму введення тексту і перетворюють його на вихід в іншій формі. Центральним завданням для систем обробки тексту на природній мові є переклад потенційно неоднозначних запитів та текстів природною мовою у однозначні внутрішні представлення, за якими можна виконувати пошук. Система обробки тексту на природній мові розпочинається з морфологічного аналізу. Підбір термінів як у запитах, так і в документах, робиться для того, щоб отримати морфологічні варіанти слів, що беруть участь. Лексична та синтаксична обробка передбачає використання лексиконів для визначення характеристик слів, розпізнавання їх частин мови, а також для синтаксичного аналізу речень. Деякі NLP системи побудовані для обробки текстів з використанням окремих невеликих підмов, щоб зменшити розмір операцій та характер складностей.

Проблема розуміння природної мови комп’ютером ховається в її неоднозначності. Типи можливих неоднозначностей: синтаксична, смислова, відмінкова, референційна.

1.8 Word embedding

Використання слів в якості підготовлених ознак в задачах обробки природної мови - інтуїтивне і природне рішення, тому що слова є основними значущими одиницями для природних мов. І не дивно, що поданням слів у вигляді числових векторів для подальшого застосування методів машинного навчання присвячена

велика кількість досліджень. Числове уявлення слів позначається терміном word embedding [15] і випливає з ідеї «вкладення» додаткової інформації про слово в векторне подання. На рисунку 1.6 представлений примітивний варіант векторного уявлення слів за допомогою one-hot кодування. [16]

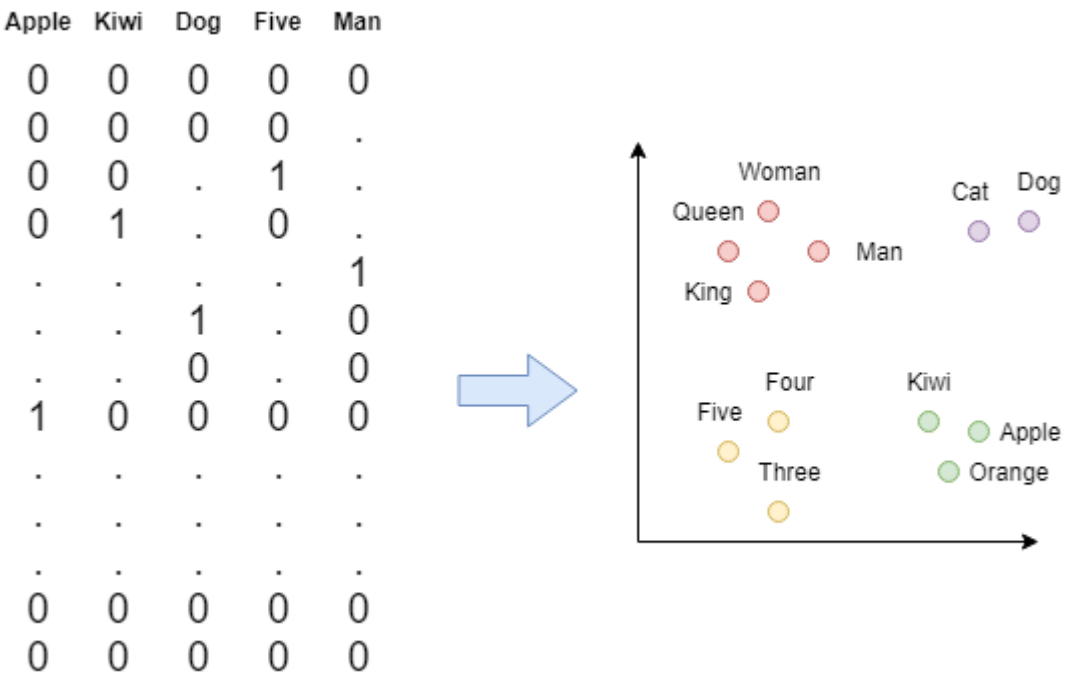


Рисунок 1.6 – Приклад one-hot кодування

Історія word embedding починається близько 30 років назад. Перші експерименти з ними проводили Хінтон і МакКлелланд в 1990 році, Хінтон і Румельзарт в 1985 і Ельман в 1990. З тих пір створення уявлень слів встановилося в двох родинах моделей: глобальних матричних методах факторизації і методах локального контекстного вікна. Глобальна матрична факторизація працює з глобальним рівнем корпусу і заснована на підрахунку частоти вживання слів в корпусі, вона добре представляє статистичні характеристики слів.[17]

Методи локального вікна працюють на більш детальному рівні корпусу, в межах невеликого вікна слів, розташованих поруч. такі методи успішно застосовуються в рішенні задач пошуку аналогічних слів, але вони не можуть відображати статистичні характеристики. методи локального вікна створюють

векторне простір для слів виходячи з такого припущення, що семантично близькі слова розташовуються близько один до одного. Семантичне уявлення пояснює їх ефективну застосовність в задачах пошуку аналогій.

Всі методи векторного уявлення слів погано узагальнюються на морфологічно багаті мови. Швидше за все, це відбувається через проблеми «Словникового вибуху». Такала [18] у своїй роботі пропонує метод векторного уявлення слів для морфологічно багатого мови, який заснований на розбитті слова на частини: основу і закінчення. Такала виявив, що word embedding на рівні частини слова прості в реалізації і можуть перевершити методи word embedding на рівні цілого слова в рішенні декількох завдань обробки природного мови. Але в його роботі не проводилося порівняння відмінностей підходу з поділом основи і закінчення слова від підходу з використанням слова цілком.

Одна з проблем, що виникають у зв'язку з використанням методів word embedding для задач обробки природної мови в високо морфологічних мовами, полягає в тому, що навіть якщо word embedding словника великий, деякі слова, як і раніше будуть в ньому відсутні. Цю проблему можна частково вирішити за допомогою додавання етапу лематизації слів перед пошуком словника векторних уявлень. Лематизації, виконана з достатньою точністю, може повністю усунути дану проблему для деяких морфологічно багатих мов, але існують і такі мови, в яких структура слів занадто складна. Обробка таких слів, що не увійшли в словник, викликає труднощі, оскільки уявлення слів не може бути апроксимувати з письмової форми. Письмова форма слова містить лише малу частина семантичного значення слова, тому методи локальних вікон обробляють семантичну інформацію виключно з контексту, в якому з'являється слово.

1.9 Модель Word2Vec

Word2Vec - це техніка для обробки природної мови. Алгоритм word2vec [19] використовує модель нейронної мережі для вивчення асоціацій слів із великого корпусу тексту. Після навчання така модель може виявляти слова синоніми або пропонувати додаткові слова для часткового речення. Як випливає з назви,

word2vec зіставляє кожне окреме слово з певним списком чисел, який називається вектором. Вектори вибираються ретельно таким чином, щоб проста математична функція (косинусова подібність між векторами) вказувала на рівень семантичної подібності між словами, представленими цими векторами.

Даний алгоритм реалізує дві основні архітектури – Continuous Bag of Words (CBOW) і Skip-gram.(Рисунок 1.7) Обидві архітектури володіють власними перевагами. наприклад, підхід skip-gram найбільш ефективний при обробці невеликого корпусу навчальних даних. Більш того, за допомогою нього добре описуються слова, що рідко зустрічаються. З іншого боку, CBOW працює швидше і краще обробляє часто вживані слова. Однак навчання вихідного вектора алгоритмів CBOW і skip-gram є одним з найбільш великих обмежень цих моделей, так як це може бути важким завданням, що вимагає великих обчислювальних витрат. Для вирішення цієї проблеми можна використовувати два алгоритми.

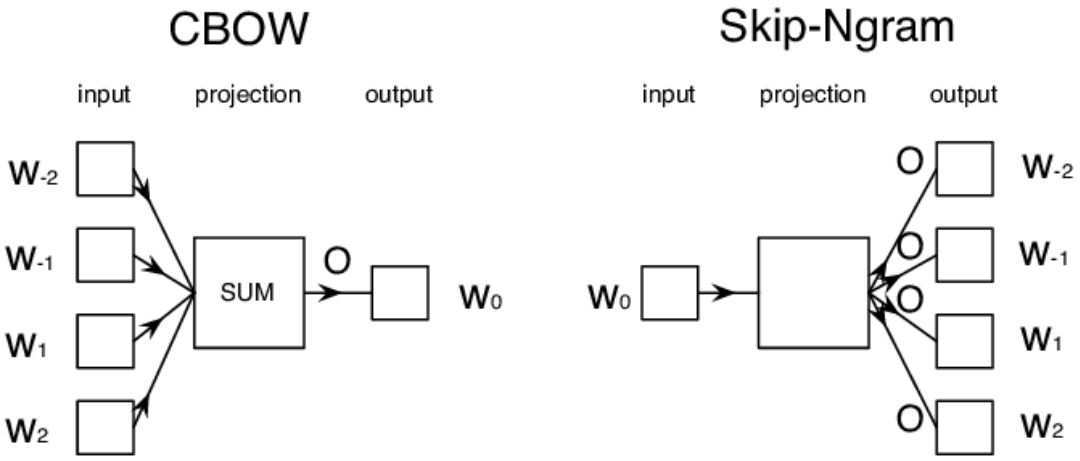


Рисунок 1.7 – Архітектури для реалізації моделі

Перший - негативний семплінг. Основна ідея алгоритму полягає в обмеженні кількості вихідних векторів, які потрібно оновлювати. Таким чином, тільки деякі вектори оновлюються випадковим чином. Цей розподіл шуму є імовірнісним і використовується в процесі семплінгу.

Другий алгоритм - ієрархічний softmax. Він заснований на дереві Хаффмана. Фактично, це двійкове дерево, яке представляє всі терміни на основі їх частоти появи в корпусі тексту. Потім кожен крок від кореня до мети нормалізується.

Кожен алгоритм має переваги в порівняно з іншим в залежності від навчальних даних. Наприклад, негативний семплінг більш ефективно працює з векторами малої розмірності і часто вживаними словами. Проте, ієрархічний softmax показує себе краще в роботі з рідко вживаними словами.

1.9.1 Модель CBOW

Continuous Bag of Words (CBOW) дозволяє передбачати поточне слово, грунтуючись на його контексті, який визначається сусідніми словами у вікні. У CBOW використовуються три шари. Вхідний шар відповідає контексту. Прихований шар - проекції кожного слова з вхідного шару в вагову матрицю, яка проектується в третій вихідний рівень. Останнім етапом моделі є порівняння її виведення з самим словом, щоб скорегувати його уявлення, засноване на зворотному поширенні градієнта помилки. Таким чином, метою нейронної мережі CBOW є максимізація рівняння показаного формулою 1.1.

$$\frac{1}{V}\sum_{t=1}^{V}\log p\left(m_t\,|m_{t-\frac{c}{2}}\dots m_{t+\frac{c}{2}}\right), \qquad (1.1)$$

де V - об'єм словника, c - розмір вікна для кожного слова.

1.9.2 Skip-gram

Метод skip-gram вирішує обернену задачу: на підставі одного слова передбачається контекст. Останній крок алгоритму - порівняння виведення з кожним словом в контексті з метою коригування уявлення, заснованого на зворотному поширенні градієнта помилки. даний метод виконує максимізацію наступного рівняння(Формула 1.2).

$$\frac{1}{V}\sum_{t=1}^{V}\sum_{j=t-c,j\neq 1}^{t+c}\log p(m_j|m_t), \qquad (1.2)$$

де V - об'єм словника, c - розмір вікна для кожного слова.

1.10 Методи класифікації даних

Задача класифікації полягає в тому щоб зіставити кожному об’єкту з множини об’єктів клас до якого він відноситься. Спочатку задається вибірка даних – множина об’єктів для яких відомо якому класу вони належать. Класова

належність всіх інших об’єктів є невідомою. Необхідно побудувати алгоритм, який може класифікувати довільний об’єкт з множини.

У математичній статистиці завдання класифікації називаються також завданнями дискримінантного аналізу. У машинному навчанні завдання класифікації вирішується різними методами, зокрема, за допомогою методів штучних нейронних мереж та ансамблевих методів при постановці експерименту у вигляді навчання з учителем.

1.10.1 Метод Random forest

Random forest - це ансамблевий метод[20] навчання для класифікації та регресії який виконуються шляхом побудови множини дерев рішень під час навчання та виведення класу, до якого належить об’єкт при класифікації або середнього значення для прогнозу окремих дерев при регресії. Випадкові ліси допомагають перевизначати навчальний набір даних для дерев рішень. Випадкові ліси, як правило, дають кращий результат ніж дерева рішень. Однак є різні характеристики, що можуть впливати на їх ефективність.

Дерева рішень доволі популярний метод для різних завдань машинного навчання. Зокрема, дерева, які розгортаються дуже глибоко, схильні до перенавчання, засвоюючи не потрібні зв’язки: вони перевантажують свої навчальні набори, тобто мають низьке відхилення, але дуже велику дисперсію. Випадкові ліси - це спосіб усереднення певної сукупності дерев рішень, що навчаються на різних частинах одного і того ж навчального набору, з метою зменшення дисперсії. Це відбувається за рахунок невеликого збільшення упередженості та певної втрати в зрозумілості, але, як правило, це значно підвищує ефективність роботи в кінцевій моделі. Ліси - це як об’єднання зусиль алгоритму дерева рішень. Використовуючи командну роботу багатьох дерев, тим самим покращуючи роботу одного випадкового дерева. Принцип роботи випадкового лісу спостерігаємо на рисунку 1.8.

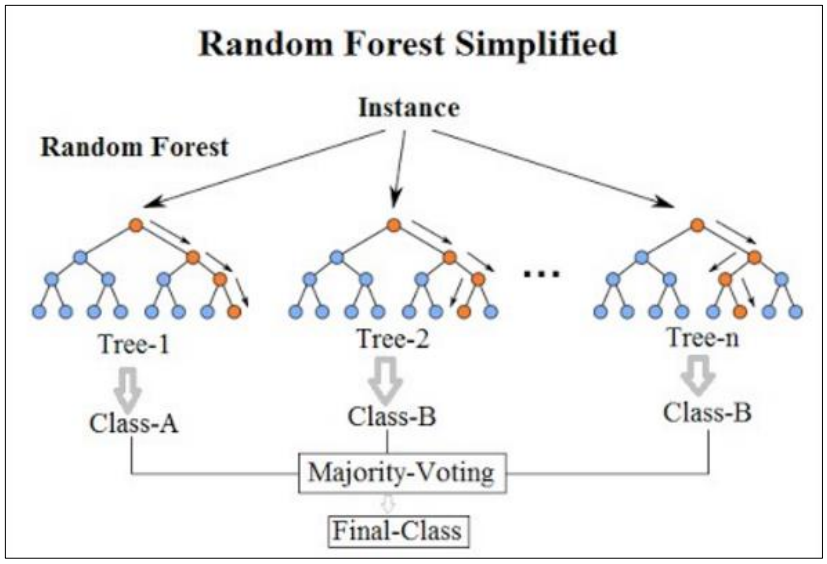


Рисунок 1.8 – Класифікація методом випадкового лісу

1.11 Постановка завдання

В рамках даного дослідження мають бути вирішені наступні завдання:

- вибір набору даних для обробки;
- розбиття ЕКГ-сигналу на послідовність серцебиття;
- виділення хвиль для кожного серцебиття;
- кластеризація виділених хвиль та створення словника;
- переведення вхідного сигналу ЕКГ до набору символів, де кожний символ відповідає певній хвилі (частині такту ЕКГ);
- створення слів та речень на основі створеного словника;
- побудова моделі Word2Vec на основі створених слів;
- застосування моделі для аналізу даних методами NLP.

Висновки до розділу

У розділі розглянуто поняття ЕКГ сигналу та його складових, методи виявлення R-піків в електрокардіограмі. В методі Пана-Томпкінса спочатку для вхідного сигналу ЕКГ застосовують серію фільтрів, щоб прибрати зайвий шум і тільки після цього починається пошук R-піків з подальшим виявленням QRS комплексу хвиль.

Також розглянуто поняття кластерного аналізу, алгоритми кластеризації та методи пониження розмірності даних. Серед алгоритмів кластеризації можемо

виділити ієрархічну кластеризацію, яка має два різні підходи для побудови ієрархічного дерева: агломеративний та розділювальний. Це дозволяє ефективно кластеризувати дані різної структури.

Також було описано методи природної обробки мови, поняття векторного представлення слів і алгоритми для створення Word2Vec моделі. Розписано завдання, які мають бути вирішенні в рамках дослідження.

2 ЗАСТОСУВАННЯ WORD2VEC МОДЕЛІ ДЛЯ АНАЛІЗУ ЕКГ

2.1 Переведення ЕКГ сигналу в набір символів

Перейдемо до нового напрямку дослідження для аналізу сигналів ЕКГ, представивши нову структуру, яка називається ECG language processing (ELP)[21] яка обробляє сигнал ЕКГ подібно до обробки природною мовою текстового документа. Запропонована структура застосовується до різних біомедичних застосувань, а також може покращити ефективність роботи неглибоких алгоритмів машинного навчання. Мова складається з кінцевого / нескінченного набору речень, що складають слова. Подібно до природних мов, сигнал ЕКГ складається з послідовностей трьох або чотирьох різних хвиль, включаючи хвилю P, комплекс хвиль QRS, хвилю T та хвилю U (Рисунок 2.1). Кожна нормальна ЕКГ включає різні різновиди кожної хвилі.

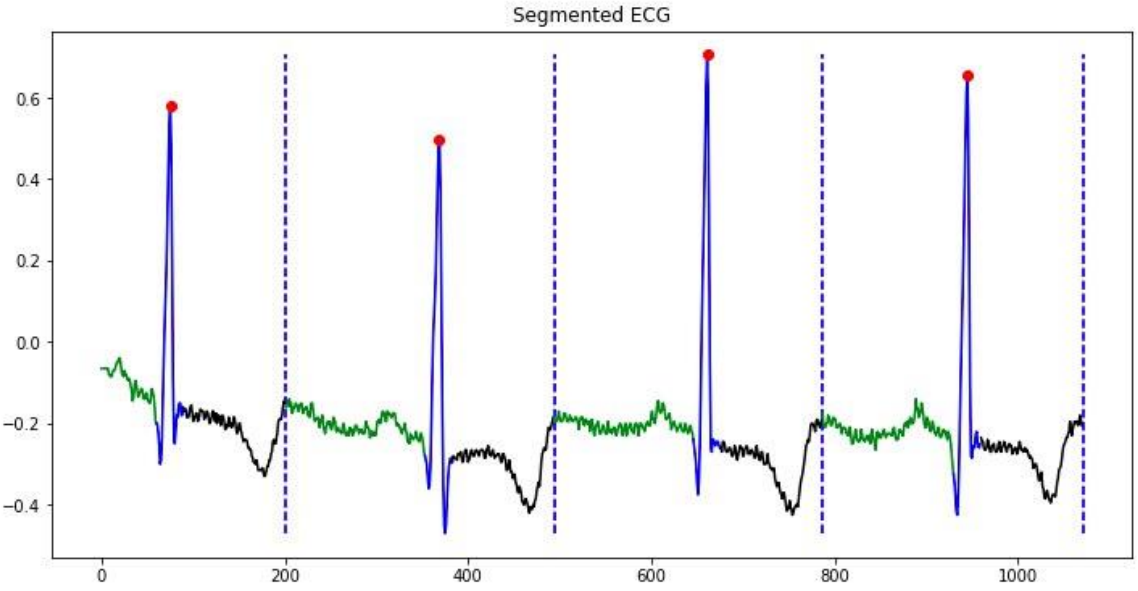


Рисунок 2.1 – Послідовність хвиль в ЕКГ

Наприклад, комплекс хвиль QRS може мати різну форму, як показано на рисунку 2.2. Отже, сигнал ЕКГ є послідовністю серцебиттів (як речення в природній мові), і кожне серцебиття складається з множини хвиль (як слова в реченні) різної морфології. Аналогічно NLP, який використовується для того, щоб допомогти комп’ютерам / машинам зрозуміти та інтерпретувати природну мову людини, запропонована нами обробка мови ЕКГ на основі НЛП може допомогти комп’ютерам глибше зрозуміти сигнали електрокардіограми.

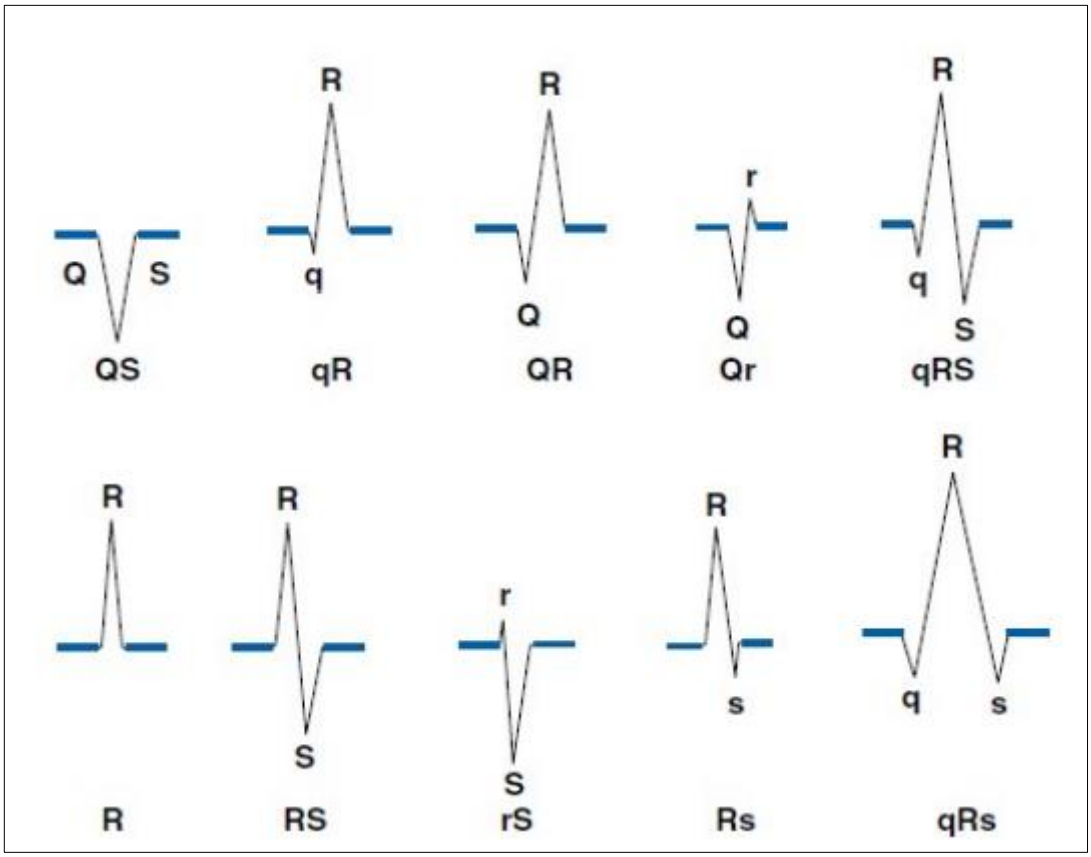


Рисунок 2.2 – Різні форми комплексу QRS

2.2 Етапи обробки мови ЕКГ

2.2.1 Виявлення R-піків

Етап включає виявлення R-піків поточного сигналу ЕКГ або виявлення комплексів хвиль QRS. Для цієї мети використовуються один з алгоритмів Пана – Томпкінса. Червоними колами на рисунку 2.1 зображені R-зубці ЕКГ-сигналу.

2.2.2 Розбиття ЕКГ сигналу

Етап включає поділ безперервного ЕКГ-сигналу на певну послідовність серцебиттів та розбиття кожного серцебиття на окремі одиниці, які називаються хвилями. Після виявлення R-хвиль, наявність інших складових хвиль (тобто P, QRS і хвилі T) в ЕКГ-сигналі можуть бути вилучені за допомогою адаптивних вікон пошуку. Для виконання сегментації сигналу на серцебиття ідентифікуємо один сегмент як фіксовану кількість екземплярів зняття сигналу до розташування піку R та до фіксованої кількості екземплярів після розташування піку R або від початку хвилі P до зміщення послідовної хвилі T. На малюнку рисунку 2.2 зображена сегментований ЕКГ сигнал, розфарбований R-хвилями, P, QRS і T-хвилями.

2.2.3 Створення словника хвиль

Етап включає побудову словника хвиль на основі вилучених хвиль з ЕКГ сигналів. Шляхом групування хвиль формуємо середнє значення кожної групи як вхід до словника. Це може бути здійснено шляхом подачі всіх хвиль на вхід алгоритмів кластеризації, таких як K-середніх, спектральна кластеризація або агломеративні алгоритми кластеризації[. Після кластеризації хвиль середнє значення кожного кластера може представляти окрему хвилю словникового запасу. На рисунку 2.4 зображено кластеризацію набору даних ЕКГ-сигналу з використанням техніки t-розподіленого стохастичного сусідства (t-SNE).

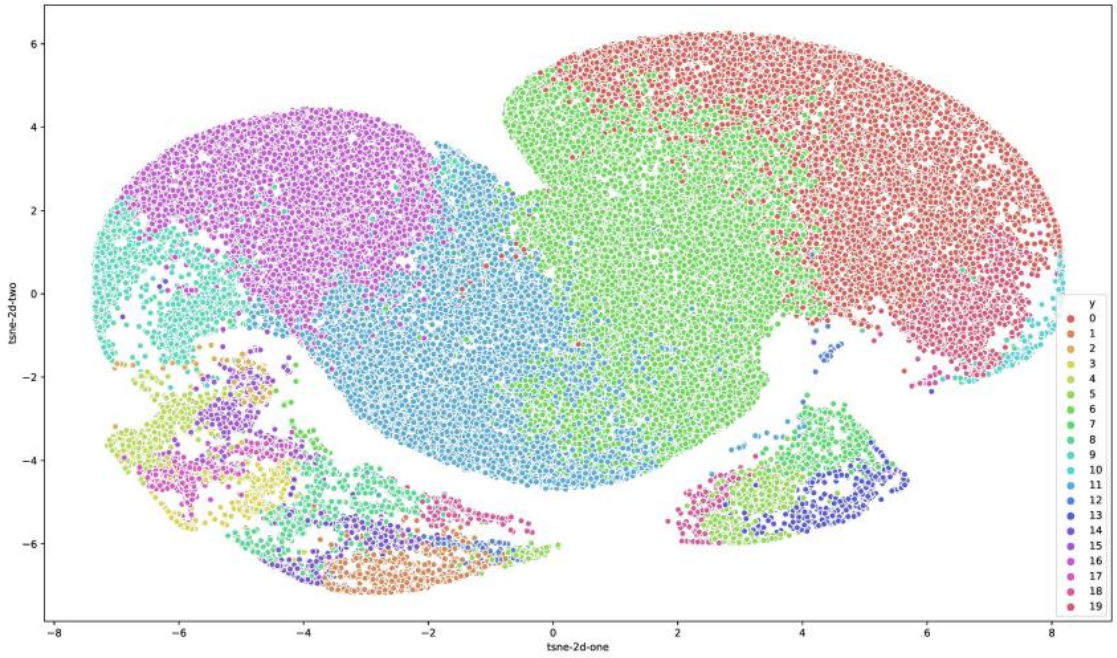


Рисунок 2.4 – Результат кластеризації хвиль методом t-SNE

2.2.4 Сегментація хвиль

Процес сегментації розбитих хвиль створює послідовність хвиль для кожного сигналу ЕКГ. Потім, кластер кожної хвилі в послідовності ідентифікується за допомогою виходу попереднього кроку(тобто попереднього етапу конвеєра).Іншими словами, він присвоює унікальний символ (який відповідає певному кластеру) кожній хвилі в послідовності. Таким чином, кожен ЕКГ сигнал кодується послідовністю символів так, що кожний символ представляє певну хвилю (або кластер) у словниковому запасі.

2.2.5 Векторизація хвиль

На цьому етапі приймаємо кодований словник виділенних хвиль і будуємо вектор вбудовування (тобто вектор заданої довжини) для кожної хвилі словникового запасу. Основна причина вбудовування слів полягає в тому, що це дозволяє нам застосовувати вдосконалену модель, що спирається як на штучні нейронні мережі так і на цілочисельні кодовані сигнали ЕКГ для конкретного завдання. За допомогою NLP можемо використовувати кілька підходів, такі як граф векторизатор, у якому послідовність хвиль перетворюється у вектор фіксованої довжини із розміром словникового запасу. Значення в кожному положенні у векторі буде підрахунком кожної хвилі в закодованому сигналі, або підхід Word2Vec, який використовує нейромережеві методи для представляють хвилі у векторному просторі. Останній підхід є більш ефективним, оскільки він розпізнає контекст, відношення та подібність між хвилями.

2.2.6 Тренування моделі

Етап передбачає використання методів машинного навчання та глибокого навчання для створених моделей, які зможуть виконувати будь-які навчальні завдання, включаючи класифікацію, прогнозування тощо.

2.3 Створення Word2Vec моделі

Після перетворення всього сигналу ЕКГ на речення можемо перейти до створення та тренування Word2Vec моделі. Використаємо skip-gram архітектуру за основу нашої моделі, яка на відміну від CBOW розглядає центральне слово з вінка та передбачає контекстні слова.

Модель Skip-gram приймає корпус тексту і створює one-hot вектор для кожного слова. One-hot вектор - це векторне представлення слова, де вектор має розмір словника (загальна кількість унікальних слів). Для всіх вимірів встановлено значення 0, крім виміру, що представляє слово, яке використовується як вхідний сигнал у цей момент часу. В нашому випадку one-hot вектор буде представляти одне серцебиття з ЕКГ. Приклад one-hot вектора зображено на рисунку 2.5.

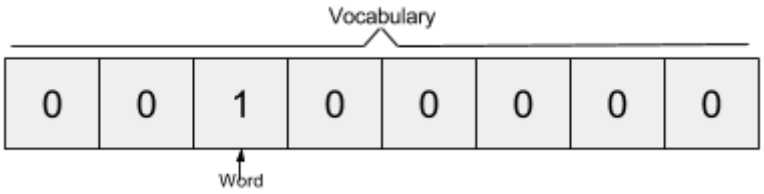


Рисунок 2.5 - one-hot вектор

Векторне представлення всіх слів ми подаємо на вхід до згорткової нейромережі з одним прихованим шаром. Результатом роботи нейромережі є єдиний вектор (такого ж розміру як і one-hot вектор), що містить для кожного слова в нашому словнику ймовірність того, що випадковим чином вибране поруч слово є цим словом із словника.

У word2vec використовується розподілене представлення слова. Візьмемо для прикладу вектор із кількома сотнями вимірів (скажімо, 1000). Кожне слово буде представлене розподілом ваги між цими елементами. Отже, замість однозначного відображення між елементом у векторі та словом, подання слова розподіляється по всіх елементах у векторі, і кожен елемент у векторі сприяє визначенню багатьох слів. Такі вектори допомагають, якимось абстрактним чином чисельно представити значення слова. Ми бачимо, що просто дослідивши великий корпус, можна вивчити вектори слів, які здатні фіксувати взаємозв’язки між словами на диво точно.

Вихідним шаром в нейромережі буде друга матриця ваги, яку можна використовувати для обчислення оцінки для кожного слова в словниковому запасі, а softmax - для отримання попереднього розподілу слів. Модель skip-gram є протилежністю моделі CBOW. Вона будується за рахунок слова, яке є єдиним вхідним вектором, і цільові контекстні слова тепер знаходяться на вихідному рівні. Функція активації прихованого шару просто зводиться до копіювання відповідного рядка з матриці ваг. На вихідному рівні ми тепер виводимо n багаточленних розподілів замість одного. Метою навчання є мінімізація сумарної помилки передбачення для всіх контекстних слів вихідного рівня. Приклад архітектури зображено на рисунку 2.6.

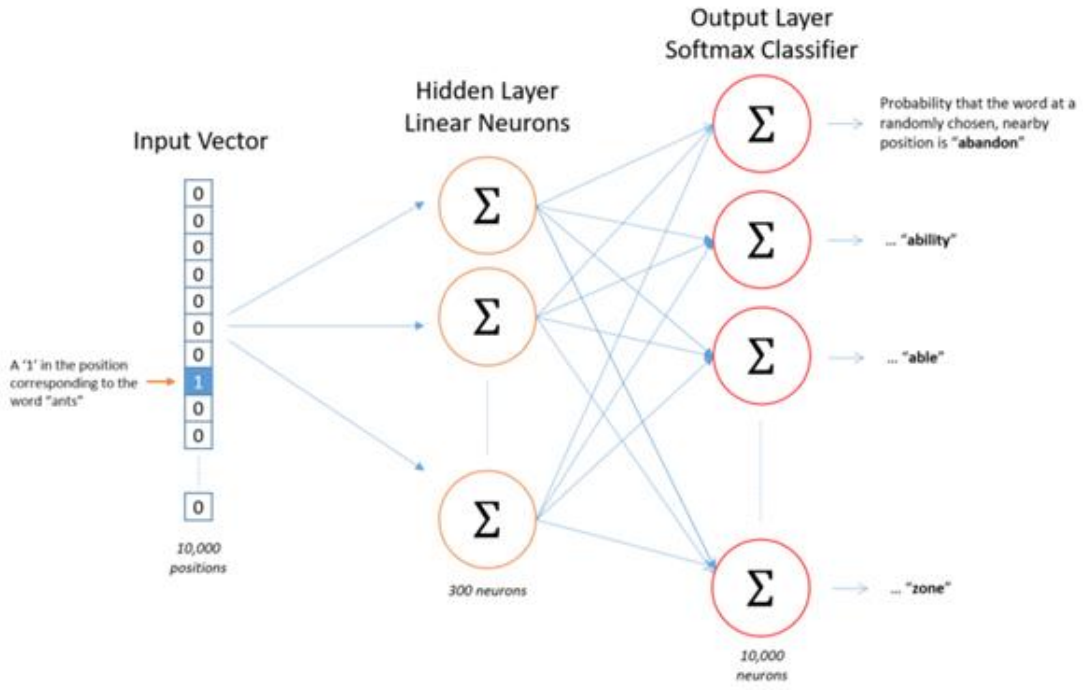


Рисунок 2.6 – Архітектура мережі для skip-gram моделі

Кінцевою метою всього цього насправді є лише вивчення прихованої матриці ваги вихідного шару, який ми просто передаємо на вхід іншій мережі. Вихідний рівень мережі - це класифікатор регресії softmax. Зокрема, кожен вихідний нейрон має вектор ваги, який він множить на вектор слова із прихованого шару, а потім він застосовує функцію експоненти до результату. Нарешті, для того, щоб отримати результати, ми ділимо цей результат на суму результатів з усіх вихідних вузлів. Якщо два різні слова мають дуже схожі контексти, то наша модель повинна вивести дуже подібні результати для цих двох слів. І один із способів для мережі вивести подібні передбачення контексту для цих двох слів – це вивести подібні вектори для цих слів. Отже, якщо два слова мають подібний контекст, то наша мережа мотивована вивчати подібні вектори слів для цих двох слів.

Після навчання нейромережі на всьому корпусі серцебиттів ЕКГ, ми отримуємо готову word2vec модель з лінгвістичним ланцюгом представлення зв’язку серцебиття у вигляді слова та відповідного йому векторного представлення. Приклад лінгвістичного ланцюга спостерігаємо на рисунках 2.7, 2.8. За допомогою створеної моделі можна виявляти подібність між серцебиттями,

які ми хочемо проаналізувати та з серцебиттями, які промаркеровані деякими мітками(аритмія, захворювання, без аномалій).

```
'kwb': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99550>,
'kwc': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88e48>,
'kwd': <gensim.models.keyedvectors.Vocab at 0x7f55d8ea1da0>,
'kwe': <gensim.models.keyedvectors.Vocab at 0x7f55d8eab438>,
'kwf': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95160>,
'kwg': <gensim.models.keyedvectors.Vocab at 0x7f55d8eab358>,
'kwi': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95f98>,
'kwj': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99588>,
'kwk': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88f98>,
'kwl': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95208>,
'kwm': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99f60>,
'kwn': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99fd0>,
'kxa': <gensim.models.keyedvectors.Vocab at 0x7f55d8e91d68>,
'kxb': <gensim.models.keyedvectors.Vocab at 0x7f55d8e995f8>,
'kxc': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95ac8>,
'kxe': <gensim.models.keyedvectors.Vocab at 0x7f55d8ea12e8>,
'kxi': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95278>,
'kxj': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99e10>,
'kxk': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88da0>,
'kxm': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99f28>,
'kxn': <gensim.models.keyedvectors.Vocab at 0x7f55d8ea1748>,
'lqa': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95a90>,
'lqb': <gensim.models.keyedvectors.Vocab at 0x7f55d8ea1710>,
'lqc': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88cf8>,
'lqe': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99b00>,
'lqf': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95400>,
'lqi': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88f60>,
```

Рисунок 2.7 – Приклад лінгвістичного ланцюга

```
kwb : [-4.52855631e-04  2.69200653e-03 -2.85770698e-03 -3.71513655e-03
  7.24595389e-04 -6.08999457e-04  2.23849644e-03 -3.87491775e-03
 -1.76306057e-03  6.52205083e-04  7.47220067e-04 -2.37480062e-03
  2.80323718e-03  1.15808658e-03 -1.81753351e-03 -2.80798832e-03
 -5.11458609e-03  3.07700131e-03  2.54009705e-04  4.26528323e-03
 -1.63862924e-03  4.70669661e-03  3.97165120e-03  3.91787663e-03
 -3.11121764e-03  3.04562040e-03 -2.35919980e-03  1.19900316e-04
 -5.36466995e-03  2.48389272e-03 -1.44891499e-03 -7.83059513e-04
  3.53254564e-03 -2.12145061e-03 -1.40378485e-03  3.31551279e-03
  4.04210994e-03  1.01519178e-03  1.57758989e-03  4.68220329e-03
  1.86871411e-03 -6.41508261e-04  2.16125301e-03  1.22731004e-03
 -1.09590031e-03 -6.07335009e-03  1.30223471e-03 -3.83058796e-03
  2.22818181e-03 -4.63103503e-03 -1.52695086e-03  3.62753542e-03
 -2.10220553e-03  4.25343588e-03  2.28840276e-03  3.60550778e-03
 -5.99312079e-05 -3.19302292e-03 -4.41611465e-03  5.54119237e-04
 -5.55756828e-03  4.10656631e-03  2.68298457e-03  4.14208882e-03
 -8.60912609e-04  5.34155697e-04  3.25926510e-03  2.76880083e-03
  1.42918539e-03 -3.18572158e-03 -2.72700260e-03 -2.76432396e-03
  2.55474891e-03  2.83807237e-03 -8.61959648e-04  4.05401969e-03
  3.96387372e-03 -2.75863800e-03 -4.49991878e-03 -1.00883329e-03
  4.77602240e-03  2.75431201e-03 -1.71090907e-03 -1.92280975e-03
  4.18488542e-03  6.76358759e-04 -7.16106326e-04  1.82076625e-03
  3.56791681e-03 -1.83787569e-03 -8.79865547e-05  1.55338924e-03
 -2.37935386e-03 -4.98656929e-03 -4.83540772e-03 -3.00369668e-03
  1.35593917e-04  3.89454677e-03  3.50713078e-03 -5.40652266e-03]
```

Рисунок 2.8 – Векторне представлення окремого серцебиття

Висновки до розділу

У даному розділі було детально описано підхід представлення структури для обробки електрокардіограми подібно до обробки природною мовою текстового документа, описано етапи попередньої обробки ЕКГ сигналу, перетворення кожного серцебиття в слово, а всього ЕКГ-сигналу в послідовність речень. Так для початку потрібно виявити R-піки, потім розбити сигнал на окремі серцебиття та виділити різні типи хвиль для кожного серцебиття. Після цього відбувається створення словника хвиль, кластеризація хвиль та присвоєння кожній з груп словника окремого символа. Далі кожна хвиля замінюється символом кластера, якому вона належить та відбувається перетворення електрокардіограми на послідовність слів.

Розглянуто процес створення та тренування Word2Vec моделі. Для цього обирається один з двох алгоритмів побудови моделі: CBOW або skip-gram. Далі вхідний текст розподіляється на набір даних для тренування нейромережі за допомогою центрального вікна. І в результаті навчання мережі отримується готова для роботи Word2Vec модель.

3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Програмне забезпечення виконано на інтерпретованій мові програмування високого рівня Python[22]. Python використовує строгу динамічну типізацію. Мова була розроблена у 1990 році нідерландським програмістом Гвідо ван Россумом. Python підтримує об’єктно орієнтований, процедурний та функціональні підходи до програмування, що дозволяє розробляти різноманітне програмне забезпечення для різних сфер діяльності.

Серед основних її переваг можна назвати такі:

- відкритий код (можливість редагувати його іншими користувачами);

- наявність діалогового режиму (корисно для проведення експериментів та вирішення задач);

- переносність програм (що властиве більшості інтерпретованих мов);

- чистий синтаксис;

- зручний для розв'язання математичних проблем.

Для зчитування сигналу ЕКГ з бази даних була використана бібліотека wfdb, яка містить інструменти для читання, запису, редагування ЕКГ. Основні компоненти цього пакету базуються на оригінальних специфікаціях роботи з базою даних для зберігання сигналів.

Для попередньої обробки сигналу ЕКГ та маніпулювання даними був застосований модуль numpy. Numpy це бібліотека [23] з відкритим кодом для мови програмування Python, що надає функціонал для виконання математичних та числових операцій. Функції цієї бібліотеки об’єднані в пакети високого рівня, вони забезпечують розробника функціоналом подібним до MatLab. Numpy надає базові методи для маніпуляції з великими масивами і матрицями. Scipy розширює функціонал бібліотеки величезною колекцією корисних алгоритмів, таких як мінімізація, перетворення Фур'є, регресія, і інші прикладні математичні техніки.

Головною особливістю numpy є об'єкт array, який є схожим зі звичайними списками Python окрім того, що елементи масиву повинні мати однаковий тип

даних (float, int). Однак з масивами можна проводити числові та математичні операції зі значним обсягом інформації в рази швидше та ефективніше ніж зі списками.

Для візуалізації ЕКГ та результатів кластеризації хвиль ми використали бібліотеку matplotlib. Бібліотека matplotlib - це бібліотека[24] двовимірної графіки для мови програмування python за допомогою якої можна створювати високоякісні малюнки різних форматів. Matplotlib реалізує широкий функціонал для візуалізації різноманітних даних. Функції та класи бібліотеки ієрархічно пов'язані між собою тому об’єднуються в множину модулів.

Створення малюнка в matplotlib схоже з малюванням в реальному житті. Так художнику потрібно взяти основу (полотно або папір), інструменти (кисті або олівці), мати уявлення про майбутнє малюнку і, нарешті, виконати все це і намалювати малюнок.

У matplotlib всі ці етапи також існують, і в якості художника-виконавця тут виступає сама бібліотека. Від користувача потрібно управляти діями художника-matplotlib, визначаючи що саме він повинен намалювати і якими інструментами. Зазвичай створення основи і процес відображення малюнка віддає повністю на відкуп matplotlib. Таким чином, користувач бібліотеки matplotlib виступає в ролі управлінця. І чим простіше йому управляти кінцевим результатом роботи matplotlib, тим краще.

Пакет забезпечує можливість візуалізувати різні типи графіків та діаграм:

- гістограми;
- стовпчасті діаграми;
- секторні діаграми;
- графіки функцій;
- точкові графіки;
- поля градієнтів;
- діаграми розсіювання;
- контурні графіки.

Для проведення кластеризації виділених хвиль з серцебиття та для класифікації серцебиття була використана бібліотека для роботи з алгоритмами машинного навчання scikit-learn. Scikit-learn [25] – це бібліотека з відкритим кодом написана на мові програмування Python, що використовується для швидкої реалізації та ефективного виконання задач машинного навчання. Її функціонал дозволяє вирішувати різноманітні задачі за рахунок підтримки алгоритмів навчання з вчителем (використовуються переважно для регресії, класифікації) та навчання без учителя (для кластеризації). Навчання з учителем використовується якщо є набір даних з об'єктами та їх мітками належності об'єкта до певного класу. Натомість при навчанні без вчителя об'єкти вибірки не мають маркерів і задача алгоритму знайти корисну інформацію про їх взаємозв'язки. Бібліотека використовує ряд інших бібліотек для математичних обрахунків, що відкриває можливість самостійно розширювати функціонал. Докладно розписана документація та підтримка бібліотеки іншими розробниками роблять scikit-learn однією із найбільш застосовуваних інструментів мови Python для вирішення задач машинного навчання. Також бібліотека застосовується для промислових систем, в яких потрібна реалізація алгоритмів машинного навчання, для досліджень, а так само для новачків, які тільки робить перші кроки в області машинного навчання.

Бібліотека підтримує такі алгоритми та моделі:

- алгоритми регресії та класифікації даних;
- алгоритми кластеризації даних: k-means, ієрархічна кластеризація;
- нейроні мережі;
- методи пониження розмірності: метод головних компонент, стохастичне вкладення сусідів з t-розподілом;
- алгоритми підбору гіперпараметрів моделі: пошук по сітці (навчання моделі використовуючи всі можливі комбінації зі створеної сітки параметрів);
- дерева рішень;
- сингулярний розклад матриці;

- ансамблеві методи: використовують множину дерев рішень тим самим збільшують ефективність роботи за рахунок усереднення результатів для всіх дерев;

- алгоритми для обчислення числових метрик використовуються для визначення відстані між об’єктами вибірки;

- наївний Байес: прямий розподіл усього моделювання для задач класифікації;

- крос-валідація: в методі немає розподілу набору даних на тренувальну та тестову вибірки, замість цього алгоритм запускається задану кількість разів вибираючи випадковим чином валідаційну вибірку. Підсумковий результат уявляють собою усереднення отриманих результатів.

Для створення Word2Vec моделі та для роботи з її методами була використана бібліотека genism. Gensim – бібліотека[26] обробки природної мови призначення для «Тематичного моделювання». З її допомогою можна обробляти тексти, працювати з векторними моделями слів (такими як Word2Vec, FastText) і створювати тематичні моделі текстів. Тематичне моделювання - це метод вилучення основних тем яким присвячений опрацьований текст. У пакеті Gensim реалізовані основні алгоритми тематичного моделювання LDA і LSI.

3.2 Конструювання програмного забезпечення

Розглянемо функції, які відповідають за роботу бібліотеки. Інформація про них представлена в таблиці 3.1.

Таблиця 3.1 – Специфікації функцій

Назва функції	Вхідні параметри	Вихідні параметри	Призначення
detect_r_peaks	ЕКГ запис, частота	R-піки	Знаходження R-піків в сигналі ЕКГ
select_peaks_by_ecg_size	R-піки, розмір сигналу	R-піки	Знаходження R-піків для проміжку сигналу заданого розміру
draw_ecg_with_r_peak	ЕКГ запис, R-піки	-	Візуалізація R-піків на ЕКГ
find_heartbeat_len	R-піки	Довжина серцебиття	Знаходження тривалості серцебиття
separate_ecg_to_heartbeats	ЕКГ запис, R-піки	Серцебиття	Розподіл запису ЕКГ на серцебиття
separate_ecg_to_heartbeats_with_mark	ЕКГ запис, мітки	Серцебиття з мітками про хворобу	Розподіл запису ЕКГ на серцебиття з мітками про хворобу

Продовження таблиці 3.1

Назва функції	Вхідні параметри	Вихідні параметри	Призначення
select_beats_by_type	Серцебиття	Серцебиття	Фільтрація серцебиттів за типом міток
ecg_wave_detection	Серцебиття	Виділені хвилі	Виділення T-хвиль, QRS-хвиль, P-хвиль в кожному серцебитті
waves_clustering	Хвилі, кількість кластерів	Прогнозовані кластери для кожної хвилі	Кластеризація виділених хвиль
generate_dict_symbol_to_cluster	Кількість кластерів, алфавіт	словник	Створення словника для представлення кожного кластеру символом
transform_cluster_to_symbol	Словник, список кластерів	Список символів	Перетворення номера кластера на символ

Продовження таблиці 3.1

Назва функції	Вхідні параметри	Вихідні параметри	Призначення
transform_heartbeat_to_word	Символи, які відповідають за кластери	Слова, що представляють серцебиття	Представлення кожного серцебиття у вигляді слова
create_word2vec_model	Слова, мінімальна кількість співпадінь	Word2Vec модель	Створення Word2Vec моделі на основі слів що представляють сигнал ЕКГ
create_word2vec_based_on_record	Серцебиття	Виділені хвилі	Виділення T-хвиль, QRS-хвиль, P-хвиль для кожного серцебиття

Висновки до розділу

У розділі було розглянуто засоби для розробки програмного забезпечення, компоненти і технології використанні при проектуванні. Так для розробки бібліотеки була обрана мова програмування Python, для проведення маніпуляцій над даними використана бібліотека numpy, для візуалізації даних модуль matplotlib, для проведення кластеризації та використання методів машинного навчання бібліотека scikit-learn та для створення Word2Vec моделі бібліотека genism.

Також докладно розглянуто структуру бібліотеки та функції для роботи, які вона надає. Для кожної функції описано вхідні параметри, результат виконання функції та її призначення.

4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ WORD2VEC МОДЕЛІ

4.1 Набори даних

4.1.1 MIT-BIH AFIB Dataset

Цей набір даних включає 23 довгострокові записи ЕКГ суб'єктів переважно з аритмією AFIB [27]. Кожен предмет AFIT MIT-BIH містить два 10-годинні записи ЕКГ. Записи ЕКГ відбираються з частотою 250 Гц з 12-бітовою роздільною здатністю в діапазоні ± 10 мілівольт. Для роботи розділяється кожен сигнал ЕКГ на 5-ти секундні сегменти даних, яким ставиться мітка на основі порогового параметра. Дійсно, 5-секундний сегмент даних розглядається як AFIB, якщо відсоток мічених серцебиття AFIB у сегменті становить більше або дорівнює пороговому параметру p, інакше це позначено як неритмічну аритмію. Подібно до літератури, був встановлений параметр p до 50%. Ми отримали в цілому 167 422 5-ти секундних сегментів даних із записів набору даних. Номер AFIB та зразки не AFIB складали 66, 939 та 100, 483 відповідно. Щоб впоратися із проблемою дисбалансу класів, що існує в витягнувши сегменти даних, ми випадковим чином вибрали однакову кількість сегментів як для класів AFIB, так і для не AFIB, в яких ми розглянули 66, 939 сегментів даних для обох класів.

4.1.2 PhysioNet MIT-BIH

Цей набір даних аритмії містить ЕКГ-сигнали для 48 різних суб’єктів. Сигнали були записані при частоті дискретизації 360 Гц, і кожен запис включає два відведення ЕКГ. У цьому дослідженні, щоб бути послідовним з попередніми літературними роботами відвід ЕКГ II використовується для побудови анотатора серцебиття. Рекомендується набір даних Американською асоціацією медичних приладів і складається з п’яти основних груп аритмії. У рисунку 4.1 представлені статистичні дані про кількість маркерів для кожної з груп серцебиття в базі даних MIT-BIH.

Dataset	N	S	V	F	Q	Total
MIT-BIH Arrhythmia	90,462	2,777	7,223	802	8,027	109,291

Рисунок 4.1 – Статистичні дані MIT-BIH

4.2 Зменшення об’єму даних ЕКГ після обробки

Досліджено як зміниться об’єм даних вхідного ЕКГ сигналу і сигналу перетвореного в речення за допомогою лінгвістичного методу. Перетворення відбувається за рахунок заміни кожної виділеної хвилі з серцебиття на символ кластера, якому належить хвиля. Таким чином кожне серцебиття перетворюється в слово, а весь сигнал ЕКГ в послідовність речень.

Проведемо ряд експериментів, використовуючи ЕКГ різної тривалості для перетворення та порівняємо об’єм даних перед обробкою і після.(Таблиця 4.1)

Таблиця 4.1 – Тестування зміни об’єму даних ЕКГ

Початковий розмір сигналу, байт	Кількість кластерів для представлення хвиль	Розмір сигналу після перетворення, байт
1000	25	9
2000	25	21
3000	25	30
5000	25	51
10000	25	105
20000	25	207
30000	25	312
40000	25	417
50000	25	522
75000	25	786
100000	25	1047

Візуалізацію результату виконання спостерігаємо на рисунку 4.2.

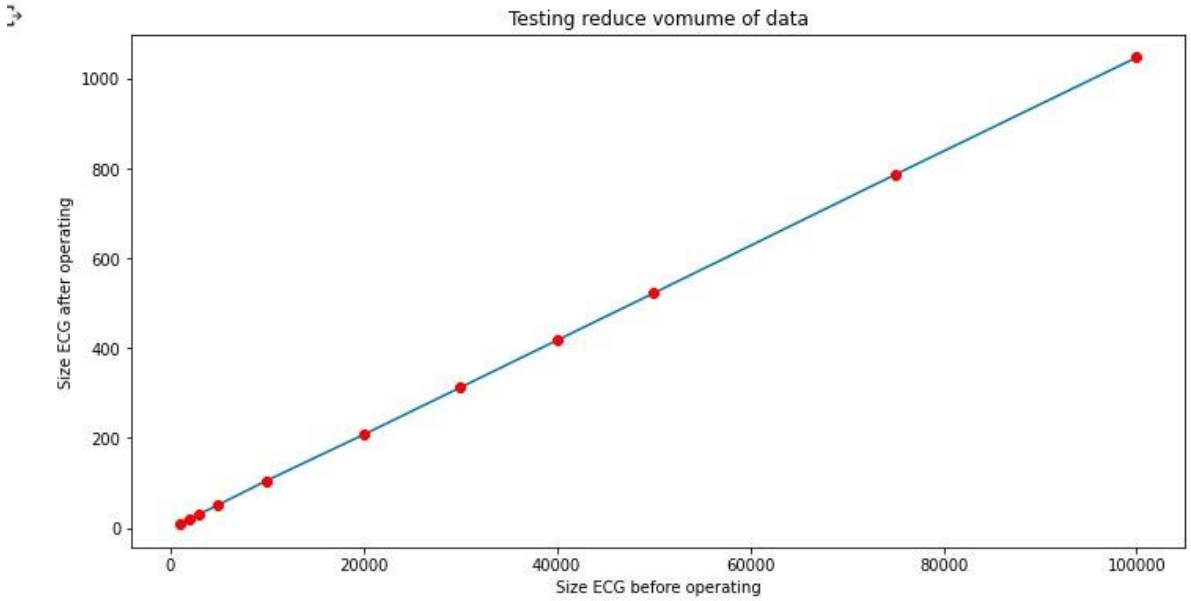


Рисунок 4.2 – Візуалізація зміни об’єму даних

Проаналізувавши графік можемо зробити висновок, що зменшення об’єму даних після обробки сигналу має лінійну залежність з початковим розміром ЕКГ.

4.3 Класифікація серцебиття методом Random Forest

Після створення Word2Vec моделі на основі сигналу ЕКГ, можемо провести класифікацію використовуючи векторне представлення серцебиття з створеної моделі. Класифікувати серцебиття будемо на ознаку наявності аритмії. Для цього проведемо тестування та визначимо час виконання та точність класифікації при різних параметрах.(Таблиця 4.2)

Таблиця 4.2 – Тестування класифікації методом Random Forest

Кількість дерев	Час класифікації, мс	Загальна точність, %
2	23	87
5	47	88
10	79	92
15	110	93
20	146	89

Отже, як видно з результатів тестування, найкращий показник точності спостерігається на конфігурації в 15 дерев. Приблизна точність на цьому показникові – 93 відсотки.

4.4 Порівняння результатів класифікації

Для порівняння результатів класифікації береться набір даних з промаркерованими серцебиттями до перетворення сигналу ЕКГ в речення та набір даних з серцебиттями після використання векторного представлення серцебиття за допомогою лінгвістичного ланцюга Word2Vec моделі. Набори даних були розподілені на тренувальну та тестову вибірки. Класифікація виконувалась методом Random Forest при конфігурації в 15 дерев. Результати порівняння записані в таблиці 4.3.

Таблиця 4.3 – Порівняння класифікації серцебиттів ЕКГ сигналу

Кількість серцебиттів	Розмір сигналу, кілобайт	Метод представлення ЕКГ	Час виконання класифікації, мс	Точність, %
200	57,2	Звичайний	68	91,3
200	20	Word2Vec модель	37	85,1
500	143	Звичайний	152	96,7
500	50	Word2Vec модель	68	90,3
1000	286	Звичайний	336	97,9
1000	100	Word2Vec модель	182	92,9

З результатів порівняння видно, що точність класифікації серцебиттів представлених звичайним способом є трішки вищою (в середньому 5 відсотків) ніж точність класифікації векторного представлення серцебиття за допомогою лінгвістичного ланцюга створеної моделі Word2Vec. Проте видно, що зменшився розмір ЕКГ сигналу, оскільки за допомогою меншого об’єму даних представляється така ж кількість серцебиттів в сигналі. Також видно, що час класифікації при однакових параметрах кількості серцебиттів менший для перетворених серцебиттів. Тому за рахунок того, що швидкість є обернено пропорційною величиною до часу виконання отримується виграш у швидкості.

Висновки до розділу

У розділі розглянуто набори даних, які використовувались для визначення ефективності роботи створеної моделі. Проведено аналіз зменшення об’єму даних вхідного ЕКГ-сигналу та перетвореного в слова. Виявилось, що після перетворення об’єм даних зменшується в середньому в 100 разів.

Було проведено класифікацію методом Random Forest для серцебиття на ознаку аритмії. Найкращий результат роботи алгоритму досягнуто при значенні в 15 дерев, точність класифікації склала близько 93 відсотків.

Проведено порівняння класифікації серцебиттів сигналу ЕКГ для двох випадків представлення ЕКГ сигналу: звичайний та представлення за допомогою лінгвістичного ланцюга моделі Word2Vec. За результатами порівняння зроблено висновок, що даний метод дещо програє в точності, але виграє в збільшенні швидкості та зменшенні об’єму даних (майже в 3 рази) для представлення сигналу ЕКГ.

5 РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

5.1 Опис ідеї проекту

Зміст ідеї

Вихід на ринок ПЗ для інтелектуального аналізу електрокардіограм з використанням спеціалізованих баз знань. Метою проекту є заключення партнерського договору з провідними виробниками подібних систем для отримання фінансування подальшої розробки, та впровадження розробленої системи в якості інтегрованого модуля для відповідного обладнання.

Напрямки застосування

Впровадження програмного забезпечення в лікарні та провідні клініки.

Вигоди для користувача

Підвищення швидкості виявлення захворювань лікарями, автоматизація процесу аналізу ЕКГ, допомога у постановці діагнозу пацієнтам.

Таблиця 5.1 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	*Ідея*	*(потенційні) товари/концепції конкурентів*		*W (слабка сторона)*	*N (нейтральна сторона)*	*S (сильна сторона)*
		Мій проект	*Heart Rate Variability Analysis Software*			
1.	Обробка ЕКГ	Наявний	Наявний		N	
2.	Кластерний аналіз	Наявний	Відсутній		N	
3.	Обробка ЕКГ методами NLP	Наявний	Відсутній			S
4.	Розрахунок подібності двох ЕКГ	Наявний	Відсутній			S

З таблиці 5.1 видно, що мій проект має достатню кількість характеристик з сильними сторонами, і на відміну від його конкурентів реалізує новий підхід, тому є конкурентоспроможним.

5.2 Технологічний аудит ідеї проекту

Таблиця 5.2 – Технологічна здійсненність ідеї проекту

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
1	Обробка ЕКГ	Система розроблена на мові програмування Python 3.8	Наявна	Доступна
		Бібліотека numpy	Наявна	Доступна
2	Кластерний аналіз	Метод K-середніх	Наявна	Доступна
		Агломеративно-ієрархічна кластеризація	Наявна	Доступна
3	Обробка ЕКГ методами NLP	Мова обробки ЕКГ	Наявна	Доступна, є потреба у додатковому вивчені підходів
4	Розрахунок подібності двох ЕКГ	Використання функцій моделі Word2Vec	Наявна	Доступна

Обрана технологія реалізації ідеї проекту: Обрано технологію Python та популярні бібліотеки numpy, matplotlib, sklearn, які є кращими серед наявних на ринку та доступними для членів команди.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

5.3.1 Аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку

Таблиця 5.3 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількість головних гравців, од	1
2	Загальний обсяг продаж, грн/ум.од	Невідомо
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Юридичні обмеження
5	Специфічні вимоги до стандартизації та сертифікації	Вимоги до роботи інформаційної системи у медичних закладах
6	Середня норма рентабельності в галузі (або по ринку), %	Невідома

5.3.2 Визначення потенційних груп клієнтів, їх характеристики, та формування орієнтовного переліку вимог до товару для кожної групи

Таблиця 5.4 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
1	Швидкому аналізи даних електрокардіограм	Пацієнти	відсутні	Автоматизація процесів, швидкість аналізу

Продовження таблиці 5.4

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
2	Консультативна допомога інформаційної системи	Лікарі	відсутні	Допомога у постановці діагнозу, та складання передбачення щодо захворювання пацієнта

5.3.3 Аналіз ринкового середовища

Проведемо аналіз ринкового середовища: складемо таблиці факторів, що сприяють ринковому впровадженню проекту (таблиця 5.5), та факторів, що йому перешкоджають (таблиця 5.6). Фактори в таблиці подані в порядку зменшення значущості.

Таблиця 5.5 – Фактори загроз

№ п/п	*Фактор*	*Зміст загрози*	*Можлива реакція компанії*
1	Персонал	Необхідні навички для роботи з ПЗ	Пошук та навчання людей
2	Недостатнє фінансування	Отримання замалої кількості коштів від інвесторів	Аналіз та оптимізація витрат, маркетингові заходи з пошуку клієнтів
3	Відмова у співпраці	Медична компанія відмовляється від співробітництва	Отримання коментарів щодо причини відмови, пропонування компромісу, введення змін

Таблиця 5.6 – Фактори можливостей

№ п/п	*Фактор*	*Зміст можливості*	*Можлива реакція компанії*
1	Науково-технічний	Вдосконалення інформаційної системи	Впровадження в роботу
2	Робота з закордонними замовниками	Встановлення контактів з іноземними замовниками	Локалізація, інтернаціоналізація та сертифікація системи
3	Економічний	Підтримка інновацій у виробництві	Підвищення / Пониження ціни на послугу

5.3.4 Аналіз пропозиції

Далі проведемо аналіз пропозиції: визначаються загальні риси конкуренції на ринку.

Таблиця 5.7 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Вказати тип конкуренції - монополія	В галузі домінує одна фірма	Надання конкурентоспроможних послуг
2. За рівнем конкурентної боротьби - міжнародний	Компанії конкуренти з інших країн	Створити основу ПП таким чином, щоб можна було легко переробити даний ПП для використання у галузях інших країн.
3. За галузевою ознакою - внутрішньогалузева	Продукт використовується тільки в одній галузі	Постійне вдосконалення продукту

Продовження таблиці 5.7

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
4. Конкуренція за видами товарів: товарно-видова	Конкуренція між видами ПП, їх особливостями.	Вдосконалити ПП, враховуючи недоліки конкурентів
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 - марочна	Бренд присутній, але його роль незначна	Створення власної марки, Реклама, участь у конференціях, семінарах.

5.3.5 Більш детальний аналіз умов конкуренції в галузі

Після аналізу конкуренції розглянемо детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера).

Таблиця 5.8 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	*Конкуренти*		*Постача - льники*	*Клієнти*	*Товари замінники*
	Прямі	*Потенційні*			
	HRV software	Невідомі	-	Контроль якості продукту	Наявність більш широкого функціоналу , зручнішого інтерфейсу та авторитет (якість), наявність необхідних сертифікацій

Продовження таблиці 5.8

	Конкуренти		*Постача-льники*	*Клієнти*	*Товари замінники*
Висновки	Досить інтенсивна конкурентна боротьба з вже закріпленими на ринку гравцями	Дані відсутні	-	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.

Висновок: З проведеного аналіз у конкуренції на ринку нами було виявлено, що існує можливість виходу на ринок. Як стратегію на початку роботи можна обрати напрямок роботи на встановлення зв’язків з існуючими замовниками послуг.

5.3.6 Обґрунтування переліку факторів конкурентоспроможності

На основі аналізу конкуренції, проведеного в п. 5.3.5 (таблиця 5.8), а також із урахуванням характеристик ідеї проекту (таблиця 5.1), вимог споживачів до товару (таблиця 5.4) та факторів маркетингового середовища (таблиці 5.5 і 5.6) визначимо та обґрунтовуємо перелік факторів конкурентоспроможності (таблиця 5.9).

Таблиця 5.9 – Обґрунтування факторів конкурентоспроможності

№ п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
1	Точність	Використання сучасних технологій для отримання найбільш точних даних
2	Ціна	Можливість давати більш точні результати аналізу, та вихід на медичні ринки.
3	Дані, що зчитуються	Можливість безпечно змінювати дані
4	Орієнтованість на кінцевого споживача	Продукт орієнтований на взаємодію з клієнтом

5.3.7 Аналіз сильних та слабких сторін проекту

За визначеними факторами конкурентоспроможності (таблиця 5.9) проводиться аналіз сильних та слабких сторін стартап-проекту.

Таблиця 5.10 – Порівняльний аналіз сильних та слабких сторін «назва проекту»

№ п/п	*Фактор конкурентоспроможності*	*Бали 1-20*	*Рейтинг товарів-конкурентів у порівнянні з моїм проектом*						
			–3	*–2*	*–1*	*0*	*+1*	*+2*	*+3*
1	Точність	15			+				
2	Ціна	19		+					
3	Дані, що зчитуються	20	+						
4	Орієнтованість на кінцевого споживача	12			+				

5.3.8 SWOT-аналіз

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 5.11) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 5.10).

Перелік ринкових загроз та ринкових можливостей складається 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 5.11 – SWOT- аналіз стартап-проекту

Сильні сторони: - Порівняно низька ціна - Широкий функціонал - Точність	Слабкі сторони: - Нерозуміння потреб ринку
Можливості: - Конкуренція - Нові методи аналізу ЕКГ	Загрози: - Персонал - Недостатнє фінансування - Відмова у співпраці

5.3.9 Альтернативи ринкової поведінки

На основі SWOT-аналізу розробимо 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 5.8, аналіз потенційних конкурентів). Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів.

Таблиця 5.12 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
1	Безкоштовне розповсюдження створеного Програмного продукту	Дуже висока	9-12 місяців
2	Створення програмного продукту з подальшим розповсюдженням за певну оплату	Висока	18-24 місяці

Отже, логічно обрати першу альтернативу та згодом розробити розширений функціонал, якщо підхід буде мати попит на ринку.

5.4 Розроблення ринкової стратегії проекту

5.4.1 Опис цільових груп потенційних споживачів

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 5.13 – 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 діяльність яких пов'язана з сферою страхової медицини.*	Висока	Високий	Сильна	Складно

Продовження таблиці 5.13

№ 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*
2	Приватні підприємства міського та міжнародного рівня, діяльність яких пов'язана з медичними дослідженнями	Висока	Високий	Сильна	Складно
3	Приватні клініки	Помірна	Помірний	Помірна	Середня
4	Заклади медичного туризму	Помір на	Слабкий	Слабка	Просто
5	Держ. клініки	Слабка	Слабкий	Слабка	Просто

За результатами аналізу потенційних груп споживачів було обрано цільові групи – 1,2,3, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку – стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

5.4.2 Базова стратегія розвитку

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку.

Таблиця 5.14 – Визначення базової стратегії розвитку

№ п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку**
1	Стратегія спеціалізації	Налагодження зв'язків з клієнтами, індивідуальна модифікація ПЗ під. потреби	Висока якість та точність, ухил на довготривалі стосунки	Стратегія диференціації

Базовою стратегією оберемо стратегію диференціації – орієнтування на потреби користувача. Альтернативною до неї (у разі провалу) буде обрано стратегію спеціалізації – налаштування під окремий цільовий сегмент.

5.4.3 Вибір стратегії конкурентної поведінки

Наступним кроком є вибір стратегії конкурентної поведінки.

Таблиця 5.15 – 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	Проект не є «першопрохідцем»	Пошук нових клієнтів та перехоплення існуючих	Копіювання лише спільного функціоналу та його розширення	Стратегія заняття конкурентної ніші

5.4.4 Стратегія позиціонування

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

Таблиця 5.16 – Визначення стратегії позиціонування

№ п/п	*Вимоги до товару цільової аудиторії*	*Базова стратегія розвитку*	*Ключові конкурентоспроможні позиції власного стартап-проекту*	*Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)*
1	Легкість розуміння, зручний інтерфейс, надійний,	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Економія часу; Зручність застосування;
2	Швидкий, точний та достовірний ПП для аналізу ЕКГ	Стратегія диференціації	Потужна рекламна компанія, Висока якість програмного засобу	Практичність та точність результату
3	Невелика вартість	Спеціальні пропозиції	Період безкоштовного користування усіма функціями	Доступність

5.5 Розроблення маркетингової програми стартап-проекту

5.5.1 Маркетингова концепція товару

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього потрібно підсумуємо результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.17 – 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;	Швидке зняття кардіограми	Відсутність необхідності звертатися до сторонньої особи/компанії для зняття електрокардіограми. Дані 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 проводити зняття, та отримувати аналіз електрокардіограми	ПП має вбудовану інструкцію по користуванню
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. Отриманий прогноз можна для інтерпретації стану здоров’я.

5.5.2 Маркетингова модель товару

Розробимо трирівнева маркетингова модель товару: уточнення ідеї продукту та/або послуги, його фізичні складові, особливості процесу його надання.

Таблиця 5.18 – Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="3">Сутність та складові</td></tr>
<tr><td></td><td colspan="3"></td></tr>
<tr><td>I. Товар за задумом</td><td colspan="3">Зручність та швидкість отримання практичного результату щодо побудови моделі та прогнозування процесів</td></tr>
<tr><td rowspan="5">II. Товар у Реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх /Тл/Е/Ор</td></tr>
<tr><td>Якість
Точність
Встановлення у медичний заклад
Ціна</td><td>Нм
Нм
М
Нм</td><td>Тл
Тл
Тх
Е</td></tr>
<tr><td colspan="3">Якість: достовірність побудови аналізу здоров’я серця у домашніх умовах</td></tr>
<tr><td colspan="3">Пакування: відсутнє</td></tr>
<tr><td colspan="3">Марка: ECG Analyzer</td></tr>
<tr><td>III. Товар із підкріпленням</td><td colspan="3">Після продажу: персональна підтримка в обслуговуванні за додаткову платню.</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент на винахід.</td></tr>
</table>

За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності.

5.5.3 Визначення цінових меж встановлення ціни

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. Аналіз проводиться експертним методом.

Таблиця 5.19 – 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	-	3000 грн одноразово	10 000 грн / міс і вище	1000 грн одноразово для користувачів

5.5.4 Оптимальна система збуту

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення:

– проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);

– вибір та обґрунтування оптимальної глибини каналу збуту;

– вибір та обґрунтування виду посередників.

Таблиця 5.20 – Формування системи збуту

№ п/п	*Специфіка закупівельної поведінки цільових клієнтів*	*Функції збуту, які має виконувати постачальник товару*	*Глибина каналу збуту*	*Оптимальна система збуту*
1	Орієнтація на безкоштовний функціонал	1. Забезпечення ефективного алгоритму 2. Пробний безкоштовний період користування повним функціоналом	Середня	Безпосередня

5.5.5 Розроблення стратегії маркетингових комунікацій

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

Таблиця 5.21 – 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*
1	Орієнтація на якісний застосунок	StackOverFlow, GitHub,	Співвідношення ціна/якість	Якість продукту та рекомендацій	Не вигадуй велосипед. Рішення вже існує

Висновки до розділу

Отже, в даному розділі було проведено аналіз розробленого методу, як частину стартап проекту. Можна зазначити, що проект є досить цікавим з точки зору комерціалізації, так як ринок на який виходить проект все іще формується. На ринку наявна певна кількість конкурентів, але завдяки грамотній стратегії виходу, можливо зайняти правильну нішу. Можна сказати, що подальший розвиток проекту є доцільним, оскільки він знайде свою цільову аудиторію.

ВИСНОВКИ

В рамках магістерської дисертації розроблено програмне забезпечення створення моделі Word2Vec на основі сигналу ЕКГ. Програмний продукт написаний на мові програмування Python у вигляді бібліотеки, яка містить методи для обробки сигналу, перетворення електрокардіограми в послідовність речень та функції для створення Word2Vec моделі.

Розглянуто поняття ЕКГ сигналу та його складових, методи виявлення R-піків в електрокардіограмі, поняття кластерного аналізу та алгоритми кластеризації, методи пониження розмірності даних, методи природної обробки мови, векторного представлення слів, а також алгоритми для створення Word2Vec моделі. Розписано завдання, які мають бути вирішенні в рамках дослідження.

Детально описано підхід представлення структури для обробки електрокардіограми подібно до обробки природною мовою текстового документа, описано етапи попередньої обробки ЕКГ сигналу, перетворення кожного серцебиття в слово, а всього ЕКГ-сигналу в послідовність речень. Було розглянуто процес створення та тренування Word2Vec моделі за допомогою двох різних підходів.

Розглянуто засоби для розробки програмного забезпечення, компоненти і технології використанні при проектуванні. Так для розробки бібліотеки була обрана мова програмування Python, для проведення маніпуляцій над даними використана бібліотека numpy, для візуалізації даних модуль matplotlib, для проведення кластеризації та використання методів машинного навчання бібліотека scikit-learn та для створення Word2Vec моделі бібліотека genism.

Для визначення ефективності роботи програмного забезпечення було проведено аналіз зменшення об’єму даних вхідного ЕКГ-сигналу та перетвореного в слова. Виявилось, що після перетворення об’єм даних зменшується в середньому в 100 разів. А також для визначення точності моделі було проведено класифікацію методом Random Forest для кожного серцебиття з ЕКГ на ознаку аритмії. Найкращий результат роботи алгоритму досягнуто при формуванні в 15 дерев,

точність класифікації склала близько 93 відсотків. З експериментальних досліджень можна зробити висновок, що модель є ефективною для роботи з великими об’ємами даних.

Наукова новизна розробки полягає у новому підході для представлення у векторній структурі серцевого такту, можливості знаходження різниці між сигналами за рахунок обрахунку косинуса подібності та для знаходження ключових тактів, що показують важливість серцебиття для всього сигналу ЕКГ.

Усі поставленні задачі наукової роботи були виконані. Було створено програмну бібліотеку з усіма методами необхідними для створення Word2Vec моделі на основі сигналу ЕКГ, а також проведено ефективність її роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) О. Й. Жарінов, В. О. Куць (2017) Основи електрокардіографії:[навч. посіб. для лікарів-слухачів закл. (ф-тів) післядиплом. освіти / Жарінов О. Й. та ін.] — Бібліогр.: с. 235—236

2) Baklan I., Mukha I., Oliinyk Y., Lishchuk K., Nedashkivsky E., Gavrilenko O. (2020) Anomalies Detection Approach in Electrocardiogram Analysis Using Linguistic Modeling. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol 938. Springer, Cham; pp 513-522, DOI - https://dx.doi.org/10.1007/978-3-030-16621-2_48; (Scopus)

3) J. Pan and W. J. Tompkins, “A real-time qrs detection algorithm,” IEEE Trans. Biomed. Eng, vol. 32, no. 3, pp. 230–236, 1985.

4) Engelse, W. A. H. and Zeelenberg, C. (1979). A single scan algorithm for QRS-detection and feature extraction. Computers in Cardiology, 6:37–42.

5) Hamilton, P. (2002). Open source ecg analysis. Computersin Cardiology

6) Журавлев Ю. И., Рязанов В. В., Сенько О. В. Распознавание. Математические методы. Программная система. Практические применения. — М.: Фазис, 2006.

7) Tryon, Robert C. (1939). Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality. Edwards Brothers.

8) J. A. Lozano J. M. Pena and P. Larranaga, An empirical comparison of four initialization methods for the k-means algorithm," Pattern Reognition Letters, vol. 20, pp. 1027-1040,1999.

9) М. Жамбю, Ієрархічний кластерний аналіз та відповідності. — М.: Фінанси і статистика, 1988. — 345 с.

10) M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. 9th International Conference on Artificial Intelligence and Statistics, 2002.

11) Abdi H., Williams L.J. (2010). Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2: 433–459.

12) Van der Maaten, L.J.P.; Hinton, G.E. (Nov 2008). Visualizing Data Using t-SNE. Journal of Machine Learning Research 9: 2579–2605.

13) Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc. — P.1

14) История Компьютера Обработка естественного языка: офіційний сайт: [Електрон. ресурс]. – Режим доступа: http://chernykh.net/content/view/1105/1189/

15) Mikolov, T., Yih W., Zweig G. Linguistic regularities in continuous space word representations. // Proc of NAACL-HLT 2013. P. 746–751.

16) McGinnis W. Beyond one-hot: an exploration of categorical variables // Data science, technology, Atlanta. – 2015;

17) Pennington, J., Socher R., Manning C.D. Global Vectors for Word Representation. // Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing, P. 1532–1543.

18) Takala, P. Word Embeddings for Morphologically Rich Languages // Computational Intelligence and Machine Learning. Belgium. Bruges. 2016. P. 27–29.

19) Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // Proc. of Workshop at ICLR. 2013. P. 1301-3781.

20) Ho, Tin Kam (1995). Random Decision Forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

21) Sajad Mousavi, Fatemeh Afghah, Fatemeh Khadem, and U. Rajendra Acharya ECG Language Processing (ELP): New Technique to Analyze ECG Signals, 2020, pp. 2-4

22) Python Release Python 3.9.1: офіційний сайт: [Електрон. ресурс]. – Режим доступа: Python.org

23) NumPy: офіційний сайт: [Електрон. ресурс]. – Режим доступа: numpy.org

24) Matplotlib: офіційний сайт: [Електрон. ресурс]. – Режим доступа: https://matplotlib.org

25) Scikit-learn: офіційний сайт: [Електрон. ресурс]. – Режим доступа: – https://scikit-learn.org

26) Gensim: офіційний сайт: [Електрон. ресурс]. – Режим доступа: https://radimrehurek.com/gensim_3.8.3/

27) Munger TM, Wu LQ, Shen WK (2014). "Atrial fibrillation". Journal of Biomedical Research. pp. 1–17

ДОДАТОК А ПРОГРАМНИЙ КОД

```
import numpy as np
import matplotlib.pyplot as plt
from ecgdetectors import Detectors
from sklearn.cluster import KMeans
from gensim.models import Word2Vec




def detect_r_peaks(ecg_record, fs):
    detector = Detectors(fs)
    unfiltered_ecg = ecg_record[0][:, 1]
    r_peaks = detector.engzee_detector(unfiltered_ecg)
    return r_peaks




def select_peaks_by_ecg_size(peaks, ecg_size):
    selected_peaks = []
    for peak in peaks:
        if peak < ecg_size:
            selected_peaks.append(peak)
        else:
            break
    return selected_peaks




def draw_ecg_with_r_peak(peaks, record, ecg_size):
    draw_peaks = select_peaks_by_ecg_size(peaks, ecg_size)
    plt.figure(figsize=(24, 6))
    ecg_data = record[0][:, 1]
    plt.plot(ecg_data[:ecg_size])
    plt.plot(draw_peaks, ecg_data[draw_peaks], 'ro')
    plt.show()




def find_heartbeat_len(peaks):
    distances = []
    for i in range(len(peaks)):
        if i + 1 < len(peaks):
            distances.append(peaks[i + 1] - peaks[i])
```

```
    distances = np.array(distances)
    return int(distances.mean())



def separate_ecg_to_heartbeats(record):
    fs = record[1]['fs']
    unfiltered_ecg = record[0][:, 1]
    r_peaks = detect_r_peaks(record, fs)
    heartbeat_len = find_heartbeat_len(r_peaks)
    beats = []
    for i in range(len(r_peaks)):
        if i != 0:
            r_index = r_peaks[i]
            retreat = heartbeat_len // 2
            beats.append(unfiltered_ecg[r_index - retreat:r_index +
retreat])
    return beats



def separate_ecg_to_heartbeats_with_annotation(record, annotation):
    fs = record[1]['fs']
    sample = annotation.__dict__['sample']
    marker = annotation.__dict__['symbol']
    unfiltered_ecg = record[0][:, 1]
    r_peaks = detect_r_peaks(record, fs)
    heartbeat_len = find_heartbeat_len(r_peaks)
    beats = []
    annotated_beats = []
    for i in range(len(r_peaks)):
        if i != 0:
            r_index = r_peaks[i]
            retreat = heartbeat_len // 2
            beats.append(unfiltered_ecg[r_index - retreat:r_index +
retreat])
            for j in range(len(sample)):
                if r_index - retreat < sample[j]:
                    annotated_beats.append(marker[j])
                    break
    return {'beats': beats, 'annotated_beats': annotated_beats}
```

```
def select_beats_by_type(beats, annotated_beats, markers):
    selected_beats = []
    selected_annotation = []
    for i in range(len(beats)):
        if annotated_beats[i] in markers:
            selected_beats.append(beats[i])
            selected_annotation.append(annotated_beats[i])
    return {'beats': selected_beats, 'annotated_beats': selected_annotation}
```

```
def ecg_wave_detection(heart_beats):
    p_waves = []
    qrs_waves = []
    t_waves = []
    retreat = len(heart_beats[0]) // 2
    for j in range(len(heart_beats)):
        if j + 1 != len(heart_beats):
            p_waves.append(heart_beats[j][:retreat - 15])
            qrs_waves.append(heart_beats[j][retreat - 15:retreat + 15])
            t_waves.append(heart_beats[j][retreat + 15:])
    return {'p_waves': p_waves, 'qrs_waves': qrs_waves, 't_waves': t_waves}
```

```
def waves_clustering(waves, amount_clusters):
    waves = np.array(waves)
    kmeans = KMeans(init='k-means++', n_clusters=amount_clusters, n_init=10)
    kmeans.fit(waves)
    predict_cluster = kmeans.predict(waves)
    return predict_cluster
```

```
def generate_dict_symbol_to_cluster(amount_cluster, alphabet):
    return {i: alphabet[i] for i in range(amount_cluster)}
```

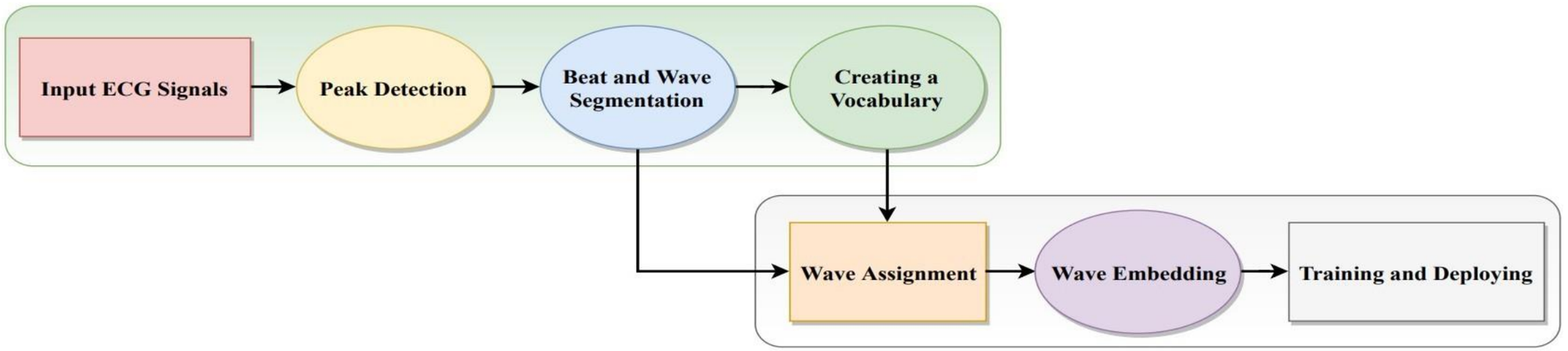
```
def transform_cluster_to_symbol(predicted_cluster, vocab):
    def get_item(x):
        return vocab[x]
```

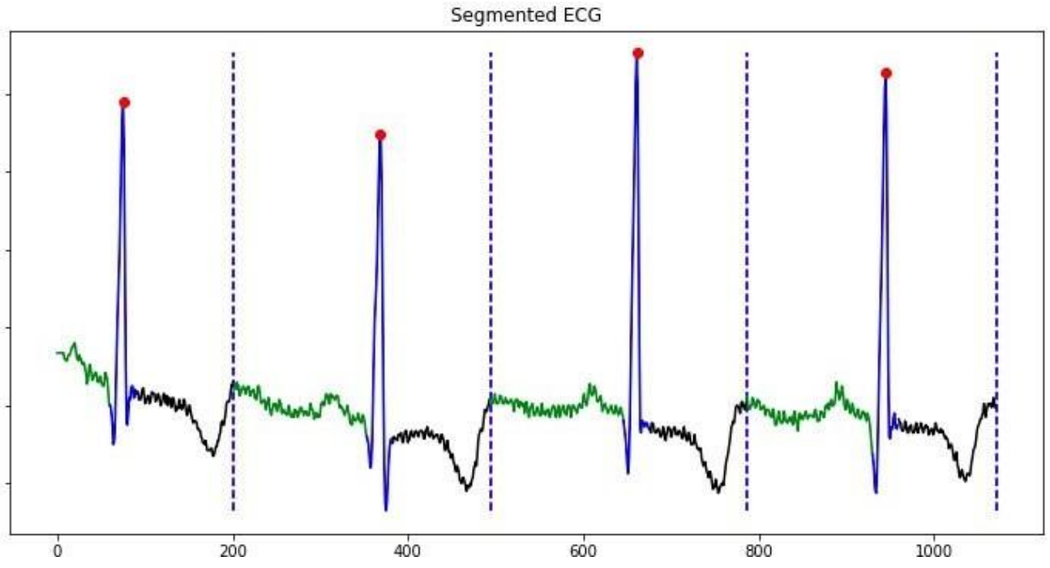
```
    vfunc = np.vectorize(get_item)
    predicted_symbol = vfunc(predicted_cluster)
    return predicted_symbol



def transform_heartbeat_to_word(pt_symbol, qrs_symbol):
    words = []
    for i in range(len(qrs_symbol)):
        word = ''
        word += pt_symbol[i]
        word += qrs_symbol[i]
        word += pt_symbol[i + len(qrs_symbol) // 2]
        words.append(word)
    return words



def create_word2vec_model(words, min_count=0):
    word2vec = Word2Vec([words, ], min_count=min_count)
    return word2vec



def create_word2vec_based_on_record(record, annotation):
    data = separate_ecg_to_heartbeats_with_annotation(record, annotation)
    beats = data['beats']
    waves_data = ecg_wave_detection(beats)
    p_waves = waves_data['p_waves']
    qrs_waves = waves_data['qrs_waves']
    t_waves = waves_data['t_waves']
    alphabet = 'abcdefghijklmnopqrstuvwxyz'
    amount_cluster_pt = 15
    pt = np.array(p_waves + t_waves)
    pt_clustering = waves_clustering(pt, amount_cluster_pt)
    vocab = generate_dict_symbol_to_cluster(amount_cluster_pt,
                                            alphabet[:amount_cluster_pt])
    pt_symbols = transform_cluster_to_symbol(pt_clustering, vocab)
    amount_cluster_qrs = 10
    qrs_clustering = waves_clustering(qrs_waves, amount_cluster_qrs)
    vocab = generate_dict_symbol_to_cluster(amount_cluster_qrs,
                                            alphabet[amount_cluster_pt:])
    qrs_symbols = transform_cluster_to_symbol(qrs_clustering, vocab)
```

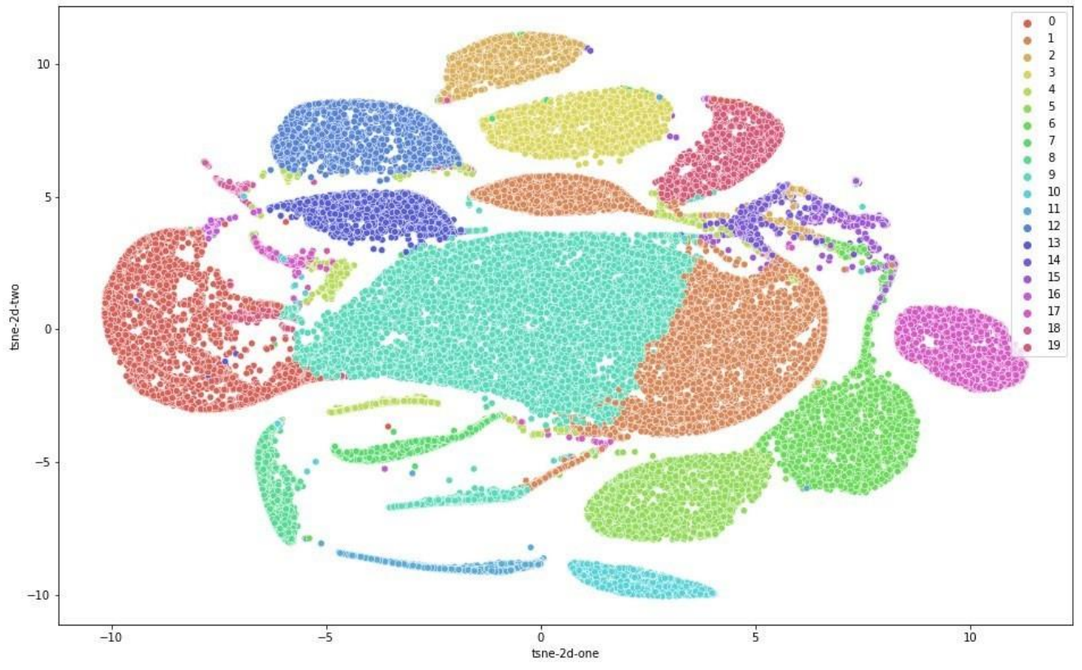
```
words = transform_heartbeat_to_word(pt_symbols, qrs_symbols)
word2vec = Word2Vec([words, ], min_count=0)
return {'model': word2vec, 'words': words, 'beats': data['beats'],
        'annotation': data['annotated_beats']}
```

ДОДАТОК Б ГРАФІЧНІ МАТЕРІАЛИ

Процес створення моделі







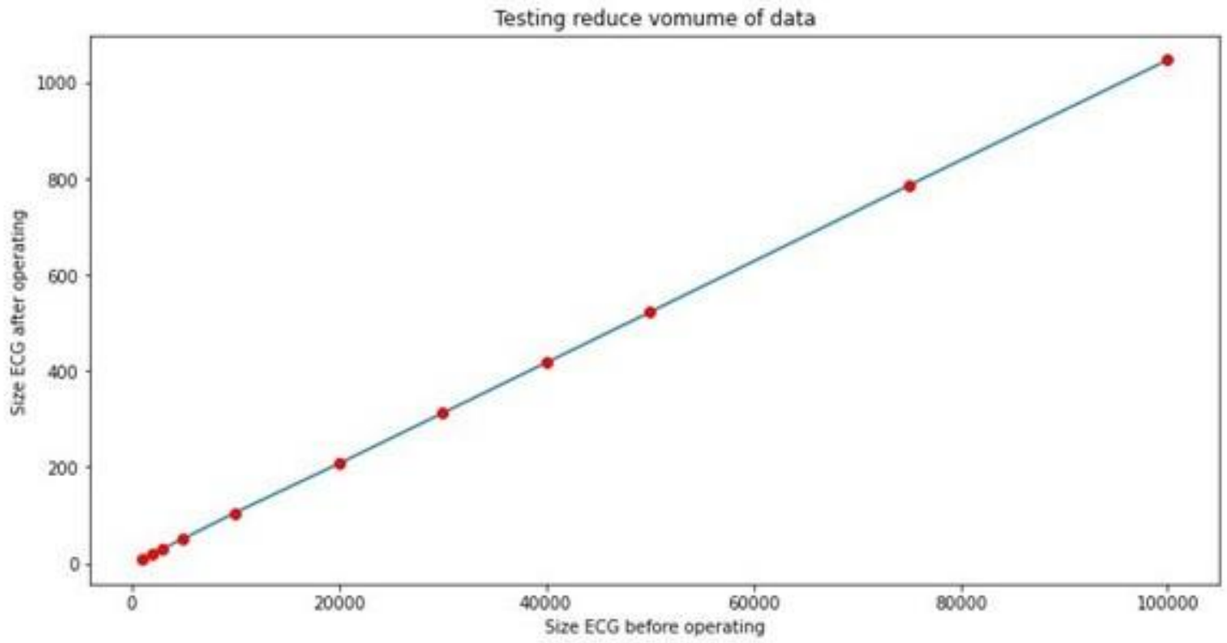

```
{'bue': <gensim.models.keyedvectors.Vocab at 0x7fb3146bb128>,
 'bun': <gensim.models.keyedvectors.Vocab at 0x7fb3146467f0>,
 'buo': <gensim.models.keyedvectors.Vocab at 0x7fb3146b9d68>,
 'buq': <gensim.models.keyedvectors.Vocab at 0x7fb314646828>,
 'bur': <gensim.models.keyedvectors.Vocab at 0x7fb3146bb6a0>,
 'but': <gensim.models.keyedvectors.Vocab at 0x7fb314646668>,
 'bva': <gensim.models.keyedvectors.Vocab at 0x7fb3146462e8>,
 'bvq': <gensim.models.keyedvectors.Vocab at 0x7fb314646860>,
 'byl': <gensim.models.keyedvectors.Vocab at 0x7fb3146b9dd8>,
 'bym': <gensim.models.keyedvectors.Vocab at 0x7fb3146465f8>,
 'byo': <gensim.models.keyedvectors.Vocab at 0x7fb3146bb160>,
 'byt': <gensim.models.keyedvectors.Vocab at 0x7fb314646710>,
```

Демонстраційний плакат до магістерської дисертації
Процес створення моделі

Виконав студент гр. ІП-92мп Терещенко А.С.
Керівник Олійник Ю.О.

Дослідження ефективності

Початковий розмір сигналу, байт	Кількість кластерів для представлення хвиль	Розмір сигналу після перетворення, байт
1000	25	9
2000	25	21
3000	25	30
5000	25	51
10000	25	105
20000	25	207
30000	25	312
40000	25	417
50000	25	522
75000	25	786
100000	25	1047



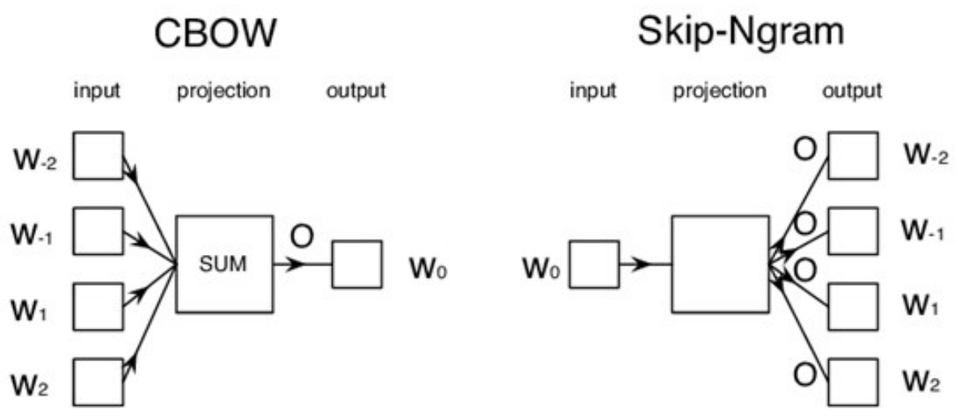
Кількість серцебиттів	Розмір сигналу, кілобайт	Метод представлення ЕКГ	Час виконання класифікації, мс	Точність, %
200	57,2	Звичайний	68	91,3
200	20	Word2Vec модель	37	85,1
500	143	Звичайний	152	96,7
500	50	Word2Vec модель	68	90,3
1000	286	Звичайний	336	97,9
1000	100	Word2Vec модель	182	92,9

Кількість дерев	Час класифікації, мс	Загальна точність, %
2	23	87
5	47	88
10	79	92
15	110	93
20	146	89

Демонстраційний плакат до магістерської дисертації
Процес створення моделі

Виконав студент гр. ІП-92мп Терещенко А.С.
Керівник Олійник Ю.О.

Робота з Word2Vec моделлю




```
kwb : [-4.52855631e-04  2.69200653e-03 -2.85770698e-03 -3.71513655e-03
  7.24595389e-04 -6.08999457e-04  2.23849644e-03 -3.87491775e-03
 -1.76306057e-03  6.52205083e-04  7.47220067e-04 -2.37480062e-03
  2.80323718e-03  1.15808658e-03 -1.81753351e-03 -2.80798832e-03
 -5.11458609e-03  3.07700131e-03  2.54009705e-04  4.26528323e-03
 -1.63862924e-03  4.70669661e-03  3.97165120e-03  3.91787663e-03
 -3.11121764e-03  3.04562040e-03 -2.35919980e-03  1.19900316e-04
 -5.36466995e-03  2.48389272e-03 -1.44891499e-03 -7.83059513e-04
  3.53254564e-03 -2.12145061e-03 -1.40378485e-03  3.31551279e-03
  4.04210994e-03  1.01519178e-03  1.57758989e-03  4.68220329e-03
  1.86871411e-03 -6.41508261e-04  2.16125301e-03  1.22731004e-03
 -1.09590031e-03 -6.07335009e-03  1.30223471e-03 -3.83058796e-03
  2.22818181e-03 -4.63103503e-03 -1.52695086e-03  3.62753542e-03
 -2.10220553e-03  4.25343588e-03  2.28840276e-03  3.60550778e-03
 -5.99312079e-05 -3.19302292e-03 -4.41611465e-03  5.54119237e-04
 -5.55756828e-03  4.10656631e-03  2.68298457e-03  4.14208882e-03
 -8.60912609e-04  5.34155697e-04  3.25926510e-03  2.76880083e-03
  1.42918539e-03 -3.18572158e-03 -2.72700260e-03 -2.76432396e-03
  2.55474891e-03  2.83807237e-03 -8.61959648e-04  4.05401969e-03
  3.96387372e-03 -2.75863800e-03 -4.49991878e-03 -1.00883329e-03
  4.77602240e-03  2.75431201e-03 -1.71090907e-03 -1.92280975e-03
  4.18488542e-03  6.76358759e-04 -7.16106326e-04  1.82076625e-03
  3.56791681e-03 -1.83787569e-03 -8.79865547e-05  1.55338924e-03
 -2.37935386e-03 -4.98656929e-03 -4.83540772e-03 -3.00369668e-03
  1.35593917e-04  3.89454677e-03  3.50713078e-03 -5.40652266e-03]
```

```
'kwb': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99550>,
'kwc': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88e48>,
'kwd': <gensim.models.keyedvectors.Vocab at 0x7f55d8ea1da0>,
'kwe': <gensim.models.keyedvectors.Vocab at 0x7f55d8eab438>,
'kwf': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95160>,
'kwg': <gensim.models.keyedvectors.Vocab at 0x7f55d8eab358>,
'kwi': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95f98>,
'kwj': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99588>,
'kwk': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88f98>,
'kwl': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95208>,
'kwm': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99f60>,
'kwn': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99fd0>,
'kxa': <gensim.models.keyedvectors.Vocab at 0x7f55d8e91d68>,
'kxb': <gensim.models.keyedvectors.Vocab at 0x7f55d8e995f8>,
'kxc': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95ac8>,
'kxe': <gensim.models.keyedvectors.Vocab at 0x7f55d8ea12e8>,
'kxi': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95278>,
'kxj': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99e10>,
'kxk': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88da0>,
'kxm': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99f28>,
'kxn': <gensim.models.keyedvectors.Vocab at 0x7f55d8ea1748>,
'lqa': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95a90>,
'lqb': <gensim.models.keyedvectors.Vocab at 0x7f55d8ea1710>,
'lqc': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88cf8>,
'lqe': <gensim.models.keyedvectors.Vocab at 0x7f55d8e99b00>,
'lqf': <gensim.models.keyedvectors.Vocab at 0x7f55d8e95400>,
'lqi': <gensim.models.keyedvectors.Vocab at 0x7f55d8e88f60>,
```

```
similary = model.wv.most_similar('gxo')
similary
```

```
[('fyj', 0.288496196269989),
 ('gya', 0.2614765167236328),
 ('brl', 0.25019213557243347),
 ('bxm', 0.2501004636287689),
 ('gqf', 0.2496567666530609),
 ('mtm', 0.24112728238105774),
 ('mrj', 0.2403239607810974),
 ('mpj', 0.2283603549003601),
 ('dpg', 0.228246808052063),
 ('gqo', 0.22730636596679688)]
```

Демонстраційний плакат до магістерської дисертації
Процес створення моделі

Виконав студент гр. ІП-92мп Терещенко А.С.
Керівник Олійник Ю.О.

UNICHECK

Ім'я користувача:
Попенко Володимир Дмитрович

ID перевірки:
1005462518

Дата перевірки:
15.12.2020 15:02:13 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
15.12.2020 15:31:23 EET

ID користувача:
77149

Назва документа: **Tereschenko_magistr_ip92mp (1)**

Кількість сторінок: **44** Кількість слів: **7995** Кількість символів: **59745** Розмір файлу: **2.39 MB** ID файлу: **1005752240**

10.8%
Схожість

Найбільша схожість: **1.86%** з джерелом з Бібліотеки (ID файлу: **1003961331**)

6.55% Джерела з Інтернету	46	Сторінка 46
8.43% Джерела з Бібліотеки	128	Сторінка 47

0% Цитат

Не знайдено жодних цитат

Вилучення списку бібліографічних посилань вимкнене

72.2%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за **8 слів** та **0%**)

Немає вилучених Інтернет-джерел

72.2% Вилученого тексту з Бібліотеки	1	Сторінка 47