Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія класифікації кардіограм на основі спайкінгової нейромережі»

Виконав: студент 2-го курсу, групи 1КН-23м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
_____ Завальнюк Я. Є.
(прізвище та ініціали)

Керівник: к.т.н., професор каф. КН
_____ Колесницький О.К.
(прізвище та ініціали)
«05» 12 2024 р.

Опонент: к.т.н., професор каф. КСУ
_____ Биков М. М.
(прізвище та ініціали)
«05» 12 2024 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)
«06» 12 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти ІІ-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.
11.10 **2024 року**

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Завальнюку Ярославу Євгенійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія класифікації кардіограм на основі спайкінгової нейромережі.

керівник роботи к.т.н., проф. кафедри КН Колесницький О. К.
затверджені наказом вищого навчального закладу від "17" 09 2024 року №310
2. Строк подання студентом роботи 11.11 2024 року

3. Вихідні дані до роботи:
Кількість класів ЕКГ– не менше 15, вид класифікатора – нейронна мережа, кількість навчальних зразків ЕКГ – не менше 500, кількість тестових зразків ЕКГ – не менше 100, кількість нейронів у мережі – не менше 250, використання об'єктно-орієнтованої мови програмування.
4. Зміст текстової частини:
Вступ, аналіз предметної області класифікації кардіограм, розробка інформаційної технології класифікації кардіограм, програмна реалізація інформаційної технології класифікації кардіограм за допомогою спайкінгової нейронної мережі, тестування та аналіз результатів роботи програми класифікації кардіограм на основі спайкінгової нейронної мережі, економічна частина, висновки, перелік використаних джерел, додатки
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
Алгоритм роботи програми, UML діаграма класів, робочі вікна програми, результати тестування програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Колесницький О.К., к.т.н., проф. каф. КН		
5	Лесько О. Й., к. е. н., проф. зав. каф. ЕПВМ		

7. Дата видачі завдання 02.09 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасного рівня інформаційних технологій класифікації кардіограм. Постановка задач дослідження	02.09.24 - 09.09.24	
2	Побудова моделей класифікації кардіограм	10.09.24-28.09.24	
3	Практичне застосування та оцінка ефективності розроблених моделей	29.09.24-18.10.24	
4	Підготовка економічної частини	19.10.24 - 01.11.24	
5	Апробація та/або впровадження результатів дослідження	2.11.24 - 4.11.24	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	05.11.24 - 11.11.24	

Студент _____ Завальнюк Я. Є.
(підпис)

Керівник роботи _____ Колесницький О.К.

АНОТАЦІЯ

УДК 004.8

Завальнюк Я. Є. Інформаційна технологія класифікації кардіограм на основі спайкінгової нейромережі. Магістерська кваліфікаційна робота зі спеціальності 122 – «Комп'ютерні науки», освітня програма – «Системи штучного інтелекту». Вінниця: ВНТУ, 2024. 92 с.

На укр. мові. Бібліогр.: 33 назв; рис.: 12; табл. 9.

У роботі було обґрунтовано вибір спайкінгової нейронної мережі для класифікації кардіограм, яка перетворює вхідну кардіограму у послідовність спайків, які потім розпізнаються спайкінговою нейромережею, яка має 4 вхідних нейрони, 18 вихідних нейронів і рекурентний шар із 256 збудливих нейронів та 64 гальмівних нейронів та може розпізнавати 18 видів аритмії. Програмне забезпечення класифікації кардіограм створено на мові програмування Python з використанням спеціалізованих бібліотек NumPy та Matplotlib. Навчання та тестування програмного забезпечення відбувалось з використанням набору даних кардіограм PhysioNet Arrhythmia Database (MIT/BIH). Навчальна вибірка складалась із 15773 кардіограм. Тестова вибірка складалась із 3955 кардіограм. Розроблене програмне забезпечення має достовірність класифікації кардіограм на тестовій вибірці 96,3%, а програма-аналог - 92,4%. Таким чином, розроблене програмне забезпечення має порівняно з аналогом збільшену на 3,9% достовірність класифікації кардіограм.

Графічна частина складається із 6 плакатів.

У економічному розділі доведено доцільність проведення науково-дослідної роботи за даною темою. Термін окупності становить 2,7 р., що свідчить про комерційну привабливість науково-технічної розробки.

Ключові слова: класифікація, кардіограма, спайкінгова нейронна мережа, аритмія.

ABSTRACT

Zavalnyuk Ya.E Information technology for cardiogram classification based on spiking neural network. Master's thesis in the specialty 122 - «Computer sciences», educational program – «Artificial intelligence systems». Vinnytsia: VNTU, 2024. 92 p.

In Ukrainian language. Bibliogr .: 33 titles; fig . 12; table 9.

In this master's qualification work the choice of a spiking neural network for cardiogram classification was justified. Developed spiking neural network transforms the input cardiogram into a sequence of spikes, which are then recognized by the network, which has 4 input neurons, 18 output neurons, and a recurrent layer of 256 excitatory neurons and 64 inhibitory neurons, and can recognize 18 types of arrhythmia. Cardiogram classification software was created in the Python programming language using specialized NumPy and Matplotlib libraries. The software was trained and tested using the PhysioNet Arrhythmia Database (MIT/BIH) cardiogram data set. The training sample consisted of 15,773 cardiograms. The test sample consisted of 3955 cardiograms. The developed software has a reliability of cardiogram classification on the test sample of 96.3%, and the analogue program - 92.4%. Thus, compared to the analogue, the developed software has a 3.9% increase in the reliability of cardiogram classification.

The graphic part consists of 6 posters.

In the economic section, the expediency of conducting research work on this topic is proven. The payback period is 2.7 years, which indicates the commercial attractiveness of scientific and technical development.

Keywords: classification, cardiogram, spiking neural network, arrhythmia.

ЗМІСТ

ВСТУП

Актуальність. Серцеву діяльність можна контролювати за допомогою електрокардіограми (ЕКГ), яка широко використовується для виявлення серцевих захворювань завдяки своїй неінвазивній природі. Кваліфікований кардіологи можуть виявити аномалії шляхом візуального огляду записів ЕКГ-сигналів. Однак, аритмії виникають періодично особливо на ранніх стадіях, і тому вони можуть бути пропущені при рутинних контрольних записах. Зараз популярні апаратні установки, які дозволяють при їх носінні здійснювати постійний моніторинг ЕКГ-сигналів.

Нейроморфні обчислення є перспективним альтернативним обчислювальним підходом до стандартних машин фон Неймана для розв'язання задач обробки сенсорних даних або класифікації. Нейроморфні системи емулюють біофізику нейробіологічних архітектур, емулюючи здатність мозку до масової паралельної обробки інформації. Обчислення та комунікація в цьому новому типі пристроїв співлокалізовані і базуються на гіпотезі про те, що спайкінгова активність нейронів є ключовим елементом обробки інформації в нейробіологічних системах. Сучасні нейроморфні обчислювальні системи, натхненні біологією, привертають значну увагу завдяки їхнім можливостям масової паралельної обробки інформації, низькій затримці та продуктивності з низьким енергоспоживанням.

У цій роботі пропонується рішення для виявлення серцевих аномалій на основі спайків, сумісне з принципом обробки інформації спайкінговою нейронною мережею, яка є масштабованою, працює в реальному часі та пропонує найсучасніші показники продуктивності та точності.

Зокрема, демонструється класифікація електрокардіограм (ЕКГ) за допомогою спайкінгової нейронної мережі (Spiking Neural Network - SNN).

Основні результати цієї роботи полягають в наступному:

- запропоновано метод кодування та стиснення сигналів ЕКГ у потік асинхронних цифрових подій (тобто спайків).

- продемонстровано, що стиснуті сигнали ЕКГ можуть бути правильно класифіковані після розширення розмірності, виконаного рекурентною SNN.

- продемонстровано високу точність класифікації після контрольованого навчання LIF нейронів (Leaky integrate-and-fire) на один з 18 класів. з великої кількості просторово-часових імпульсних патернів, що генеруються рекурентною SNN.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп’ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета і завдання досліджень. Метою магістерської кваліфікаційної роботи є підвищення достовірності класифікації кардіограм за рахунок використання спайкінгової нейронної мережі.

Для досягнення мети розробки необхідно виконати такі задачі:

− провести аналіз предметної області класифікації кардіограм;

− розглянути існуючі методи класифікації кардіограм та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;

− обгрунтувати вибір типу нейромережі;

− обгрунтувати вибір архітектури спайкінгової нейромережі;

− сформулювати стадії інформаційної технології, розробити структуру та алгоритм роботи програмного засобу;

− виконати програмну реалізацію запропонованої інформаційної технології;

− провести тестування програмного продукту та виконати аналіз отриманих результатів.

Об’єкт дослідження – процес комп’ютеризованої класифікації кардіограм на основі спайкінгової нейромережі.

Предмет дослідження – алгоритми та програмні засоби класифікації кардіограм на основі спайкінгової нейромережі та достовірність їх роботи..

Методи дослідження. У роботі використані наступні методи наукових досліджень: системного аналізу, теорії штучних нейронних мереж для реалізації інформаційної технології, методи математичної статистики для розробки процесу розв'язання задачі та обрахунків результатів експериментів із програмним засобом, об’єктно-орієнтованого програмування.

Наукова новизна одержаних результатів.

1. Набула подальшого розвитку інформаційна технологія класифікації кардіограм, яка відрізняється використанням спайкінгової нейронної мережі, що дозволило підвищити достовірність класифікації кардіограм.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення класифікації кардіограм.

Запропонована інформаційна технологія сприяє підвищенню достовірності класифікації кардіограм, зокрема:

- розроблено алгоритм роботи програмного забезпечення класифікації кардіограм;
- розроблено програмні засоби для класифікації кардіограм.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології класифікації кардіограм. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз класифікації кардіограм та методів підвищення достовірності програмних засобів класифікації кардіограм [1].

Апробація результатів роботи. Результати роботи були апробовані на Міжнародній науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2025)», Вінниця, 15 жовтня 2024 року - 15 червня 2025 року».

Публікації. За результатами досліджень опубліковано тези доповіді на науково-технічній конференції [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ КЛАСИФІКАЦІЇ КАРДІОГРАМ

1.1 Постановка задачі

Електрокардіографія (ЕКГ) – простий та інформативний метод функціональної діагностики, який заснований на реєстрації й аналізі електричних імпульсів, що виникають при роботі серця і їх графічного запису у вигляді зубців на спеціальну паперову плівку.

За допомогою ЕКГ можна діагностувати велику кількість серцево-судинних захворювань, так як на підставі отриманих даних можна судити не тільки про електричну активність серця, а й про структуру міокарда.

ЕКГ застосовується для діагностики:

1) аритмії (визначення частоти і регулярності серцевих скорочень);
2) інфаркту міокарда, ішемії міокарда (показує гостре або хронічне ушкодження міокарда;
3) ішемічної хвороби серця.

У цій роботі досліджується програмна реалізація класифікатора, заснованого на нейронній мережі [2] та здатного визначати вид серцевої патології по сигналам ЕКГ. Як дані для аналізу у роботі використовуються сигнали ЕКГ із бази даних аритмії PhysioNet (PhysioNet Arrhythmia Database), наданої Массачусетським технологічним інститутом (Massachusetts Institute of Technology) та лікарнею Бет Ісраель (Beth Israel Hospital). Коротко ця база даних називається MIT/BIH. У ній міститься близько 20000 записів ЕКГ.

Записи ЕКГ з набору даних MIT/BIH мають формат високої точності (high-fidelity). Ці сигнали оцифровуються зі швидкістю 360 відліків за секунду на канал з роздільною здатністю 11 біт у діапазоні 10 мВ.

Цільовим параметром є номер класу серцевої патології (аритмії).

Перелік класів аритмії представлено у табл 1.1.

Таблиця 1.1 Класи серцевої патології (аритмії).

№ п/п	Код	Класифікація захворювання (назва виду аритмії)
1.	N	Нормальне серцебиття (Normal beat)
2.	L	Блокада лівої ніжки пучка Гіса (Left bundle branch block beat)
3.	R	Блокада правої ніжки пучка Гіса (Right bundle branch block beat)
4.	B	Блокада пучка Гіса /неуточнена/ (Bundle branch block beat /unspecified/)
5.	A	Астенціальна екстрасистола (Atrial premature beat)
6.	a	Аберрантно проведена асоційована атомарна екстрасистола (Aberrated atrial premature beat)
7.	J	Вузловий екстрасистола (Nodal (junctional) premature beat)
8.	S	Надшлуночкова передчасна або ектопічна екстрасистола (Supraventricular premature or ectopic beat (atr. or nod.))
9.	V	Шлуночкова екстрасистола (Premature ventricular contraction)
10.	r	R-on-T передчасне скорочення шлуночків (R-on-T premature ventricular contraction)
11.	F	З'єднання шлуночкового та нормального скорочення (Fusion of ventricular and normal beat)
12.	e	Передсердний виснажливий поштовх (Atrial escape beat)
13.	j	Вузлове запізніле скорочення (Nodal (junctional) escape beat)
14.	n	Надшлуночкова екстрасистола (Supraventricular escape beat (atr. or nod.))
15.	E	Запізніле скорочення шлуночка (Ventricular escape beat)
16.	I	Мерехтіння шлуночків серця (Paced beat)
17.	f	Поєднання мерехтіння та нормального ритму (Fusion of paced and normal beat)
18.	Q	Некласифіковане серцебиття (Unclassifiable beat)

Потрібно розробити нейромережевий класифікатор [3,4], що буде за записом двохканальної ЕКГ визначати клас серцевої патології (аритмії).

1.2 Основні відомості про електрокардіограму

Серце є найважливішим з органів в людському організмі, без нормального функціонування якого є неможливою повноцінна робота всіх інших систем. Хвороби, пов'язані з порушеннями роботи серцево-судинної системи, є найпоширенішими на сьогоднішній день, і їх частота значно зросла за останні десятиліття. Це може бути викликано зниженням фізичної активності, збільшенням навантажень на нервову систему,

шкідливим впливом виробничих факторів і т. д. У зв'язку з цим багатьом людям необхідно проводити дослідження серця і серцево-судинної системи.

На щастя в сучасній кардіології існує безліч діагностичних можливостей, що дозволяють своєчасно виявити проблеми і відхилення в роботі серця. Серед методів діагностики можна виділити ЕКГ картування, холтерівське моніторування, ультразвукове дослідження серця, фонокардіографія і т. д., проте, самим поширеним є метод електрокардіографії (ЕКГ).

ЕКГ - електричний прояв скорочувальної активності серця, яка може бути записана у вигляді електрокардіограми за допомогою електродів, розташованих на грудях або кінцівках. Дуже важливою перевагою ЕКГ є те, що форма її хвиль може змінюватися при наявності серцево-судинних захворювань і різних патологій. Типовий вид електрокардіограми та параметри її елементів (амплітуда в мВ і тривалість в секундах) представлені на рисунку 1.1

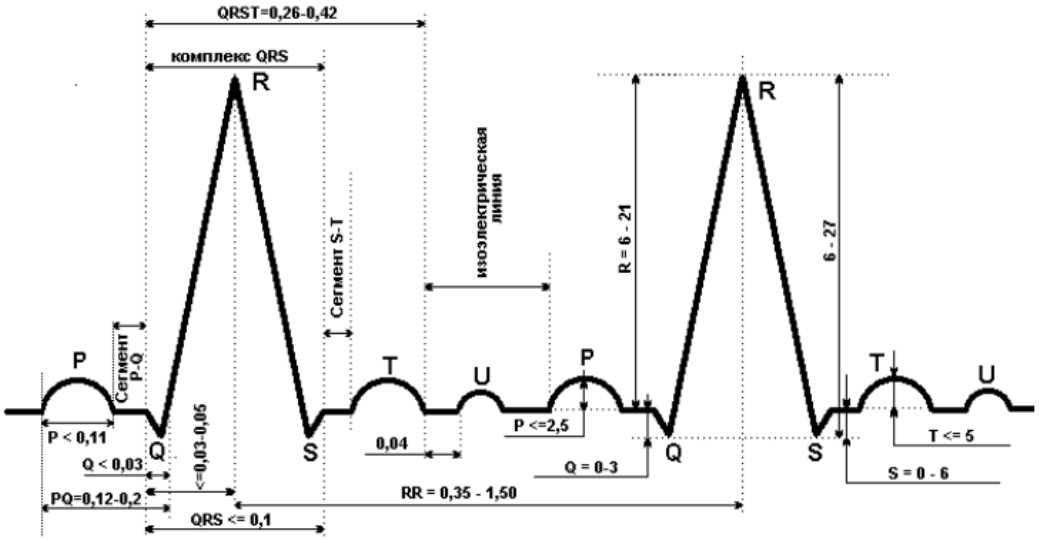


Рисунок 1.1 – Типовий вигляд ЕКГ

- далі стимулююча хвиля з великою швидкістю поширюється від верхівки серця, викликаючи швидке скорочення (Деполяризацію) шлуночків і тим самим появу QRS-комплексу – гострої двофазної або трифазної хвилі з амплітудою близько 1 мВ і тривалістю 80 мс;
- для м'язових клітин шлуночків характерна відносно велика тривалість потенціалу дії 300-350 мс. Плато на потенціалі дії викликає зазвичай ізоелектричний сегмент тривалістю 100-120 мс, наступний після QRS-комплексу і відомий як ST-сегмент;
- розслаблення шлуночків викликає поява повільного Т-зубця з амплітудою 0,1-0,3 мВ і тривалістю 120-160 мс;
- оскільки QRS-комплекс є самим вираженим елементом кардіограми, його амплітуда, тривалість і форма і інші параметри мають важливе діагностичне значення.

1.3 Огляд відомих методів класифікації кардіограм

Традиційні автоматизовані системи класифікації аритмій які використовують сигнали, виміряні з ЕКГ-пристрою, зазвичай базуються на наступних обчислювальних етапах: а) запис та збереження ЕКГ-сигналу, б) фільтрація та попередня обробка ЕКГ-сигналу, в) сегментація ЕКГ-сигналу, г) виокремлення ознак е) навчання та класифікація. На першому кроці використовується стандартний підхід фон Неймана, для отримання даних і збереження їх на зовнішньому блоці пам'яті (і споживання енергії при цьому).

Другий крок зазвичай виконується за допомогою вейвлет-перетворення, фільтрації Калмана або іншими стандартними алгоритмами обробки сигналів [5-7]. Сегментація серцевих скорочень базується на виявленні комплексу QRS сигналів ЕКГ [8, 9]. Виділення ознак

здійснюється різними методами, які включають фільтрацію, вбудовування та методи обгортання. Методи фільтрації ґрунтуються на статистичних властивостях сигналу і базуються на матриці кореляцій або лінійної декомпозиції з аналізом незалежних компонент [10]. Методи вбудовування та обгортання є більш обчислювально складними, ніж методи фільтрації, т. я. вони вимагають навченого класифікатора, а їх роль полягає в оптимізації вихідних даних з різноманітними наборами ознак.

Класифікацію ЕКГ-сигналів виконують:

- методами кластеризації на основі ознак [11, 12];
- методом опорних векторів (Support Vector Machine - SVM) [13, 14],
- нейромережевими методами [15, 16].

1.3.1 Методи кластеризації на основі ознак

Неконтрольована класифікація, так само відома як кластерний аналіз, має на меті згрупувати будь-який набір образів (тобто сигналів або випадків) в інформативні кластери, що не містять в явному вигляді апріорну інформацію про класифікацію цих образів. Основним результатом таких підходів є набір кластерів, які відносяться до групи векторів даних з чисельно близькими значеннями критерію схожості. Кластер може бути пов'язаний з великим числом класів, наприклад хворобами або групами ризику. В контексті ЕКГ кластери можуть включати групи сигналів, що представляють рекурентні QRS-комплекси і / або ST-сегменти.

Термін образ означає одиночний вектор даних, що складається з p елементів, x_{ij} = (xi_1, x_{i2},..., x_{ip}), який обробляється алгоритмом кластеризації. Окремі скалярні компоненти x_{ij} (j ∈ [1, p]) способу x_i називаються характеристиками. Звідси, маючи набір образів X = x1, x2,. . .

, xn ∈ Rp в p-вимірному просторі і грунтуючись на вимірюванні відстані або схожості, алгоритм кластеризації Q розділяє X на набір кластерів Qi, і = 1, 2,. . . , C, де c це загальне число кластерів, а n це розмірність вхідного набору даних. Типовий процес кластерної класифікації зображений на рисунку 1.2 і може бути описаний наступними кроками:

- представлення образів. Так само як визначення кількості доступних образів, а також кількість і тип характеристик, доступних алгоритму кластеризації. Витяг і вибір характеристик є основними завданнями цього кроку;

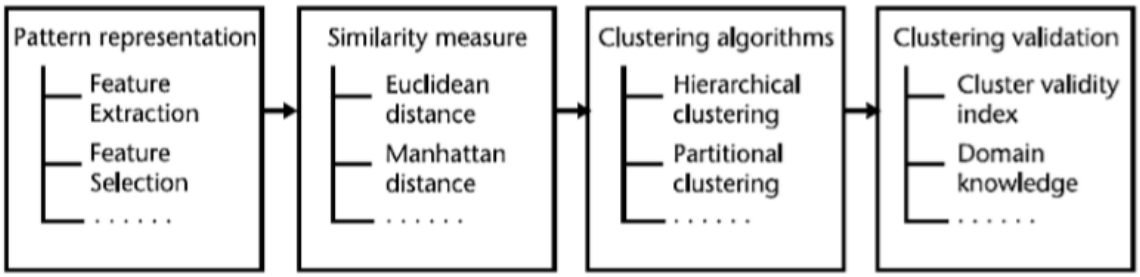


Рисунок 1.2 – Основні етапи процесу кластеризації

- вибір міри відстані або схожості. Визначення міри схожості між парами образів є першорядним етапом алгоритму кластеризації. Найбільш відома міра відстані це Евклідова відстань E = p (x2ip-x2kp) ½. Чим менший цей період, тим більше схожість;
- вибір алгоритму кластеризації. Традиційним прикладом такого роду алгоритмів є різні типи ієрархічної кластеризації, метод k-середніх і самоорганізуючі карти Кохонена (СОКК);
- перевірка відповідності кластерів.

Нейронні мережі у вигляді самоорганізованих карт, представлені Кохоненом, є одними з найвідоміших неконтрольованих нейронних мереж. Вони визначають перехід з M-мірного вхідного простору в вихідний простір нижчого порядку (зазвичай одно- або двовимірне), в

якому зберігаються вихідні ключові зв'язки між образами. Така модель характеризується скупченням вхідних образів у вигляді топографічної карти, на якій просторове розташування нейронів відображає схожість між групами образів.

Традиційний алгоритм СОКК наведено в таблиці 1.2. Відносна простота цього методу є його ключовою перевагою. Він не передбачає визначення цільової функції, тому не потрібні такі складні математичні операції як взяття похідних або обернення матриць.

Таблиця 1.2 – Алгоритм кластеризації з допомогою СОКК

№ кроку	Виконувані дії
1	Ініціалізація: Визначення топології мережі, вибір випадкового значення вагового вектора для кожного нейрона Конохена, установка параметра часу t = 0
2	Повторення
3	Вибір вхідного образу i_k з тренувального набору
4	Знаходження нейрона-переможця під час t, чий ваговій вектор wi найближче до i_k
5	Оновлення вагових вектором нейрона-переможця і його сусідів
6	Збільшення параметра часу на одиницю: t = t + 1
7	До тих пір, поки мережа не зійдеться в одній точці або поки не будуть перевищені обчислювальні обмеження, такі як заздалегідь задана кількість циклів навчання

У порівнянні з незмінною структурою ієрархічної кластеризації і недоліком структури в методі k-середніх, СОКК відображають відносини подібності між образами і кластерами шляхом адаптування їх нейронів, які використовуються для опису образів-прототипів. Однак, таке передвизначення топології карти призводить до її нездатності автоматично визначати межі кластерів.

Існує безліч методів для розширення можливостей візуалізації даних в СОКК. Один з них це побудова матриці відстаней (U-матриці), яка

описує відстані між сусідніми нейронами і зображується на карті у вигляді колірної схеми.

1.3.2 Метод опорних векторів

МОВ - це вид мережі з позитивною зворотною зв'язкою, запропонованої Вапником [17], і вже є найефективнішим інструментом у задачі класифікації. Цей метод має дуже хорошу здатність до узагальнення. На відміну від проблеми навчання класичних нейронних мереж, де мінімізована функція помилок нелінійна по відношенню до оптимізованих змінних безлічі потенційних мінімумів, МОВ приводить до квадратичного програмування з лінійними залежностями і може визначити чітко виражений глобальний мінімум. По суті, МОВ - це лінійний механізм, що працює в області характеристик високого дозволу, створеного шляхом нелінійного перетворення вихідного N-мірного вхідного вектору x в K-мірний простір характеристик (K> N) з використанням функції φ (x). Урівноваження гіперплощини, що розділяє два різних класи в N-мірному просторі, представлено нижче:

$$y(x) = w^T \varphi(x) + \omega_0 = \sum_{j=1}^{K} \omega_j \varphi_j(x) + \omega_0$$

де φ (x) = [φ1 (x), φ2 (x), ..., φK (x)] T, w це ваговий вектор мережі, w = [ω1, ω2, ..., ωK] T, а ω0 це похибка. Коли виконується y (x)> 0, вхідний вектор x відноситься до одного класу, а коли y (x) <0 - до другого. Всі математичні операції в режимі навчання і тестування виконуються з використанням так званої керн-функції K (xi, x), що задовольняє умовам Мерцера. Керн-функція визначається як внутрішнє скалярне приведення вектора $\varphi(x)$, $K(x) = \varphi^T(x_i)\varphi(x)$. Найбільш відома керн-функція - це лінійна,

Гауссіан, поліноміальна і сплайн-функція. Основним завданням навчання в МОВ є поділ тренувальних векторів xi на 2 класи, що визначаються значеннями, що приймаються di = 1 або di = -1, з максимальною роздільною здатністю. Далі виникає завдання максимізації квадратичної функції Q (x):

$$Q(\alpha) = \sum_{i=1}^{p} \alpha_i - 0.5 \sum_{i=1}^{p} \sum_{j=1}^{p} \alpha_i \alpha_j d_i d_j K(x_i, x_j)$$

при цьому:

$$\sum_{i=1}^{p} \alpha_i d_i$$
$$0 \le \alpha_i \le C$$

Змінні αi це коефіцієнти Лагранжа, di відносяться до значень, що приймаються, відповідних вхідних векторів xi, C - обумовлена користувачем константа регуляризації, а p це кількість пар учнів даних (xi, di). Вирішення цих двох завдань щодо коефіцієнтів Лагранжа дозволяє визначити оптимальний ваговий вектор wopt мережі МОВ

$$w_{opt} = \sum_{i=1}^{N_{sv}} \alpha_i d_i \varphi(x_i)$$

Nsv це кількість опорних векторів (векторів xi, для яких коефіцієнти Лагранжа відмінні від нуля). Підставивши, отримаємо вираз вихідного сигналу y (x) мережі МОВ як функції кернів

$$y(x) = \sum_{i=1}^{N_{sv}} \alpha_i d_i K(x_i, x) + \omega_0$$

Позитивне значення y (x) відповідає 1 (об'єкт належить до досліджуваного класу), а негативне значення -1 (об'єкт належить протилежного класу). Основний параметр МОВ це константа регуляризації C. Вона контролює співвідношення між шириною розділення кордону, що впливає на складність методу, і нерозділених точок на етапі навчання мережі. Маленьке значення C призводить до розширення розділення кордону і збільшення кількості нерозділені точок на в фазі навчання. Велике значення C дозволяє домогтися мінімальної кількості помилок класифікації, вузького розділення кордону і меншого числа опорних векторів. Занадто велике значення C призводить до втрати узагальнюючої здатності навченої мережі. Для нормалізованих вхідних сигналів зі значеннями в інтервалі (-1, 1) константа регуляризації зазвичай набагато більше 1 і визначається емпірично шляхом використання зразкового набору даних.

Важливим досягненням МОВ є перетворення завдання навчання в задачу квадратичного програмування. Для такого завдання оптимізації існує безліч ефективних алгоритмів навчання, які майже у всіх випадках призводять до глобального мінімуму цільової функції і вибору найкращого набору параметрів мережі. Іншим досягненням МОВ є хороша узагальнююча здатність, яка майже не залежить від кількості тренувальних зразків.

Хоча МОВ розділяє дані на 2 класи, розпізнати більшу кількість класів також нескладно шляхом застосування або методу «один проти одного» або методу «один проти всіх». Метод «один проти одного» зазвичай більш ефективний. У цьому методі незалежно навчається безліч локальних двокласових класифікаторів, що дозволяє вибрати найкращий класифікатор. Для M класів необхідно навчити M (M-1) / 2 двокласових класифікаторів, заснованій на МОВ системи розпізнавання.

1.3.3 Нейромережеві методи класифікації ЕКГ

Компоненти вхідного вектору x, що містять характеристики ЕКГ, описують вхід класифікатора. Нейронні класифікатори з навчанням вважаються одними з найбільш ефективних класифікаторів [18, 19].

У роботі [20] пропонують метод для виявлення QRS-комплексів та класифікації серцевих ритмів за допомогою адаптивної багатошарової персептронної мережі (MLP). Техніка виявлення QRS-комплексів використовує MLP, який підсилює QRS-комплекси та уникає нелінійного фонового шуму. Класифікатор працює з 12 класами.

Осовський та ін. [21] пропонують метод класифікації серцевих ритмів за допомогою гібридної нечіткої нейронної мережі. Використовуються кумулянти другого, третього та четвертого порядків. Класифікатор є комбінацією нечіткої самоорганізованої підмережі (з використанням алгоритмів k-means та Gustafson-Kessel) та багатошарової персептронної мережі. Вони досягають високої точності. Точність класифікації, представлена автором, коливається в межах 90% на семи різних типах вейвлетів.

У роботі [22] пропонують метод класифікації серцевих ритмів за допомогою ймовірнісної нейронної мережі (PNN). Піддіапазони сигналу ЕКГ, що перетворені за допомогою дискретного вейвлет-перетворення (DTW), разом з потужністю змінної складової та інтервалом RR використовуються як ознаки для класифікації.

Аль Раха та ін. [23] пропонують підхід для класифікації сигналів ЕКГ за допомогою глибокої нейронної мережі (DNN). Класифікація досягається за два кроки: (1) ознаки вилучаються із первинного сигналу ЕКГ за допомогою нагнітаючого автокодера зі змінною густотою з обмеженням розрідження, та (2) найбільш надійні та позначені сигнали ЕКГ використовуються для навчання класифікатора DNN.

Зубайр та ін. [24] пропонують метод класифікації ЕКГ за допомогою згорткових нейронних мереж. Модель вилучає приховані ознаки з первинного сигналу ЕКГ.

Раджпуркар та ін. [25] представляють алгоритм для виявлення аритмій серця за допомогою ЕКГ-сигналів. Метод використовує 34-шарову згорткову нейронну мережу (CNN) як класифікатор. Автор також пов'язує продуктивність класифікатора з більшою анотованою базою даних, в 500 разів більшою, ніж кількість, яка використовувалась в інших дослідженнях.

Нейронні мережі спроможні вирішувати завдання класифікації, набори даних у яких не є лінійно-роздільними. Крім цього, нейромережі відрізняються здатністю до апроксимації довільної функції з будь-якою бажаною точністю. Завдяки цьому застосування нейронних мереж для вирішення даної задачі класифікації ЕКГє найкращим варіантом.

1.4 Обгрунтування вибору аналогу до програмного забезпечення класифікації кардіограм

На теперішній час існує декілька відомих програмних додатків, що здатні здійснювати класифікацію кардіограм.

Як аналог візьмемо програмну реалізацію класифікатора кардіограм на основі машини опорних векторів (SVM). Для багатокласової класифікації: там використали розв'язувач, наданий відкритою бібліотекою libsvm [26], яка реалізує підхід "один проти одного" для багатокласової класифікації. Там побудували k(k -1)/2 класифікаторів (k = 18 класів), кожен з яких навчався на даних з двох класів. На етапі класифікації там використана стратегія голосування: кожна бінарна класифікація розглядається як голосування, де голоси віддаються за всі точки даних. Врешті-решт, точка відноситься до класу з максимальною кількістю голосів. Достовірність класифікації цієї системи складає 92,4 %.

До недоліків цієї програми можна віднести невисоку достовірність класифікації кардіограм. Інші програмні реалізації класифікації кардіограм також мають такий недолік як невисока достовірність. З огляду на це, постає завдання розробки нового програмного засобу для класифікації кардіограм з підвищеною достовірністю роботи.

1.5 Висновок до розділу 1

У розділі описано детальну постановку задачі класифікації кардіограм, проаналізовано основні відомості про кардіограми. Також розглядаються різні методи класифікації і оцінюється їх застосовність до завдання класифікації кардіограм. Серед таких методів як метод кластеризації на основі ознак регресії, метод машини опорних векторів та нейромережевий метод як найбільш перспективний і застосовний до даної задачі було обрано нейромережевий метод. Крім цього, було обгрунтовано вибір аналогу до програми класифікації кардіограм

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ КЛАСИФІКАЦІЇ КАРДІОГРАМ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОМЕРЕЖІ

2.1 Обґрунтування вибору типу нейронної мережі для інформаційної технології класифікації кардіограм

При виборі типу нейронної мережі для інформаційної технології класифікації кардіограм зупинимося на наступних моделях: багатошаровий персептрон [18] та спайкінгова нейронна мережа [3,4].

2.1.1 Багатошаровий перцептрон

Багатошаровий перцептрон [18, 19] це одна з найбільш відомих нейронних мереж, що складається з безлічі простих нейроноподібних модулів обробки сигмовидної функції активації, згруповані в шари. Функції активації в багатошаровому перцептроні можуть мати різну форму: логарифмічну f(x)=1/(1+exp(-x)), гіперболічний тангенс, сігнум (кусково-постійна функція) або лінійну. Класична мережа складається з одного прихованого шару, за яким слідує вихідний шар нейронів.

Інформація локально обробляється в кожному модулі шляхом обчислення скалярного добутку відповідного вхідного вектору і вагового вектору нейрона. Перед тренуванням вагові вектори створюються випадковим чином. Тренування мережі для отримання бажаного вихідного вектора di, коли відомий вхідний вектор xi, зазвичай включає систематичну зміну вагових векторів всіх нейронів до тих пір, поки мережа не видасть бажаний результат на виході, забезпечуючи при цьому допустиму похибку (помилку). Це повторюється протягом усього циклу тренування. Таким чином, навчання зводить до мінімуму процес

вимірювання помилки для всього навчального набору даних протягом кінцевого числа циклів навчання для запобігання перенавчання [19].

Найбільш ефективні методи навчання засновані на градієнтних моделях. Градієнтні вектори в багатошаровій мережі розраховуються з використанням алгоритму зворотного поширення помилки [19]. У градієнтному методі навчання вагові вектору w підбираються від циклу до циклу відповідно до інформації градієнта функції помилки

$$w(k+1) = w(k) + \eta p(k),$$

де η - це константа навчання, що розраховується в кожному циклі, а p (k) - це спрямовує вектор мінімізації в k-му циклі. Після закінчення етапу тренування знайдені вагові вектори «заморожуються» і потім використовуються в тестовому режимі, в якому вхідний вектор x, оброблений мережею для утворення вихідного нейронного сигналу, відповідає за визначення класу. Зазвичай нейрон найбільшого вихідного сигналу відповідає обумовленому класу.

Узагальнення це фундаментальна властивість, що дозволяє визначити здатність нейронної мережі до розпізнавання образів, що не входять в тренувальний набір. Якщо кількість ваг мережі занадто висока або кількість тренувальних прикладів дуже мала, то буде величезна кількість мереж, що відповідають тренувальним даним, але лише деякі будуть точно описувати простір вірних рішень. Отже, більш переважно слабке узагальнення. Щоб підвищити ймовірність правильного узагальнення, необхідно мінімізувати кількість вільних параметрів (ваг). Однак це потрібно зробити так, щоб розмір мережі не зменшився до області, в якій вже не досягти бажаного результату. Більш того, необхідно

контролювати кількість циклів навчання, щоб уникнути надмірної відповідності моделі тренувальним даним (перенавчання мережі).

Багатошаровий персептрон та інші широко відомі парадигми штучних нейронних мереж орієнтовані на обробку статичних вхідних даних [18,19]. А при обробці динамічних вхідних даних (як, наприклад, при розпізнаванні ЕКГ сигналів) процес розбивається на дискретні кроки. Крім того, результат на виході мережі отримується не одразу, а після певної кількості ітерацій. Це не узгоджується з природою кардіограми, оскільки вона є динамічним сигналом , а тому для обробки на БШП її треба дискретизувати та квантувати. Часто немає часу чекати, поки обчислення зійдеться, результати потрібні негайно (миттєве обчислення) або в межах короткого інтервалу часу («реально-часове» обчислення). Все сказане свідчить про те, що БШП та інші статичні нейронні мережі не підходять для оптимального розпізнавання ЕКГ. Тому виникає необхідність застосування такої моделі нейронної мережі, яка здійснювала б обробку в реальному часі неперервних вхідних сигналів (ЕКГ). До таких моделей відносяться спайкінгові нейронні мережі, в яких інформативним параметром є момент виникнення імпульсу нейрона, а не статичне значення.

2.1.2 Спайкінгові нейронні мережі

Структурно-функціональна модель спайкінгової нейронної мережі [2-4] будується, на відміну від традиційних мереж, на принципах роботи динамічних систем в комбінації зі статистичною теорією навчання. Кількість і ваги синаптичних зв'язків кожного нейрона у такій мережі обираються на основі нейрофізіологічних досліджень (тобто за аналогією з біологічними нейромережами). Випадковість вибору нейронів, які зв'язані з будь-яким іншим нейроном у мережі, приводить до утворення

багатоконтурних зворотних зв'язків, тобто такі спайкінгові нейромережі є рекурентними.

На рис. 2.1 показана абстрактна модель спайкінгової нейромережі [3,4] у вигляді автомата з «плаваючими» станами. Як видно з назви, ця модель має деяку схожість з кінцевим автоматом, але більш універсальна, і її «плаваючий» високорозмірний аналоговий стан $\boldsymbol{x}(t)$ змінюється неперервно у часі. І хоч динаміка такого автомата дуже складна, немає необхідності визначати його передавальну функцію із цієї динаміки, оскільки можливе відновлення інформації, яка містилася в $\boldsymbol{x}(t)$, безпосередньо з поточного стану автомата, навіть якщо він спотворений будь-яким шумом.

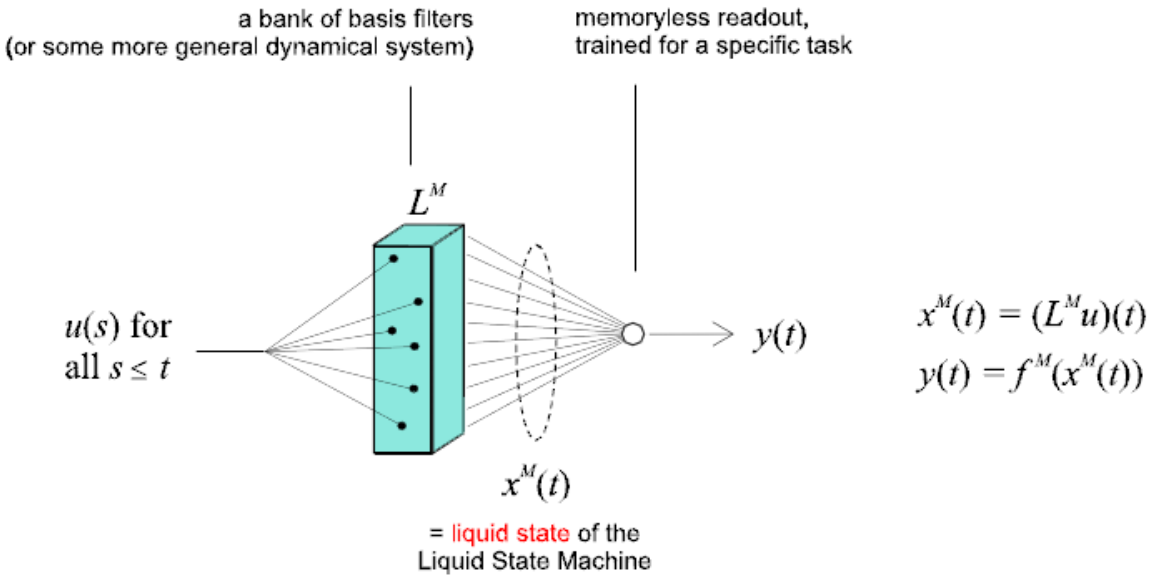


Рисунок 2.1 - Абстрактна модель спайкінгової нейронної мережі

Спайкінгові нейронні мережі, завдяки своїй схожості з мережами біологічних нейронів мають перед традиційними нейронними мережами такі переваги:

1) розпізнавання динамічних образів (мова, динамічні зображення та ін.; в разі розпізнавання мови як вхідні потоки можна використовувати, наприклад, вихідні сигнали смугових фільтрів);

2) багатозадачність (інформація про вхідні потоки циркулює в рекурентній нейронній мережі і на вихід одночасно можуть бути подані результати різних завдань за допомогою різних груп зчитувальних нейронів, навчених виконанню того або іншого завдання);

3) розпізнавання з передбаченням (будь-який динамічний процес може бути розпізнаний навіть за неповною інформацією про нього, тобто навіть раніше, ніж він завершиться). Це вельми важлива властивість, яка дозволяє підвищити швидкодію систем розпізнавання сигналів;

4) простота процедури навчання (навчаються не всі нейрони мережі, а лише вихідні зчитувальні нейрони);

5) підвищена продуктивність обробки інформації і завадостійкість завдяки частотно-імпульсному представленню інформації.

Формальна спайкінгова нейронна мережа (Spiking Neuron Network - SNN) складається з кінцевої множини V імпульсних нейронів, множини $E \subseteq V \times V$ синапсів, ваг $w_{u,v} \geq 0$ і функцій вихідного сигналів нейронів $\varepsilon_{u,v}$: $\mathbf{R}^+ \to \mathbf{R}$ для кожного синапсу $\langle u, v \rangle \in \boldsymbol{E}$ (де $\mathbf{R}^+ := \{x \in \mathbf{R} : x \geq 0\}$), і порогової функції Θ_v: $\mathbf{R}^+ \to \mathbf{R}^+$ для кожного нейрону $v \in V$.

Якщо $F_u \subseteq \mathbf{R}^+$ є множиною моментів часу видачі імпульсів нейрону u, тоді потенціал у тригерній зоні нейрона v у момент часу t являє собою наступне:

$$P_v(t) := \sum_{u:\langle u,v\rangle \in E} \sum_{s \in F_u : s<1} w_{u,v} \cdot \varepsilon_{u,v}(l-s) \qquad (2.1)$$

У моделі, яка не враховує шуми, нейрон *v* видає імпульс у момент часу *t* як тільки $P_v(t)$ досягне $\Theta_v(t - t')$, де де t' - це час найпізнішої видачі імпульсу нейроном *v*.

Для деякої визначеної підмножини $V_{input} \subseteq V$ вхідних нейронів припускається, що моменти часу пострілів («ряд імпульсів») F_u не визначені попередньо, але подаються ззовні. Моменти часу видачі імпульсів F_u для усіх інших нейронів $v \in V$ визначені за попередньо описаними правилами, і вихід мережі подається у вигляді ряду імпульсів F_u для нейронів *v* у визначеній множині вихідних нейронів $V_{output} \subseteq V$.

Експерименти показали, що біологічні нейрони видають імпульси з незначними затримками у відповіді на однакові ін'єкції струмом, що повторювались. Відомо, що лише за точних умов нейрони видають імпульси більш вірогідним чином. Тому розрізняють схоластичну або шумову версію моделі SNN, де різниця $P_v(t) - \Theta_v(t - t')$ лише регулює ймовірність того, що нейрон *v* видасть імпульс у момент часу *t*. Вибір моменту часу для пострілу припадає на деякий невідомий схоластичний процес, і може для прикладу трапитись, що *v* не видасть імпульс у інтервал часу *I*, протягом якого $P_v(t) - \Theta_v(t - t') > 0$, або видасть імпульс «спонтанно» у момент часу *t*, коли $P_v(t) - \Theta_v(t - t') < 0$. [2,3]

Найбільш відомим прикладом реалізації моделі I&F нейрону є Leaky Integrate-and-Fire нейрон (LIF – насичення та видача імпульсу з витоком) (Рисунок 2.2).

Відповідно рисунку 2.2 базова схема знаходиться у пунктирному колі..

Базова схема IF (integrate-and-fire) моделі складається з конденсатора та паралельного йому резистора. Загальний струм $I(t)$ може бути розділений на два компоненти, $I(t) = I_R + I_C$. Перший компонент – струм, I_R, що проходить через лінійний резистор *R*. За законом Ома він

буде рівний $I_R = u/R$, де u – напруга на резисторі. Другий компонент I_C струм конденсатора заряджає конденсатор C. З визначення ємності $C = q/u$ (де q це заряд, а u – напруга), можна знайти струм ємності $I_C = C\, du/dt$. Тобто струм $I(t)$ змінює RC схеми.

Таким чином.

$$I(t) = \frac{u(t)}{R} + C\frac{du}{dt} \tag{2.2}$$

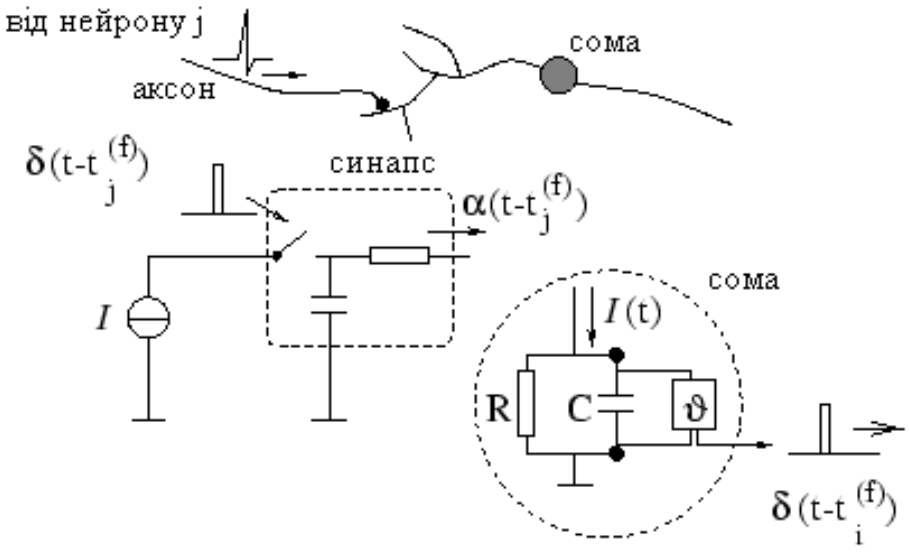


Рисунок 2.2 – Схематична діаграма I&F моделі

Помноживши рівняння на R та ввівши часову константу «накопичувача витоку» $T_m = RC$, отримаємо стандартну форму:

$$Tm\frac{du}{dt} = -u(t) + RI(t) \tag{2.3}$$

В даному випадку u є мембранним потенціалом, а T_m – часова константа мембрани.

Вираз (2.3) є лінійним диференційним рівнянням першого порядку і не може описати усю поведінку нейрона.

У IF-моделі форма діючого потенціалу точно не описана. Імпульси - це формальні події, що характеризуються «часом вистрілювання» (видачі імпульсу) $t^{(f)}$. Час $t^{(f)}$ визначається критерієм насичення.

$$t(f):\ u(t(f)) = \upsilon \tag{2.4}$$

Одразу ж після час $t^{(f)}$, потенціал встановлюється до значення u_r ,

$$\lim_{t \to t^f} u(t) = u_r \tag{2.5}$$

Для $t > t^{(f)}$ динаміка знову описується (2.3), поки не відбудеться наступне насичення. Напруга $u(t)$,що проходить через конденсатор (жирні крапки) порівнюється з пороговим значенням υ. Якщо $u(t) = \upsilon$ у момент часу $t_i^{(f)}$, то на виході генерується імпульс $\delta\ (t - t_i\ (f))$. У лівій частині рисунка 2.2 пресинаптичний імпульс $\delta\ (t - t_j(f))$ фільтрується у синапсі і генерується вхідний імпульсний струм $\alpha\ (t - t_j(f))$.

Комбінація накопичення витоку (2.3) та встановлення нового значення (2.4) визначає основу LIF – моделі. [2,4]

Таким чином, при використанні для розпізнавання ЕКГ сигналів традиційних нейронних мереж на формальних нейронах ЕКГ сигнали перетворюють в набір статичних векторів. При такому перетворенні втрачається частина інформації, а значить буде невисока точність розпізнавання схожих (сильно корельованих) образів, тобто необхідно підвищувати точність розпізнавання. Вирішити вказані вище проблеми можна за допомогою застосування спайкінгових нейронних мереж.

У цій роботі пропонується підхід, який сумісний з сучасними нейроморфними процесорами на основі спайкінгових нейромереж, які нещодавно були реалізовані як дослідницькі прототипи різними

компаніями та дослідницькими установами [3]. По суті, використовується спайкінгова нейронна мережа, яка працює з потоковими даними без необхідності використання буферів пам'яті для додаткових кроків фільтрації сигналу, без сегментації ЕКГ-сигналу і без розумного методу вилучення ознак. У цьому підході сигнали ЕКГ безпосередньо перетворюються в цифрові імпульси (тобто послідовності спайків). Тобто проблема класифікації сигналів ЕКГ перетворюється на проблему класифікації часових патернів спайків (часових рядів). Ця проблема все ще залишається основною в обчислювальних нейронауках та машинному навчанні. Вона може мати великий вплив на розробку вбудованих електронних систем [20]. Багато ключових механізмів вже доступні завдяки нещодавнім зусиллям, присвяченим розумінню ролі спайк-таймінгу в нейронній системі обробки інформації [2, 3]. системі обробки нейронної інформації [4, 7].

2.2 Структура процесів обробки інформації технології класифікації кардіограм

Структура інформаційної технології класифікації кардіограм проілюстрована на рис. 2.3. Архітектура повністю керована подіями і складається з трьох основних блоків.

1) Вхідні сигнали ЕКГ перетворюються в цифрові імпульси (тобто спайки) за допомогою двох дельта-модуляторів. Ці дельта-модулятори перетворюють сигнали ЕКГ в часові патерни спайків, як якщо б вони були вироблені чотирма спайкінговими вхідними нейронами (кола з плюсом і мінусом на рис. 2.3).

2) Рекурентна спайкінгова нейромережа SNN отримує вхідні сигнали від дельта модуляторів і виконує розширення розмірності входів.

3) Пул LIF-нейронів, що навчаються під контролем, для класифікації різних часових патернів, що генеруються рекурентною спайкінговою нейромережею. Імпульси від LIF-нейронів підраховуються і вибирається найактивніший клас.

Вхідні сигнали ЕКГ завантажуються з набору даних MIT/BIH. Два дельта-модулятори перетворюють сигнали в часові патерни спайків, кожне відведення ЕКГ кодується двома каналами ON (+) та OFF (-). Рекурентна архітектура SNN складається з двох щільних кластерів нейронів, з'єднаних між собою збуджувальними та гальмівними зв'язками. Вихідні імпульси з SNN надсилаються до вихідного шару LIF-нейронів. Ці LIF-нейрони виконують класифікацію патерну спайків, вибірково генеруючи імпульси, коли на вході присутні певні шаблони спайків.

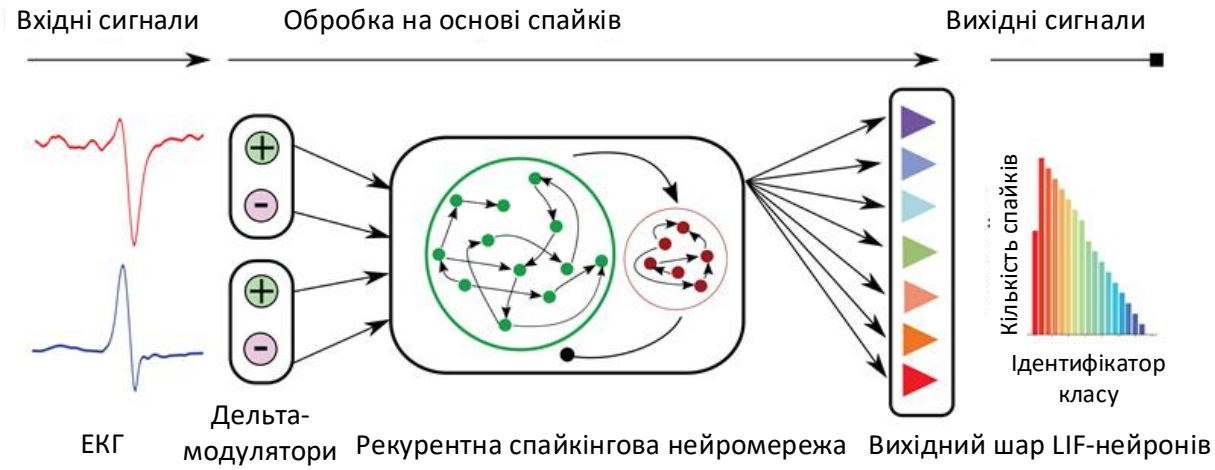


Рисунок 2.3 - Структура інформаційної технології класифікації кардіограм.

У наступних підрозділах буде детально пояснено всі ці блоки а також описано алгоритм навчання з учителем.

2.3 Стиснення та кодування сигналу ЕКГ у вигляді спайків

Роль дельта-модулятора полягає в стисненні двоканального сигналу ЕКГ і перетворенні цього сигналу в патерни спайків. Ці дельта-модулятори реалізовані в програмному забезпеченні і перетворюють високоточний ЕКГ-сигнал у чотириканальний спайкінговий вихід. Кожен канал ЕКГ кодується у два виходи (канали ON/OFF), цифрові події з кожного каналу позначають час, коли вхідний сигнал змінився більше (ON) або менше (OFF), ніж фіксований поріг. Кодування вхідного сигналу виконується для кожного вхідного каналу ЕКГ за алгоритмом по рис. 2.4.

```
Data: ECG single-lead trace
Result: UP;DN (two vectors of spikes)
dc = 0;
while input trace do
    read_trace_current ;
    current_time ;
    if read_trace_current > dc + δ then
        dc = read_trace_current;
        UP.append(current_time);
    end
    if read_trace_current < dc − δ then
        dc = read_trace_current;
        DN.append(current_time);
    end
end
```

Рисунок 2.4 - Алгоритм дельта-модулятора

Даний високороздільний ЕКГ-сигнал цей алгоритм перетворює у два вектори спайків (UP, DN).

Поріг δ являє собою інкрементну або декрементну зміну вхідного сигналу, яка викликає одиночний спайк. Маючи вхідний сигнал ЕКГ в нормалізованому діапазоні [0,1], встановлюємо поріг δ = 0.003. Нижчий поріг призводить до появи щільних спайків і навпаки. З такою

параметризацією комплекс ЕКГ загалом кодується приблизно 250 спайками. Часова точність зафіксована на рівні однієї мікросекунди.

Принцип роботи дельта-модуляторів по перетворенню вхідного сигналу ЕКГ у послідовність спайків проілюстровано на рис. 2.5.

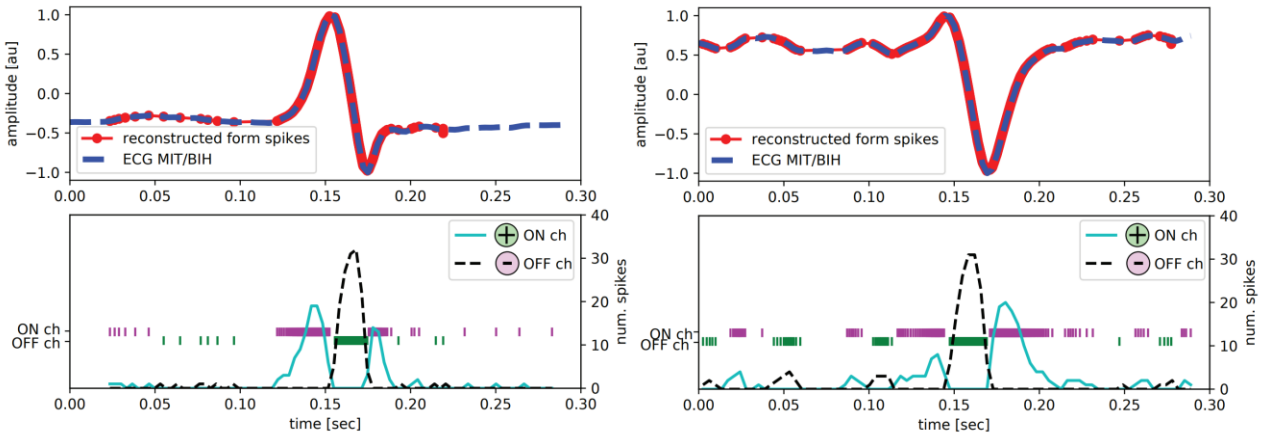


Рисунок 2.5 - Дельта-модулятор та вхідний сигнал ЕКГ

Вгорі на рис. 2.5 показано синім кольором ЕКГ з набору даних MIT/BIH, яка накладена на її реконструкцію зі спайків після стиснення, здійсненого дельта-модулятором. Знизу на рис. 2.5 показано вихід дельта-модулятора. Спайки показані вертикальними рисками (увімкнено – бузковими рисками / вимкнено – зеленими рисками). Середня кількість спайків показана суцільними та пунктирними кривими лініями. Відлік часу, що використовується для підрахунку кількості спайків, має розмір 170 мкс.

2.4 Архітектура рекурентної спайкінгової нейронної мережі

Рекурентна SNN складається з 320 спайкінгових нейронів, організованих у два щільні кластери (рис. 2.6). Один кластер містить 256 збудливих нейронів, тоді як другий кластер містить 64 гальмівних нейрони.

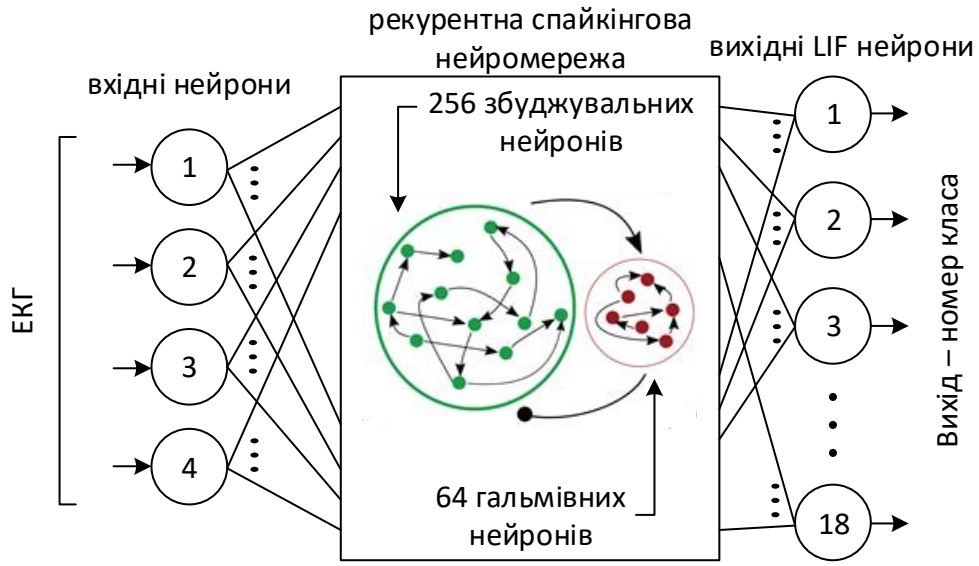


Рисунок 2.6 - Структура спайкінгової нейромережі для класифікації кардіограм.

На вхідний шар мережі (4 нейрони) подаються сигнали ЕКГ після дельта-модуляторів: ON (+) та OFF (-). Ці чотири вхідні нейрони генерують спайк-патерни. Вхідні нейрони мають фіксовані випадкові зв'язки з 20% синапсів збуджувального кластера нейронів. Ваги вхідних з'єднань беруться з Гаусівського розподілу з центром навколо нуля ($\mu = 0$) зі стандартним відхиленням $\sigma = 0.08$. Рівні зв'язності (щільності зв'язків) між збудливими (Exc) та гальмівними (Inh) пулами нейронів у SNN є наступними: Exc-Exc = 30%, Exc-Inh = 20%, Inh-Exc = 10%, Inh-Inh = 20%. Рівні зв'язності означають ймовірність того, що один нейрон має встановити зв'язок з будь-яким іншим нейроном у кластері призначення.

SNN була змодельована в програмному забезпеченні за допомогою спрощеної моделі нейронів типу "інтегрувати і вистрілювати" (Integrate-and-Fire, I&F). Крім того, SNN може бути реалізовано в апаратному нейроморфному процесорі [8] і отримано результати роботи в реальному часі. При цьому апаратні аналогові нейрони поводяться відповідно до біо-реалістичної адаптивної експоненціальної моделі LIF-нейрона.

Рекурентна SNN моделюється за допомогою спрощеної моделі I&F нейрона без витоку. Нейрон інтегрує передсинаптичні вхідні спайки, які передаються у вигляді цифрових імпульсів. Ці імпульси викликають миттєве зростання мембранного потенціалу нейрона (V_{mem}). Коли V_{mem} досягає порогового значення, генерується вихідний спайк, і мембранний потенціал скидається до постійного рівня (V_{rst}). Мембранний потенціал (V_{mem}) для нейронів у рекурентній SNN змінюється відповідно до рівняння:

$$V_{mem}^{i} = V_{rst} + \sum_j w_i \sum_j (t - t_{ij}), \tag{2.6}$$

Тут t позначає час, а w_i - набір синаптичних ваг. t_{ij} - час появи j-го спайку i-го пресинаптичного нейрона. Було задано $V_{rst} = 0$ і поріг $\varphi = 1$.

Синаптичні ваги для всіх збуджувальних зв'язків були отримані з гаусівського розподілу з $\mu = 0.04$ та $\sigma = 0.04$. У той час як гальмівні зв'язки також були отримані з гаусівського розподілу в протилежному напрямку ($\mu = -0.04$, $\sigma = -0.04$).

Всі нейрони рекурентного шару SNN під'єднані за принципом кожен з кожним до пулу вихідних LIF-нейронів (рис. 2.6), мембранний потенціал (V_{LIF}) яких описується наступним рівнянням:

$$V_{LIF}^{i} = V_{rst} + \sum_j w_i \sum_j k(t - t_{ij}), \tag{2.7}$$

в якому ми ввели член $k(t)$. Цей член являє собою експоненційну функцію ядра, яка визначається так:

$$k(t) = e^{\frac{-(t-t_i)^2}{\tau_e}}, \tag{2.8}$$

де t_i - час надходження спайку, а фіксовану сталу розпаду встановлено на рівні $\tau_e = 0.0003$.

2.5 Метод навчання класифікатора на основі вихідних LIF нейронів

Вихід рекурентної мережі був використаний для контрольованого навчання (за допомогою програмних симуляцій) пулу LIF-нейронів, які б вибірково спрацьовували, коли на вході з'являється патерн спайків відповідного класу.

Для того, щоб досягти такої класифікації з використанням LIF-нейронів, навчання проводилося за допомогою стандартного методу SpikeProp [3]. Метод, який тут використовується, вперше було представлено в [9], тому тут опишемо деякі математичні кроки та деталі використання цього методу.

Розглянемо класи паттернів P^p, $p = 1,...,18$. Кожен патерн складається з вхідних спайків, які генеруються рекурентним пулом з 320 нейронів за час пред'явлення T. Нехай t_{ij} – час появи j-го спайку i-го нейрона. Ми можемо розглянути вплив пачки спайків на постсинаптичну ділянку після застосування постійної функції ядра $k(t)$. Іншими словами, в кожен момент часу t ми оцінюємо вектор $f(t) = (f_1(t), f_2(t),.., f_n(t))$, де $f_i(t) = \sum j\ k(t - t_{ij})$. Це означає, що паттерн спайків трансформується у набір точок $f_p = \{ f(t_l)\}$, які відбираються через фіксовані інтервали $t_l = \Delta_l$ (див. $f(t_l)$ на рис. 2.7). Метою класифікації є пошук гіперплощини $X(W,b)$, яка визначається як

$$W_1 f_1 + W_2 f_2 + ... + W_N f_N - \rho = 0, \tag{2.9}$$

і яка відокремлює принаймні одну точку від шаблону, що належить до класу P^q, від всіх шаблонів іншого класу P^h. Для розв'язання цієї задачі можна використати SVM. У нашій багатокласовій задачі використовується підхід "один проти одного", як це було запропоновано в [10].

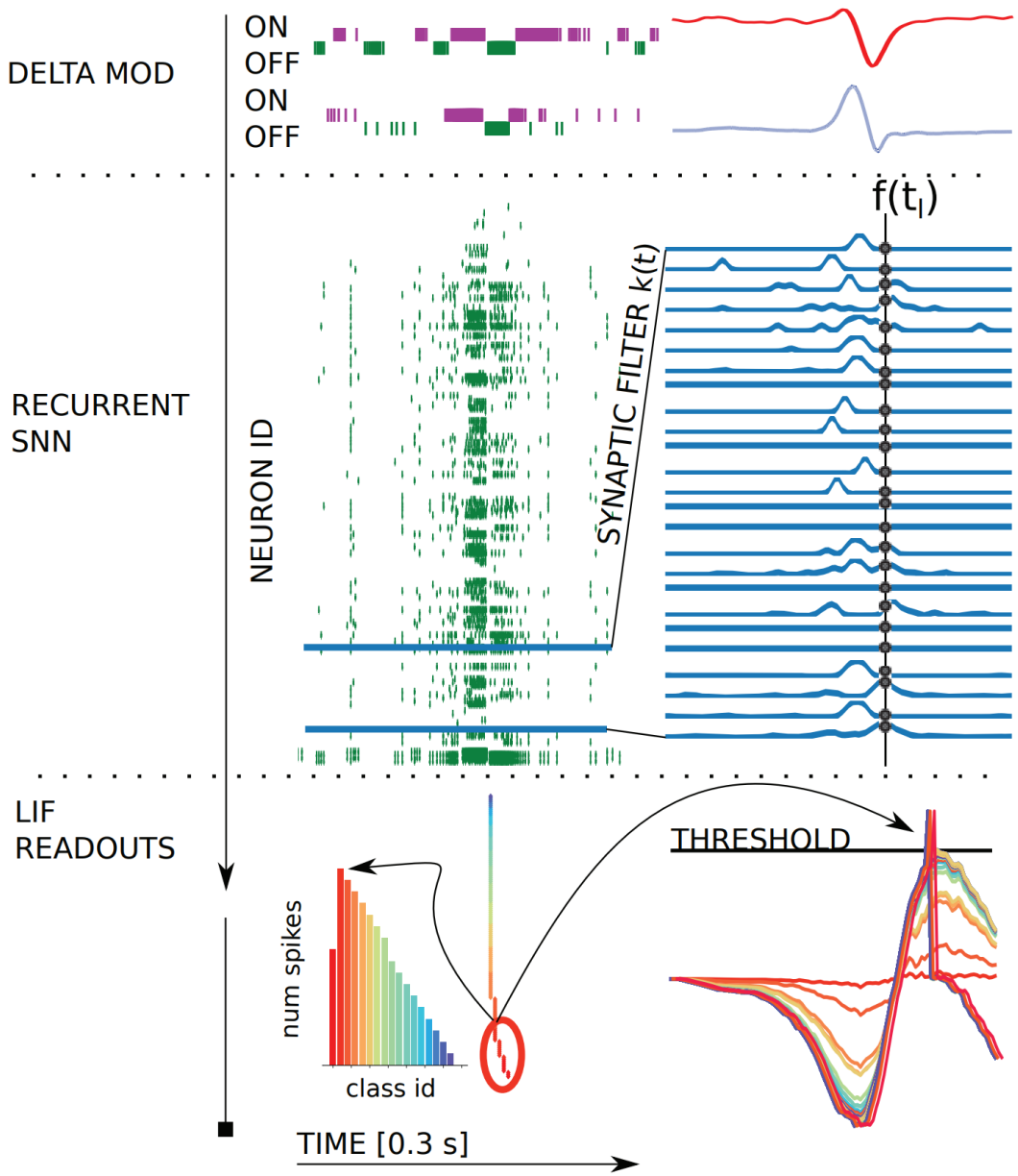


Рисунок 2.7 - Повний конвеєр обробки на основі спайків

На рис. 2.7 потік обробки йде зверху до низу. Вгорі – робота дельта-модуляторів). Вхідні сигнали ЕКГ перетворюються у спайки двома дельта-модуляторами. Посередині – рекурентна спайкінгова нейронна мережа із 320 нейронів, але показано растровий графік спайків для перших 180 нейронів рекурентної ШНМ із 320. Кожна рисочка представляє один спайк, випромінюваний нейроном. Горизонтальні лінії відокремлюють 31

нейрон. Сині сліди праворуч представляють вплив спайків цих 31 нейрона на вихідний LIF нейрон зчитування після застосування синаптичного ядра $k(t)$ з постійною часу $\tau_e = 3x10^{-5}$. Внизу – зчитувальні вихідні LIF нейрони. Растровий графік для 18 LIF-нейронів. Кольори кодують ідентифікатор LIF нейрона. Гістограма групує спайки, отримані для кожного класу, зібрані як «голоси за». Праворуч показано мембранний потенціал для всіх зчитувальних LIF нейронів. Поріг було визначено за допомогою еквівалентних SVM.

Навчання вихідного LIF-нейрона означає знаходження оптимальних синаптичних ваг. Синаптичні ваги для нейрона можна отримати, перетворивши коефіцієнти гіперплощини W відповідної SVM. Для цього потрібно лише помножити коефіцієнти W на φ/ρ, де φ - поріг спрацьовування LIF нейрона. Таким чином, мембранний потенціал LIF-нейронів описується наступним точковим добутком:

$$V_{LIF}(t)=(\varphi/\rho)W \cdot f(t_i), \tag{2.10}$$

Результат можна інтуїтивно описати як обмеження синаптичних ваг таким чином, щоб мембранний потенціал LIF-нейрона був нижче порогу спрацьовування для патернів, які не належать до бажаного класу, і вище порогу (принаймні на час t) для бажаного цільового класу (див. LIF-нейрони на рис. 2.7).

Стратегія "переможець отримує все" для всіх нейронів LIF може бути застосована як схема голосування. Цей процес забезпечує повну відповідність між класифікацією, яка виконується SVM, і шаром LIF-нейронів.

2.6 Розробка алгоритму роботи програмного забезпечення класифікації кардіограм

Відповідно до описаного у попередніх параграфах методу класифікації кардіограм на основі спайкінгової нейромережі, було розроблено алгоритм роботи програми класифікації кардіограм, який представлено на рис. 2.8.

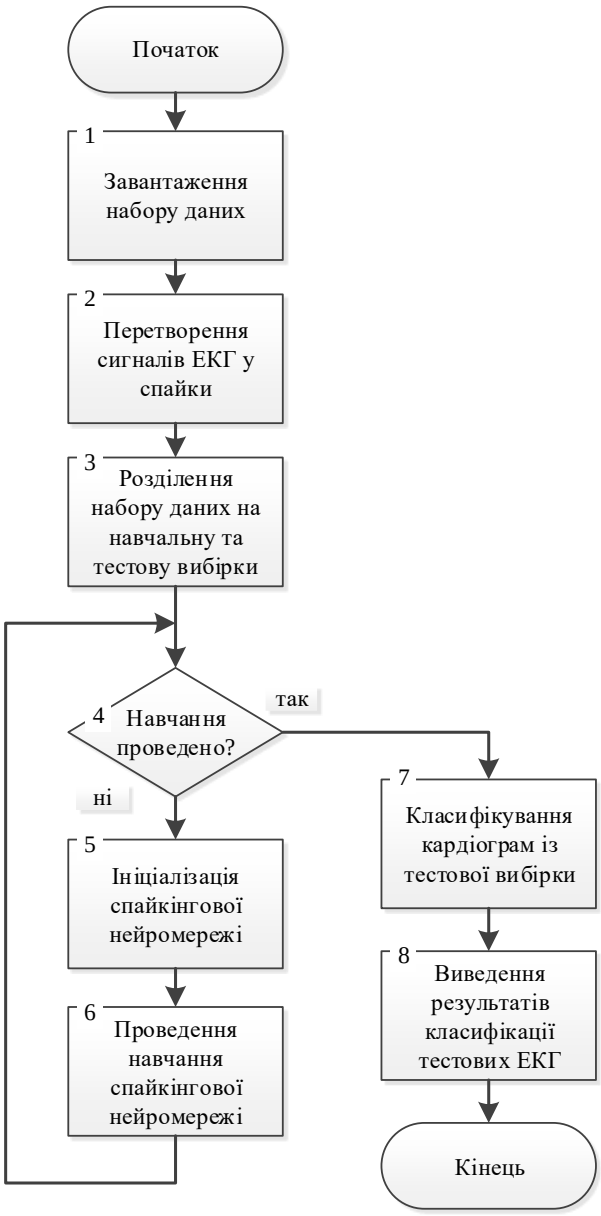


Рисунок 2.8 – Схема алгоритму роботи програмного забезпечення класифікації кардіограм на основі спайкінгової нейромережі

З отриманої схеми алгоритму можна розкласти роботу програми класифікації кардіограм на основі спайкінгової нейромережі на наступні ключові моменти:

- Завантаження набору даних (блок 1) – завантаження потрібних записів із набору даних PhysioNet Arrhythmia Database [27], які представлено у текстовому форматі .
- Перетворення сигналів ЕКГ у спайки (блок 2) – вхідні сигнали ЕКГ перетворюються в цифрові імпульси (тобто спайки) за допомогою дельта-модуляторів.
- Розділення набору даних на навчальну та тестову вибірки (блок 3) – в принципі таке розділення вже є у самому наюорі даних PhysioNet Arrhythmia Database [27], але треба зчитати ідентифікатори цих наборів.
- Навчання проведено? (блок 4) – перевірка проведення циклу навчання. Якщо навчання не проведено, то йдемо на блок 5, а якщо проведено, то йдемо на блок 7.
- Ініціалізація спайкінгової нейромережі (блок 5) – створюється спайкінгова нейронна мережа по структурі на рис. 2.6, формуються зв`язки між нейронами та ініціалізуються початковими значеннями коефіцієнти синаптичних зв`язків нейронів.
- Проведення навчання спайкінгової нейромережі (блок 6) – проводиться навчання спайкінгової нейромережі згідно методу навчання SpikeProp на основі навчальної вибірки.
- Класифікування кардіограм із тестової вибірки (блок 7) – здійснюється навченою спайкінговою нейронною мережею. Результати класифікування накопичуються і порівнюються з правильними діагнозами тестових кардіограм.

- Виведення результатів класифікації тестових ЕКГ (блок 8) – виводиться значення достовірності класифікації кардіограм тестової вибірки.

2.7 Висновок до розділу 2

У розділі було обґрунтовано вибір типу спайкінгової нейронної мережі, проведено розробку структури процесів обробки інформації інформаційної технології класифікації кардіограм. Було розроблено процес стиснення та кодування сигналу ЕКГ у вигляді спайків. Обрано архітектуру рекурентної спайкінгової нейронної мережі, яка має 4 вхідних нейрона, 18 вихідних нейронів і рекурентний шар із 256 збудливих нейронів та 64 гальмівних нейронів. Розглянуто метод навчання класифікатора на основі вихідних LIF нейронів. Розроблено алгоритм роботи програмного забезпечення класифікації кардіограм.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ КЛАСИФІКАЦІЇ КАРДІОГРАМ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОМЕРЕЖІ

3.1 Обґрунтування вибору мови програмування та спеціалізованих бібліотек

Для програмної реалізації інформаційної технології класифікації кардіограм на основі спайкінгової нейромережі після аналізу декількох мов програмування було обрано мову Python [28].

Python - інтерпретована, об'єктно-орієнтована, високого рівня мова програмування з динамічною семантикою. Розроблена у 1991 році Гвідо ван Россумом. Вона відома своєю простотою, елегантністю та читабельністю коду. Python популярний серед програмістів і використовується для розробки різноманітних програмних продуктів, веб-додатків, наукових обчислень, штучного інтелекту, машинного навчання та багатьох інших сфер. Вбудовані високого рівня структури даних у поєднанні з динамічною типізацією та зв'язуванням для розроблення додатків (RAD Rapid Application Development).

Крім того, її можна використовувати як сценарну мову для зв'язку програмних компонентів. Синтаксис Python простий у вивченні, в ньому надається особливе значення читабельності коду, а це скорочує витрати на супровід програмних продуктів. Python підтримує модулі та пакети, заохочуючи модульність і повторне використання коду.

Python підтримує об'єктно-орієнтоване програмування, функціональне програмування, а також імперативний та структурний стиль програмування. Мова Python має вбудовану підтримку роботи з текстом, регулярними виразами, обробкою файлів та мережевим програмуванням. Вона пропонує великий набір стандартних бібліотек, які

полегшують розробку і забезпечують широкий функціонал. Python є крос-платформовою мовою, програми написані на Python, можуть працювати на різних операційних системах, таких як Windows, macOS та Linux тощо.

Переваги мови Python:

1. Простота в освоєнні: Python має легкий для вивчення синтаксис та зрозумілість коду, що робить його ідеальним вибором.

2. Широкий спектр застосувань: Python може бути використаний у багатьох сферах, включаючи веб-розробку, наукові обчислення, штучний інтелект, машинне навчання та автоматизацію.

3. Простота синтаксису. Із синтаксису було прибрано все зайве, код чистий і зрозумілий без зайвих дужок і виразів.

4. Велика кількість бібліотек: Python має розширену екосистему сторонніх бібліотек, які допомагають вирішувати різноманітні задачі швидко та ефективно.

5. Інтерпретованість. Інтерпретатор Python існує для всіх популярних платформ і за замовчуванням входить до більшості дистрибутивів Linux, а значить є на більшості серверів.

6. Open Source - код інтерпретатора Python є відкритим, що дає змогу будь-кому, хто зацікавлений у розвитку мови, взяти участь у її розробці та поліпшити її.

7. Підтримка спільноти: Python має велику та активну спільноту розробників, яка надає підтримку, допомогу та ресурси для програмістів у вигляді документації, форумів та онлайн-курсів.

8. Крос-платформовість: Python підтримується на різних операційних системах, що дозволяє розробляти програми, які працюють на різних платформах.

Недоліки Python:

1. Швидкодія: Python є інтерпретованою мовою, що означає, що вона виконується повільніше порівняно з компільованими мовами, такими

як C++ або Java. Це може бути проблемою, особливо для завдань, які вимагають великої швидкодії, таких як обробка великих обсягів даних або високоефективні алгоритми.

2. Обмеження потоків: У Python є обмеження щодо паралельного виконання завдань через Global Interpreter Lock (GIL). GIL – це механізм, який дозволяє тільки одному потоку виконуватися одночасно в межах одного процесу Python. Це означає, що багатопотокові програми в Python можуть не використовувати повністю потужність багатоядерних систем.

3. Менша підтримка для мобільних платформ: Python не є найкращим вибором для розробки мобільних додатків. Хоча є деякі фреймворки, такі як Kivy або BeeWare, які дозволяють розробляти мобільні додатки на Python, вони не мають такої широкої підтримки та екосистеми, як у випадку мови програмування, таких як Java або Kotlin для Android, або Swift для iOS.

4. Обмежена підтримка для розробки вбудованих систем: Python не є найкращим вибором для розробки вбудованих систем або низькорівневих додатків. Через свою інтерпретовану природу та високорівневу абстракцію, вона має більші вимоги до ресурсів порівняно з мовами, такими як C або C++, що робить її менш придатною.

Використання Python.

Веб-розробка: Python використовується для створення веб-додатків і серверних додатків. Фреймворки, такі як Django і Flask, допомагають розробникам швидко створювати потужні та масштабовані вебдодатки.

Наукові обчислення і обробка даних: Python має багато бібліотек, таких як NumPy, Pandas, SciPy та Matplotlib, які сприяють вирішенню складних задач наукових обчислень, обробки даних, статистики та візуалізації.

Штучний інтелект та машинне навчання: Python є однією з основних мов програмування для розробки моделей штучного інтелекту, машинного

навчання та глибинного навчання. Бібліотеки, такі як TensorFlow, Keras, PyTorch і Scikit-learn, дозволяють створювати та навчати складні моделі.

Скриптована автоматизація: Python може бути використаний для написання скриптів, які автоматизують рутинні завдання, обробку файлів, роботу з базами даних, мережами тощо.

Інтернет-розробка та API: Python може бути використаний для створення розширень, плагінів, парсерів, скраперів та розробки API.

Ігрова розробка: Python використовується для створення ігор, як для веб, так і для настільних платформ. Фреймворки, такі як Pygame, допомагають розробникам швидко створювати ігрові додатки.

Таким чином, Python є чудовою мовою програмування. Він має широкий спектр застосувань, дозволяючи розробляти веб-додатки, наукові проекти, аналізувати дані та створювати штучний інтелект. Крім того, Python має велику кількість бібліотек та фреймворків, що полегшують розв'язання різноманітних задач. Він є портативним, підтримується на різних платформах і має зрілу та стабільну основу. Загалом, Python є потужним інструментом, який комбінує простоту з ефективністю і є вартим розгляду для різних проектів та рівнів досвіду програмістів.

Оскільки ми використовуємо нейронні мережі, а їх моделювання пов`язане з множенням векторів на матриці, то для цієї роботи знадобиться бібліотека NumPy [29]. NumPy (скорочення від Numerical Python) — це фреймворк з відкритим кодом для мови Python. Фреймворк має такі можливості:

- підтримка багатовимірних масивів (у тому числі матриці);
- підтримка високорівневих математичних функцій, які призначено для роботи з багатовимірними масивами.

Математичні алгоритми, які реалізовані інтерпретованими мовами (напр., Python), часто працюють набагато повільніше тих алгоритмів, що

реалізовані компільованими мовами (наприклад, Фортран, Сі, Java). Бібліотека NumPy надає реалізації різних обчислювальних алгоритмів (у вигляді функцій та операторів), оптимізованих для роботи з багатомірними масивами. У результаті будь-який алгоритм, що може бути виражений у вигляді послідовності операцій над матрицями (масивами) та реалізований з використанням NumPy, працює так само швидко, як еквівалентний код, що виконується у MATLAB.

Оскільки при проектуванні програми класифікації кардіограм на основі спайкінгової нейромережі ми виконуємо побудову різних діаграм та графіків, то нам стане у нагоді бібліотека Matplotlib [30].

Matplotlib — це комплексна бібліотека для створення статичних, анімованих та інтерактивних візуалізацій на Python. Matplotlib володіє такими функціональні можливості:

- Створення якісних сюжетів для публікації.
- Створення інтерактивних фігур, які можна масштабувати, панорамувати, оновлювати.
- Налаштування візуального стилю і макету.
- Експорт у багато форматів файлів.
- Інтегрування в JupyterLab і графічний інтерфейс користувача.
- Використання багатого набору сторонніх пакетів, побудованих на Matplotlib.

Matplotlib створює цифрової якості публікації в різноманітних форматах друкованих копій та інтерактивних середовищах на різних платформах. Matplotlib можна використовувати в сценаріях Python, оболонках Python/IPython, серверах веб-додатків і різноманітних наборах інструментів графічного інтерфейсу користувача.

3.2 Опис набору даних кардіограм для навчання та тестування програми

У роботі використовуються сигнали ЕКГ із бази даних аритмії PhysioNet (PhysioNet Arrhythmia Database), наданої Массачусетським технологічним інститутом (Massachusetts Institute of Technology) та лікарнею Бет Ісраель (Beth Israel Hospital). Коротко ця база даних називається MIT/BIH. У ній міститься близько 20000 (19728) записів ЕКГ. Основні показники набору даних MIT/BIH наведено у табл. 3.1.

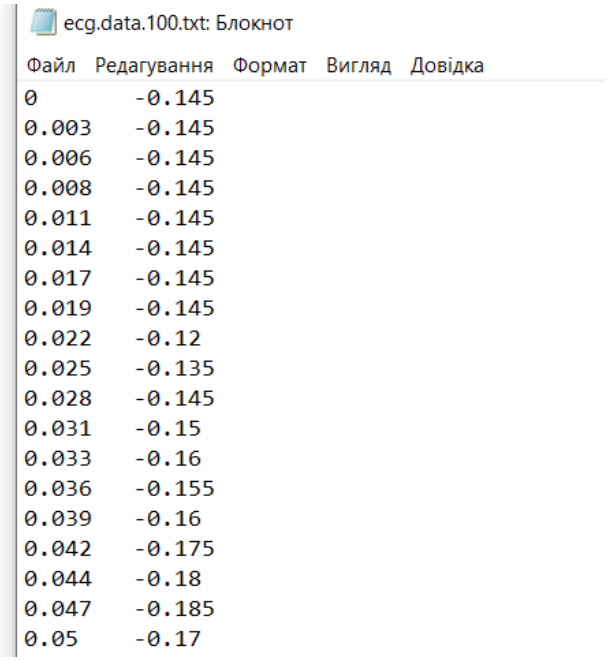
Таблиця 3.1 Кількість окремих комплексів ЕКГ для кожного класу аритмії.

№ п/п	Код	Класифікація захворювання (назва класу аритмії)	Число навчальних ЕКГ	Число тестових ЕКГ
1.	N	Нормальне серцебиття (Normal beat)	10882	2714
2.	L	Блокада лівої ніжки пучка Гіса (Left bundle branch block beat)	1050	266
3.	R	Блокада правої ніжки пучка Гіса (Right bundle branch block beat)	860	250
4.	B	Блокада пучка Гіса /неуточнена/ (Bundle branch block beat /unspecified/)	8	2
5.	A	Астенціальна екстрасистола (Atrial premature beat)	948	206
6.	a	Аберрантно проведена асоційована атомарна екстрасистола (Aberrated atrial premature beat)	70	8
7.	J	Вузловий екстрасистола (Nodal (junctional) premature beat)	28	4
8.	S	Надшлуночкова передчасна або ектопічна екстрасистола (Supraventricular premature or ectopic beat (atr. or nod.))	1128	300
9.	V	Шлуночкова екстрасистола (Premature ventricular contraction)	10	2
10.	r	R-on-T передчасне скорочення шлуночків (R-on-T premature ventricular contraction)	78	12

Таблиця 3.1 (Продовження)

№ п/п	Код	Класифікація захворювання (назва класу аритмії)	Число навчальних ЕКГ	Число тестових ЕКГ
11.	F	З'єднання шлуночкового та нормального скорочення (Fusion of ventricular and normal beat)	34	6
12.	e	Передсердний виснажливий поштовх (Atrial escape beat)	34	6
13.	j	Вузлове запізніле скорочення (Nodal (junctional) escape beat)	11	11
14.	n	Надшлуночкова екстрасистола (Supraventricular escape beat (atr. or nod.))	244	68
15.	E	Запізніле скорочення шлуночка (Ventricular escape beat)	154	22
16.	I	Мерехтіння шлуночків серця (Paced beat)	64	24
17.	f	Поєднання мерехтіння та нормального ритму (Fusion of paced and normal beat)	106	44
18.	Q	Некласифіковане серцебиття (Unclassifiable beat)	64	10
		Разом (із **19728**)	**15773**	**3955**

Вхідні сигнали ЕКГ завантажуються з набору даних MIT/BIH. Кожна кардіограма – це файл формату txt з переліком часових відліків кардіограми. Фрагмент файлу кардіограми представлено на рис. 3.1. Записи ЕКГ з набору даних MIT/BIH мають формат високої точності (high-fidelity). Ці сигнали оцифровуються зі швидкістю 360 відліків за секунду на канал з роздільною здатністю 11 біт у діапазоні 10 мВ. На рис. 2.5 показано зразок траси ЕКГ, яка була перетворена в піки за допомогою дельта-модулятора з втратами. Перевагами цього типу кодування є розрідженість піків, низькі вимоги до пропускної здатності та кодування «на вимогу» (коли вхідний сигнал не змінюється - вихідні спайки не генеруються).


ecg.data.100.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

```
0        -0.145
0.003    -0.145
0.006    -0.145
0.008    -0.145
0.011    -0.145
0.014    -0.145
0.017    -0.145
0.019    -0.145
0.022    -0.12
0.025    -0.135
0.028    -0.145
0.031    -0.15
0.033    -0.16
0.036    -0.155
0.039    -0.16
0.042    -0.175
0.044    -0.18
0.047    -0.185
0.05     -0.17
```

Рисунок 3.1 – Фрагмент txt файлу кардіограми з набору даних MIT/BIH

Із табл. 3.1 видно, що у набору даних розподіл кількості записів по видам аритмії є дуже незбалансованим: максимальна кількість кардіограм є для «Нормальне серцебиття» (10882+2714), а мінімальна кількість кардіограм є для «Блокада пучка Гіса /неуточнена» (8+2). І у самому наборі даних кардіограми вже розділені на навчальні та тестові.

3.3 Програмна реалізація класифікації кардіограм на основі спайкінгової нейромережі

Спайкінгова нейронна мережа, що застосовується у програмі класифікації кардіограм, має 18 вихідних нейронів (по кількості видів аритмії у табл. 3.1). Номер вихідного нейрона відповідає номеру класа аритмії у табл. 3.1. Інформація на виході нейромережі кодується кількістю імпульсів за період тривалості фрагменту кардіограми. Так, наприклад, якщо на вхід подається кардіограма, яка відноситься до 5-го класу аритмії у табл. 3.1, то максимальна кількість імпульсів буде зафіксована саме на 5-

му вихідному нейроні за період часу тривалості цієї кардіограми. На інших вихідних нейронах або будуть зафіксовані одиничні імпульси, або взагалі не буде зафіксовано імпульсів.

Програмні параметри LIF: вивчені вектори підтримки кодуються як синаптичні ваги зв'язків між кожним нейроном SNN з усіма нейронами зчитування LIF (загалом 153). Вектори підтримки - це точки в наборі даних, які є найближчими до меж прийняття рішень, деякі з них представляють приклади, які були неправильно класифіковані (тобто знаходяться не по той бік межі прийняття рішень) і за які в процесі навчання було призначено штрафні санкції. Таке представлення дозволяє отримати границі рішення шляхом простого обчислення точкового добутку між вхідними тестовими прикладами та ваговими зв'язками. У нашій реалізації зі спайками цей точковий добуток обчислюється між вихідними спайками рекурентної ШНМ після застосування синаптичного фільтра k(t), зваженого на синаптичні вхідні зв'язки для всіх нейронів, що зчитують LIF. Межі прийняття рішень - це порогові значення мембранного потенціалу LIF-нейронів; якщо LIF-нейрон отримує достатньо позитивних внесків, він досягне порогового значення і видасть спалах. Цей сплеск сигналізуватиме про те, що він голосує за обраний клас з-поміж двох, що належать до його компетенції. Оскільки набір даних MIT/BIH дуже незбалансований, кількість опорних векторів варіюється для кожного класу, тому кожен LIF нейрон має різну кількість вхідних синапсів. В результаті навчання було отримано 3745 опорних векторів. Кількість опорних векторів для кожного класу варіюється від максимального значення 1219 (%N) до мінімального 6 (%J). Кожен LIF-нейрон отримує вхідні дані через певну кількість синаптичних контактів, яка залежить від кількості опорних векторів. Наприклад, LIF-нейрон, роль якого полягає у прийнятті рішення між класами %N та %J, має синаптичне

дендритне дерево (вхідні зв'язки), що складається з 1219+6 = 1225 вхідних вагових векторів.

Першим кроком в програмі класифікації кардіограм на основі спайкінгової нейромережі буде завантаження бібліотек:

```
import sys
import numpy as np
import matplotlib.pyplot as plt

#add the python folder
sys.path.append('python/')
```

Далі працюємо з файлами із набору даних ЕКГ і перетворенням ЕКГ у спайки:

```
lv = levelcrossing.levelcrossing()
lv.load_data(do_plot=False, partial_load = True)
#load ecg data from folder, no plots, only first 500
points
all_id = lv.recordings['id']
lv.spikify(all_id, thr_up = 0.001, thr_dn = 0.001,
do_recontruction=True, interpfact=400000)
```

Перетворюємо фрагменти даних у спайки

```
all_id = lv.recordings['id']
```

```
lv.spikify(all_id, thr_up = 0.5, thr_dn = 0.5,
do_recontruction=True, interpfact=10000)
```

Далі виконуємо побудову графіків ЕКГ і графіків спайків, отриманих перетворенням ЕКГ у спайки:

```
plt.ion()
plt.subplot(2,1,1)
plt.plot(lv.recordings['reconstruction'][0][1],
lv.recordings['reconstruction'][0][0], 'bx-',
label='reconstructed form spikes')
plt.plot(lv.recordings['data'][0][:,0],
lv.recordings['data'][0][:,1], 'rx-',
label='original')
plt.ylabel('amplitude [au]')
plt.legend(loc='best')
plt.subplot(2,1,2)
plt.plot(lv.recordings['upch'][0],
np.repeat(1,len(lv.recordings['upch'][0])), 'mx',
marker='^',  label='up ch spikes')
plt.plot(lv.recordings['dnch'][0],
np.repeat(0.98,len(lv.recordings['dnch'][0])), 'gx',
marker='v', label='dn ch spikes')
plt.ylim([0.88, 1.1])
plt.ylabel('channel id')
plt.xlabel('time [sec]')
plt.legend(loc='best')
plt.show()
```

Фрагмент лістінгу коду програми класифікації кардіограм на основі спайкінгової нейромережі наведено у додатку Б.

3.4 Висновок до розділу 3

У розділі було обґрунтовано вибір мови програмування Python та спеціалізованих бібліотек NumPy та Matplotlib для програмної реалізації інформаційної технології класифікації кардіограм на основі спайкінгової нейронної мережі. Описано набір даних кардіограм PhysioNet Arrhythmia Database для навчання та тестування нейронної мережі, наданий Массачусетським технологічним інститутом (Massachusetts Institute of Technology) та лікарнею Бет Ісраель (Beth Israel Hospital), який містить близько 20000 записів ЕКГ. Описано програмну реалізацію інформаційної технології класифікації кардіограм на основі спайкінгової нейромережі.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ КЛАСИФІКАЦІЇ КАРДІОГРАМ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОМЕРЕЖІ

4.1 Тестування програми класифікації кардіограм на основі спайкінгової нейромережі

Результатом роботи є програмний застосунок, що здійснює класифікацію кардіограм. Для того щоб впевнитися, що програма працює правильно, проведемо її тестування. Після завантаження бібліотек та набору даних для навчання та тестування, програма виводить діаграму, що ілюструє процес перетворення сигналу кардіограми (кардіограма із набору даних MIT/BIH, вказана у коді програми) у двохканальну послідовність спайків. Дана діаграма представлена на рис. 4.1.

Вхідні сигнали ЕКГ (показані на рис. 4.1 вгорі синім кольором) перетворюються в цифрові імпульси (тобто спайки) за допомогою двох дельта-модуляторів. Знизу на рис. 4.1 показано вихід дельта-модуляторів. Ці дельта-модулятори перетворюють сигнали ЕКГ в часові патерни спайків, Кожен сигнал ЕКГ кодується двома каналами ON (+) та OFF (-). Канал спайків ON показано на рис. 4.1 знизу бузковими трикутниками (кожен трикутник відповідає одному спайку). Канал спайків OFF показано на рис. 4.1 знизу зеленими перевернутими трикутниками (кожен трикутник відповідає одному спайку).

Також вгорі на рис. 4.1 показано синім кольором реконструкцію ЕКГ зі спайків після стиснення, здійсненого дельта-модулятором, яка накладена на реальну ЕКГ (червоним кольором) з набору даних MIT/BIH.

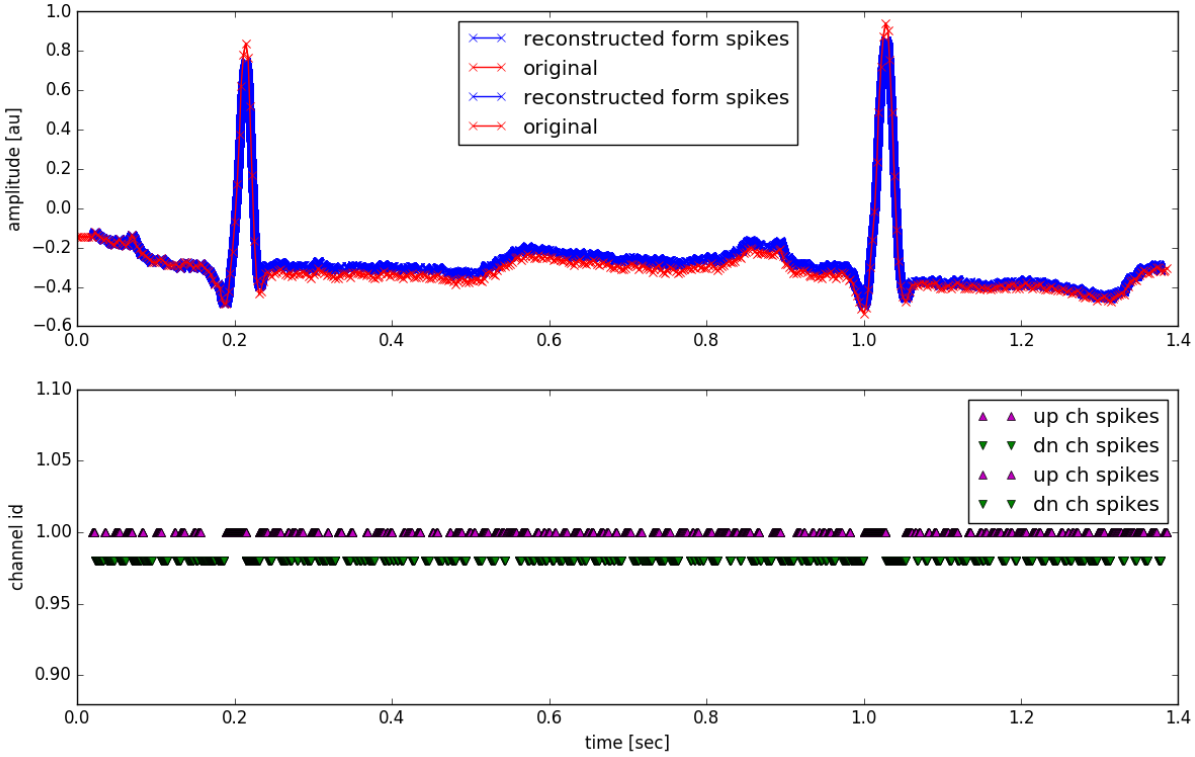


Рисунок 4.1 – Вихідне зображення програми класифікації кардіограм на основі спайкінгової нейромережі

Потім програма створює і проводить навчання спайкінгової нейромережі на навчальній вибірці.

4.2 Аналіз результатів роботи програми класифікації кардіограм на основі спайкінгової нейромережі

Після навчання проводиться тестування роботи спайкінгової нейромережі на тестовій вибірці. І програма виводить результат тестування у вигляді:

```
testing accuracy, avg / total:       0.9631   /   3955
```

Чисельні результати роботи розробленої програми, отримані в результаті тестування та наведені вище, занесено до табл. 4.1 для наочності порівняння різних методів класифікації кардіограм.

Таблиця 4.1 – Порівняння достовірності класифікації кардіограм розробленою програмою та програмою-аналогом

Програма	Кількість кардіограм у навчальній вибірці	Кількість кардіограм у тестовій вибірці	Час навчання, годин	Достовірність класифікації на тестовій вибірці
Аналог (на основі SVM)	15773	3955	8	92,4 %
Розрроблена програма на основі спайкінгової НМ	15773	3955	1,33	96,3%

Як аналог, ми взяли у п.1.3 програмну реалізацію класифікатора кардіограм на основі машини опорних векторів (SVM). Там використали розв'язувач, наданий відкритою бібліотекою libsvm [26]. Там для навчання та тестування використовувався, як і у нашому випадку, набір даних аритмії PhysioNet Arrhythmia Database (MIT/BIH).

Під час розбиття на навчальний та тестовий набори ми зберігали постійний розподіл прикладів у кожному класі, як і в наборі даних MIT/BIH (див. табл. 3.I). Саме через однаковість набору даних порівняння можна вважати адекватним. Достовірність класифікації цієї системи складає 92,4 %.

Як вказано в матеріалах аналога [26], його навчання вимагає близько 8 годин на 8-ядерному i7 стандартному настільному комп'ютері. Для навчання розробленої програми потрібно лише 1 година 20 хвилин навчального часу на 8-ядерному i7 стандартному настільному комп'ютері (див. табл. 4.1). Це є підтвердженням того, що рекурентна спайкінгова

нейромережа виконує цікаве розширення розмірності стиснених дельта-модуляторами сигналів ЕКГ.

Із табл. 4.1 видно, що розроблена програма має достовірність класифікації кардіограм на тестовій вибірці 96,3%, а програма-аналог має достовірність класифікації кардіограм на тестовій вибірці 92,4%.

Таким чином, можна зробити висновок, що розроблена програма має порівняно з аналогом збільшену на 3,9% достовірність класифікації кардіограм. Тобто мета роботи досягнута – достовірність класифікації кардіограм підвищена.

4.3 Висновок до розділу 4

У розділі в результаті тестування програми було доведено її повну працездатність та відповідність поставленому завданню. Навчання та тестування програми відбувалось з використанням набору даних аритмії PhysioNet Arrhythmia Database (MIT/BIH). Навчальна вибірка складалась із 15773 кардіограм. Тестова вибірка складалась із 3955 кардіограм. Навчання аналога вимагає близько 6 годин на 8-ядерному i7 стандартному настільному комп'ютері, а навчання розробленої програми - лише 1 годину 20 хвилин. Розроблена програма має достовірність класифікації кардіограм на тестовій вибірці 96,3%, а програма-аналог має достовірність класифікації кардіограм на тестовій вибірці 92,4%. Таким чином, розроблена програма має порівняно з аналогом збільшену на 3,9% достовірність класифікації кардіограм. Тобто мета роботи досягнута – достовірність класифікації кардіограм підвищена.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Проведення комерційного та технологічного аудиту інформаційної технології класифікації кардіограм на основі спайкінгової нейромережі.

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту [31,32] залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюємо із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 5.1 [32].

Таблиця 5.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології класифікації кардіограм на основі спайкінгової нейромережі.

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Технічна здійсненність концепції	3	2	3
Ринкові переваги (наявність аналогів)	2	3	2
Ринкові переваги (ціна продукту)	3	2	3
Ринкові переваги (технічні властивості)	3	2	3
Ринкові переваги (експлуатаційні витрати)	2	3	3
Ринкові перспективи (розмір ринку)	3	3	2
Ринкові перспективи (конкуренція)	2	3	3

Таблиця 5.1 (Продовження)

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Практична здійсненність (наявність фахівців)	3	2	3
Практична здійсненність (наявність фінансів)	2	3	3
Практична здійсненність (необхідність нових матеріалів)	3	3	3
Практична здійсненність (термін реалізації)	3	2	3
Практична здійсненність (розробка документів)	3	2	2
Сума балів	32	30	31
Середньоарифметична сума балів, СБ	31		

За результатами розрахунків, наведених в таблиці 5.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології класифікації кардіограм на основі спайкінгової нейромережі – середній [31].

5.2 Розрахунок витрат на здійснення науково-дослідної роботи

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

Основна заробітна плата дослідників. Витрати на основну заробітну плату [32] дослідників ($З_о$) розраховують відповідно до посадових окладів працівників, за формулою:

$$З_о = \sum_{i=1}^{к} \frac{М_{ni} \cdot t_i}{Т_р},$$

де к – кількість посад дослідників, залучених до процесу дослідження; $М_{ni}$ – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.; $Т_р$ – число робочих днів в місяці; приблизно $Т_р$ = (21…23) дні; t_i – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 5.2.

Таблиця 5.2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	20000	910	6	5460
Розробник	18000	818	12	9816
Консультанти	18000	818	7	5726
Всього:	21002			

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт розраховують за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i,$$

де $С_і$ – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду С і можна визначити за формулою:

$$С_і = \frac{М_м \cdot К_і \cdot К_с}{Т_р \cdot t_{зм}},$$

де $М_м$ – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2024 році $М_м$=8000 грн; $К_і$ – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду; $К_с$ – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; $Т_р$ – середня кількість робочих днів в місяці, приблизно $Т_р$ = 21…23 дні; $t_{зм}$ – тривалість зміни, год.

Таблиця 5.3 – Витрати на заробітну плату робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Аналіз предметної області	46	5	61,8	1,36	2842,8
Моделювання інформаційної технології	226	5	61,8	1,36	13966,8
Створення основної структури проєкту	36	6	65,9	1,45	2372,4
Створення та наповнення бази даних	158	6	65,9	1,45	10412,2
Тестування інформаційної технології	142	5	61,8	1,36	8775,6
Всього					36369,8

Додаткова заробітна плата. Додаткова заробітна плата $З_д$ всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$З_д = 0{,}11 \cdot (З_о + З_р) = 0{,}11 \cdot (21002 + 36369{,}8) = 6310 \text{ грн.}$$

Відрахування на соціальні заходи. Нарахування на заробітну плату $Н_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$Н_{зп} = \beta \cdot (З_о + З_р + З_д) =$$
$$= 0{,}22 \cdot (10501 + 18184{,}9 + 3155) = 14010 \text{ грн.}$$

де $З_о$ – основна заробітна плата розробників, грн.; $З_р$ – основна заробітна плата робітників, грн.; $З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Амортизація обладнання. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування А в цілому бути розраховані за формулою:

$$А = \frac{Ц_б}{Т_в} \cdot \frac{t}{12},$$

де $Ц_б$ – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; t – термін використання основного фонду, місяці; $Т_в$ – термін корисного використання основного фонду, роки.

Таблиця 5.4 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук	35000	2	2	2917
Стіл	5500	5	2	183
Стілець	4700	5	2	157
Мишка	1200	2	2	100
Клавіатура	2500	2	2	208
Настільна лампа	600	2	2	50
Всього	3615			

Витрати на електроенергію для науково-виробничих цілей. Витрати на силову електроенергію $В_е$, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Таблиця 5.5 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук	0,26	330
Освітлення	0,033	150

$$\text{Ве} = \sum \frac{W_i \cdot t_i \cdot \text{Ц}_\text{е} \cdot \text{К}_\text{впі}}{\text{ККД}}$$

$$= \frac{0{,}26 \cdot 330 \cdot 7{,}5 \cdot 0{,}85}{0{,}98} + \frac{0{,}033 \cdot 150 \cdot 7{,}5 \cdot 0{,}85}{0{,}98}$$

$$= 590 \text{ грн.},$$

W_i – встановлена потужність обладнання, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год.; Це – вартість 1 кВт електроенергії, грн.; $К_{впі}$ – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

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

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

$$\text{Ів} = \left(З_\text{о} + З_\text{р}\right) \cdot \frac{\text{Н}_\text{ів}}{100\%} = (21002 + 36369{,}8) \cdot \frac{100}{100} = 57372 \text{ грн.},$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

Накладні (загальновиробничі) витрати. До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов’язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

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

$$\text{Внзв} = \left(З_о + З_р\right) \cdot \frac{Н_{нзв}}{100\%} = (21002 + 36369{,}8) \cdot \frac{111}{100} = 63682 \text{ грн.},$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи. Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$В_{заг} = З_о + З_р + З_{дод} + З_н + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + \\ + І_в + В_{нзв} = \\ = 21002 + 36369{,}8 + 6310 + 14010 + 3615 + 590 + 57372 + \\ 63682 = 202951 \text{ грн.}$$

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

$$ЗВ = \frac{В_{заг}}{\eta} = \frac{202951}{0{,}9} = 225500 \text{ грн.},$$

де η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії технічного проектування, то η=0,9.

5.3 Розрахунок економічної ефективності науково-технічної розробки інформаційної технології класифікації кардіограм на основі спайкінгової нейромережі за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; $Ц_б$ – вартість послуги у році до впровадження інформаційної системи; $\pm\Delta Ц_о$ – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора ΔПі для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta П = (\pm\Delta Ц_о \cdot N + Ц_о \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right),$$

де ±ΔЦ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай,

таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm\Delta Ц_о$ може мати як додатне, так і від'ємне значення (від'ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни); N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки; $Ц_о$ – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році; $Ц_б$ – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів; ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки); λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість становить 20%, а коефіцієнт λ =0,8333; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати ρ =0,2...0,5; ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році ϑ =18%.

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\Delta П = \big((2600 - 1500) \cdot 1500 - (1500 - 2600) \cdot 2600\big) \cdot 0{,}8333 \cdot 0{,}2 \cdot \left(1 - \frac{18}{100}\right) = 616342 \text{ грн.}$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t} = \frac{616342}{(1+0{,}1)^0} = 377185 \text{ грн.},$$

де ΔП – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.; Т – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо Т=1 рік); τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ =0,05…0,15; t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot ЗВ = 2 \cdot 225500 = 451000 \text{ грн.}$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай кінв=2...5, але може бути і більшим; ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $Е_{абс}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$Е_{абс} = ПП - PV = 616342 - 451000 = 165342 \text{ грн.},$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.; PV – теперішня вартість початкових інвестицій, грн.

Оскільки $Е_{абс}>0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність Ев або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

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

$$Е_в = \sqrt[Т_ж]{1 + \frac{Еабс}{PV}} - 1 = \sqrt[1]{1 + \frac{165342}{451000}} - 1 = 0{,}37,$$

де $Т_ж$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування $\tau_{мін}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у

впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

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

$$\tau_{мін} = d + f = 0{,}12 + 0{,}05 = 0{,}17,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні d =0,9...0,12; f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина f =0,05...0,5, але може бути і значно вищою.

Оскільки $E_в$=0,37>$\tau_{мін}$=0,17, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій $T_о$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки інформаційної технології класифікації кардіограм на основі спайкінгової нейромережі:

$$T_о = \frac{1}{E_в} = \frac{1}{0{,}37} = 2{,}7 \text{ року.}$$

Оскільки $T_о$<1…5-х років, то це свідчить про комерційну привабливість науково-технічної розробки інформаційної технології класифікації кардіограм на основі спайкінгової нейромережі і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

5.4 Висновок до розділу 5

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія класифікації кардіограм на основі спайкінгової нейромережі» становить 31,0 балів, що вказує на комерційну вагомість проведення даних досліджень (рівень комерційного потенціалу розробки середній). Термін окупності становить 2,7 р., що менше 3-х років, що говорить про комерційну привабливість науково-технічної розробки і може спонукати потенційних інвесторів на фінансування впровадження цієї розробки та виведення її на ринок. Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія класифікації кардіограм на основі спайкінгової нейромережі».

ВИСНОВКИ

У роботі було розглянуто задачу класифікації кардіограм на основі спайкынгових нейромереж. В ході аналізу предметної області було описано детальну постановку задачі класифікації кардіограм, проаналізовано основні відомості про кардіограми. Також розглянуто різні методи класифікації і оцінено їх застосовність до завдання класифікації кардіограм. Серед таких методів як метод кластеризації на основі ознак, регресії, метод машини опорних векторів та нейромережевий метод, як найбільш перспективний і застосовний до даної задачі, було обрано нейромережевий метод. Крім цього, було обгрунтовано вибір аналогу до програми класифікації кардіограм.

Для побудови інформаційної технології було обґрунтовано вибір саме спайкінгової нейронної мережі, проведено розробку структури процесів обробки інформації в технології класифікації кардіограм. Було розроблено процес стиснення та кодування сигналу ЕКГ у вигляді спайків. Обрано архітектуру рекурентної спайкінгової нейронної мережі, яка має 4 вхідних нейрона, 18 вихідних нейронів і рекурентний шар із 256 збудливих нейронів та 64 гальмівних нейронів. Розглянуто метод навчання класифікатора на основі вихідних LIF нейронів. Розроблено алгоритм роботи програмних засобів класифікації кардіограм.

Для програмної реалізації інформаційної технології класифікації кардіограм на основі спайкінгової нейронної мережі було обґрунтовано вибір мови Python та спеціалізованих бібліотек NumPy та Matplotlib. Описано набір даних кардіограм PhysioNet Arrhythmia Database для навчання та тестування спайкінгової нейронної мережі, наданий Массачусетським технологічним інститутом (Massachusetts Institute of Technology) та лікарнею Бет Ісраель (Beth Israel Hospital), який містить

близько 20000 записів ЕКГ. Описано програмну реалізацію класифікації кардіограм на основі спайкінгової нейромережі.

В результаті тестування розробленої програми було доведено її повну працездатність та відповідність поставленому завданню. Навчання та тестування програми відбувалось з використанням набору даних аритмії PhysioNet Arrhythmia Database (MIT/BIH). Навчальна вибірка складалась із 15773 кардіограм. Тестова вибірка складалась із 3955 кардіограм. Навчання аналога вимагало близько 8 годин на 8-ядерному i7 стандартному настільному комп'ютері, а навчання розробленої програми - лише 1 годину 20 хвилин. Розроблена програма має достовірність класифікації кардіограм на тестовій вибірці 96,3%, а програма-аналог має достовірність класифікації кардіограм на тестовій вибірці 92,4%. Таким чином, розроблена програма має порівняно з аналогом збільшену на 3,9% достовірність класифікації кардіограм. Тобто мета роботи досягнута – достовірність класифікації кардіограм підвищена.

У економічній частині було визначено рівень комерційного потенціалу розробки за темою «Інформаційна технологія класифікації кардіограм на основі спайкінгової нейромережі», який становить 31,0 балів, що вказує на комерційну вагомість проведення даних досліджень (рівень комерційного потенціалу розробки середній). Термін окупності становить 2,7 р., що менше 3-х років, що говорить про комерційну привабливість науково-технічної розробки і може спонукати потенційних інвесторів на фінансування впровадження цієї розробки та виведення її на ринок. Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія класифікації кардіограм на основі спайкінгової нейромережі».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Я. Є. Завальнюк, О. К. Колесницький Інформаційна технологія класифікації кардіограм на основі спайкінгової нейромережі / в Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)», Вінниця, 2024, [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/22635/18745

2. Колесницький О. К. Методи і засоби розпізнавання сигналів мультисенсорів газів на основі імпульсних нейронних мереж: монографія / О. К. Колесницький. – Вінниця : ВНТУ, 2011. – 120 с. ISBN 978-966-641-407-9.

3. W. Maass, T. Natschläger, and H. Markram, “Real-time computing without stable states: A new framework for neural computation based on perturbations,” Neural computation, vol. 14, no. 11, pp. 2531–2560, 2002.

4. Колесницький О. К. Принципи побудови архітектури спайкових нейрокомп’ютерів / О. К. Колесницький // Вісник Вінницького політехнічного інституту. – Вінниця: УНІВЕРСУМ-Вінниця. – 2014. – №4 (115), С.70-78. [Електронний ресурс]. Режим доступу - https://visnyk.vntu.edu.ua/index.php/visnyk/article/view/911/910

5. C. Li, C. Zheng, and C. Tai, “Detection of ecg characteristic points using wavelet transforms,” IEEE Transactions on biomedical Engineering, vol. 42, no. 1, pp. 21–28, 1995.

6. M. Bahoura, M. Hassani, and M. Hubin, “Dsp implementation of wavelet transform for real time ecg wave forms detection and heart rate analysis,” Computer methods and programs in biomedicine, vol. 52, no. 1, pp. 35–44, 1997.

7. E. J. d. S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, “Ecg-based heartbeat classification for arrhythmia detection: A survey,” Computer methods and programs in biomedicine, vol. 127, pp. 144–164, 2016.

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

9. W. Zong, G. Moody, and D. Jiang, “A robust open-source algorithm to detect onset and duration of qrs complexes,” in Computers in Cardiology, 2003. IEEE, 2003, pp. 737–740.

10. J.-J. Wei, C.-J. Chang, N.-K. Chou, and G.-J. Jan, “Ecg data compression using truncated singular value decomposition,” IEEE Transactions on Information Technology in Biomedicine, vol. 5, no. 4, pp. 290–299, 2001.

11. P. De Chazal, M. O’Dwyer, and R. B. Reilly, “Automatic classification of heartbeats using ecg morphology and heartbeat interval features,” IEEE transactions on biomedical engineering, vol. 51, no. 7, pp. 1196–1206, 2004.

12. M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, and L. Sornmo, “Clustering ecg complexes using hermite functions and self-organizing maps,” IEEE Transactions on Biomedical Engineering, vol. 47, no. 7, pp. 838–848, 2000.

13. E. D. Übeyli, “Ecg beats classification using multiclass support vector machines with error correcting output codes,” Digital Signal Processing, vol. 17, no. 3, pp. 675–684, 2007.

14. F. Melgani and Y. Bazi, “Classification of electrocardiogram signals with support vector machines and particle swarm optimization,” IEEE transactions on information technology in biomedicine, vol. 12, no. 5, pp. 667–677, 2008.

15. S. Osowski and T. H. Linh, “Ecg beat recognition using fuzzy hybrid neural network,” IEEE Transactions on Biomedical Engineering, vol. 48, no. 11, pp. 1265–1271, 2001.

16. Z. Dokur and T. Ölmez, “Ecg beat classification by a novel hybrid neural network,” Computer methods and programs in biomedicine, vol. 66, no. 2-3, pp. 167–181, 2001.

17. Vapnik, V., Statistical Learning Theory, New York: Wiley, 1998.

18. Руденко О.В. Штучні нейронні мережі: Навчальний посібник / О.В.Руденко, Є.В.Бодянський. - Харків : ТОВ «Компанія СМІТ», 2006. — 404 с. - ISBN 966-8630-73-X.

19. Любунь З. М. Основи теорії нейромереж: текст лекцій / З. М. Любунь. – Львів : Видавничий центр ЛНУ ім. Івана Франка, 2006.

20. Y. H. Hu, W. J. Tompkins, J. L. Urrusti, and V. X. Afonso, Applications of artificial neural networks for ecg signal detection and classification. Journal of Electrocardiology 26, 66 (1993).

21. S. Osowski and T. H. Linh, Ecg beat recognition using fuzzy hybrid neural network. IEEE Transactions on Biomedical Engineering 48, 1265 (2001).

22. S.-N. Yu and Y.-H. Chen, Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network. Pattern Recognition Letters 28, 1142 (2007).

23. M. M. Al Rahhal, Y. Bazi, H. AlHichri, N. Alajlan, F. Melgani, and R. R. Yager, Deep learning approach for active classification of electrocardiogram signals. Information Sciences 345, 340 (2016).

24. M. Zubair, J. Kim, and C. Yoon, An automated ecg beat classification system using convolutional neural networks, 2016 6th International Conference on IT Convergence and Security (ICITCS), IEEE (2016), pp. 1–5.

25. P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv preprint arXiv:1707.01836 (2017)

26. C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu. edu.tw/~cjlin/libsvm.

27. MIT-BIH Arrhythmia Database [Електронний ресурс]. Режим доступу - https://www.physionet.org/content/mitdb/1.0.0/

28. Python [Електронний ресурс] – режим доступу: https://uk.wikipedia.org/wiki/Python

29. NumPy [Електронний ресурс] – Режим доступу: https://numpy.org/

30. Matplotlib: Visualization with Python [Електронний ресурс] – Режим доступу: https://matplotlib.org/

31. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

32. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

33. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп’ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ, 2023. – (58 с.)

Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія класифікації кардіограм на основі спайкінгової нейромережі

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Turnitin

Оригінальність 81,00% Схожість 19,00%

Аналіз звіту подібності (відмітити потрібне):

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

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

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

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

Автор роботи _____ Завальнюк Я.Є.

Керівник роботи _____ Колесницький О.К.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку _____ Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```
import sys
import numpy as np
import matplotlib.pyplot as plt


#add the python folder
sys.path.append('python/')


import levelcrossing                #deals with ecg files and conversion
to spike


lv = levelcrossing.levelcrossing()
lv.load_data(do_plot=False, partial_load = True)          #load ecg data
from folder, no plots, only first 500 points
all_id = lv.recordings['id']
lv.spikify(all_id, thr_up = 0.001, thr_dn = 0.001, do_recontruction=True,
interpfact=400000)


### do some plotting
plt.ion()
plt.subplot(2,1,1)
plt.plot(lv.recordings['reconstruction'][0][1],
lv.recordings['reconstruction'][0][0], 'bx-', label='reconstructed form
spikes')
plt.plot(lv.recordings['data'][0][:,0], lv.recordings['data'][0][:,1],
'rx-', label='original')
plt.ylabel('amplitude [au]')
plt.legend(loc='best')
plt.subplot(2,1,2)
plt.plot(lv.recordings['upch'][0],
np.repeat(1,len(lv.recordings['upch'][0])), 'mx', marker='^',  label='up
ch spikes')
plt.plot(lv.recordings['dnch'][0],
np.repeat(0.98,len(lv.recordings['dnch'][0])), 'gx', marker='v', label='dn
ch spikes')
plt.ylim([0.88, 1.1])
plt.ylabel('channel id')
plt.xlabel('time [sec]')
plt.legend(loc='best')
plt.show()
```

levelcrossing.py

```
from os import listdir
from os.path import isfile, join
import matplotlib.pyplot as plt
import numpy as np
import re
from scipy.interpolate import interp1d


#######################################################
# This class deals with the ECG data and labels
# also encodes ECG signals to spikes UP/DN channels
#######################################################

class levelcrossing:
    def __init__(self):
        self.recordings = {'id': [],                #recording/person id
                           'data': [],              #ecg signal
                           'rpeak': [],             #rpeak signal index
                           'ppeak': [],             #ppeak signal index
                           'qpeak': [],             #1peak signal index
                           'thrup': [],
                           'thrdn': [],
                           'upch': [],              # up ch adc
                           'dnch': [],              # dn ch adc
                           'reconstruction': []}    # from up/dn spikes
        self.n_data = 0;
        self.n_label = 0;

    def read_amplifier_dat_file(self, filepath, startf=0, endf=500,
do_plot = True):
        """
        This function reads the data from a .dat file created by Intan
        software and returns as a numpy array.

        Inputs:
            filepath: The path to the .dat file to be read.

        Outputs:
            amplifier_file: numpy array containing the data from the .dat
file (in uV)
        """

        with open(filepath, 'rb') as fid:
            raw_array = np.fromfile(fid, np.int16)
        amplifier_file = raw_array * 0.195 #converting from int16 to
microvolts
        datt = np.zeros([len(amplifier_file[startf:endf]),2]);
        datt[:,1] = amplifier_file[startf:endf]
```

```
        datt[:,0] =
np.linspace(0,len(amplifier_file[startf:endf]),len(amplifier_file[startf:e
ndf]))
        self.recordings['data'].append(datt)
        self.recordings['id'].append(1)

        self.n_data += 1;

        return amplifier_file

    def load_data(self, data_folder = 'data/', do_plot = False,
partial_load = False):
        '''
            Load all data from data_folder
        '''
        #find all ecg files in folder
        onlyfiles = [f for f in listdir(data_folder) if
isfile(join(data_folder, f)) and re.match(r'ecg*', f)]

        #load data
        for this_file in onlyfiles:
            id_rec = int(this_file.split('.')[2])
            try:
                self.recordings['id'].index(id_rec)
            except ValueError:
                print("adding recording id: " + str(id_rec))
                self.recordings['id'].append(id_rec)
            this_indx = self.recordings['id'].index(id_rec)
            if(partial_load):
                self.recordings['data'].append(np.loadtxt(data_folder +
this_file)[0:500])
            else:
                self.recordings['data'].append(np.loadtxt(data_folder +
this_file))
            if(do_plot):
                plt.figure()
                plt.hold(True)
                plt.title("raw_signal_with_peaks")

plt.plot(self.recordings['data'][this_indx][:,0],self.recordings['data'][t
his_indx][:,1])

            self.n_data += 1;

        #find all rpeak files in folder
        onlyfiles = [f for f in listdir(data_folder) if
isfile(join(data_folder, f)) and re.match(r'rpeak*', f)]

        for this_file in onlyfiles:
            id_rec = int(this_file.split('.')[2])
```

```
            try:
                self.recordings['id'].index(id_rec)
            except ValueError:
                print("adding label id" + id_rec)
                self.recordings['id'].append(id_rec)
            this_indx = self.recordings['id'].index(id_rec)
            self.recordings['rpeak'].append(np.loadtxt(data_folder +
this_file))

            if(do_plot):
                plt.title("lables_signal_with_peaks")
                index_r = self.recordings['rpeak'][this_indx].astype(int)
                plt.plot(self.recordings['data'][this_indx][index_r],

np.repeat(1,np.shape(self.recordings['rpeak'][this_indx])[0]), 'rx')
            self.n_label += 1;

        if(do_plot):
            plt.show()


    def spikify(self, id_to_spike, thr_up = 0.005, thr_dn = 0.005, do_plot
= False, do_recontruction = False, interpfact=10000):
        '''
            convert ecg signals to spikes (ideal asynch level crossing ADC
with 2 output channel)
               id_to_spike: ids of recordings to pass to adcs and create
up/dn spikes
        '''
        #convert to spikes
        for this_indx in id_to_spike:
            counter = 0;
            id_rec = self.recordings['id'].index(this_indx)
            actual_dc = self.recordings['data'][id_rec][0,1]   #first dc
is actual dc
            spike_up = []
            spike_dn = []
            #data = self.recordings['data'][id_rec]
            #interpolate ecg data
            f = interp1d(self.recordings['data'][0][:,0],
self.recordings['data'][0][:,1])
            rangeint = np.round((np.max(self.recordings['data'][0][:,0]) -
np.min(self.recordings['data'][0][:,0]))*interpfact)
            xnew = np.linspace(np.min(self.recordings['data'][0][:,0]),
np.max(self.recordings['data'][0][:,0]),
                               num=int(rangeint), endpoint=True)
            data = np.reshape([xnew, f(xnew)], (2, len(xnew))).T

            for i in range(len(data)):
                if( (actual_dc + thr_up) < data[i,1] ):
```

```
                    spike_up.append(data[i,0])  #spike up
                    actual_dc = data[i,1]       # update current dc value
                elif( (actual_dc - thr_dn) > data[i,1] ):
                    spike_dn.append(data[i,0])  #spike dn
                    actual_dc = data[i,1]       # update current dc value

            self.recordings['upch'].append(np.transpose(spike_up))  #add
to rec
            self.recordings['dnch'].append(np.transpose(spike_dn))

            if(do_plot):
                plt.figure()
                plt.title("id ", str(this_indx))
                plt.subplot(2,1,1)

plt.plot(self.recordings['data'][id_rec][:,0],self.recordings['data'][id_r
ec][:,1], 'bx', label='orig')
                plt.plot(data[:,0][0:10000], data[:,1][0:10000], 'gx-',
label='interp')
                plt.subplot(2,1,2)
                plt.plot(self.recordings['upch'][id_rec],
np.repeat(1,len(self.recordings['upch'][id_rec])), 'rx', label='up ch')
                plt.plot(self.recordings['dnch'][id_rec],
np.repeat(0.98,len(self.recordings['dnch'][id_rec])), 'gx', label='dn ch')

            if(do_recontruction == True):
                tot_spk =
len(self.recordings['upch'][id_rec])+len(self.recordings['dnch'][id_rec])
                current_val = self.recordings['data'][0][0][1] #first
spike
                signal_rec = []
                time_rec = []
                counter_up = 0
                counter_dn = 0
                for i in range(tot_spk):
                    if(self.recordings['upch'][id_rec][counter_up] <
self.recordings['dnch'][id_rec][counter_dn]):
                        current_val = current_val + thr_up
                        signal_rec.append(current_val)

time_rec.append(self.recordings['upch'][id_rec][counter_up])
                        if(counter_up <
len(self.recordings['upch'][id_rec])-1):
                            counter_up += 1
                    elif(self.recordings['upch'][id_rec][counter_up] >
self.recordings['dnch'][id_rec][counter_dn]):
                        current_val = current_val - thr_dn
                        signal_rec.append(current_val)

time_rec.append(self.recordings['dnch'][id_rec][counter_dn])
```

```
                        if(counter_dn <
len(self.recordings['dnch'][id_rec])-1):
                            counter_dn +=1

                self.recordings['reconstruction'].append([signal_rec,
time_rec])

        if(do_plot):
            plt.show()
```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЛАСИФІКАЦІЇ
КАРДІОГРАМ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОМЕРЕЖІ

Виконав: студент 2-го курсу,
групи 1КН-23м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
_____ Завальнюк Я. Є.
(прізвище та ініціали)

Керівник: к.т.н., професор каф. КН
_____ Колесницький О.К.
(прізвище та ініціали)
«_05_»_____12_____2024 р.

Вінниця ВНТУ - 2024 рік

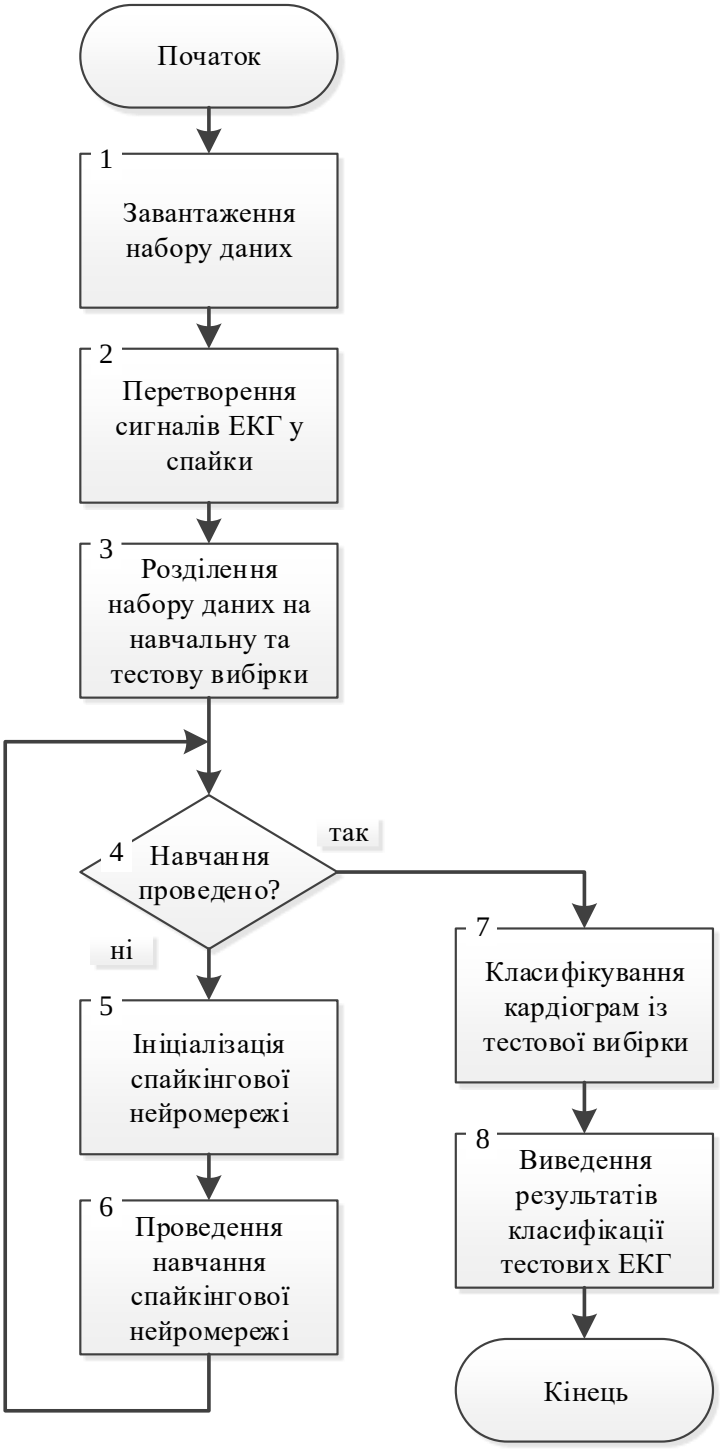


Рисунок В.1– Схема алгоритму роботи програмного забезпечення класифікації кардіограм на основі спайкінгової нейромережі

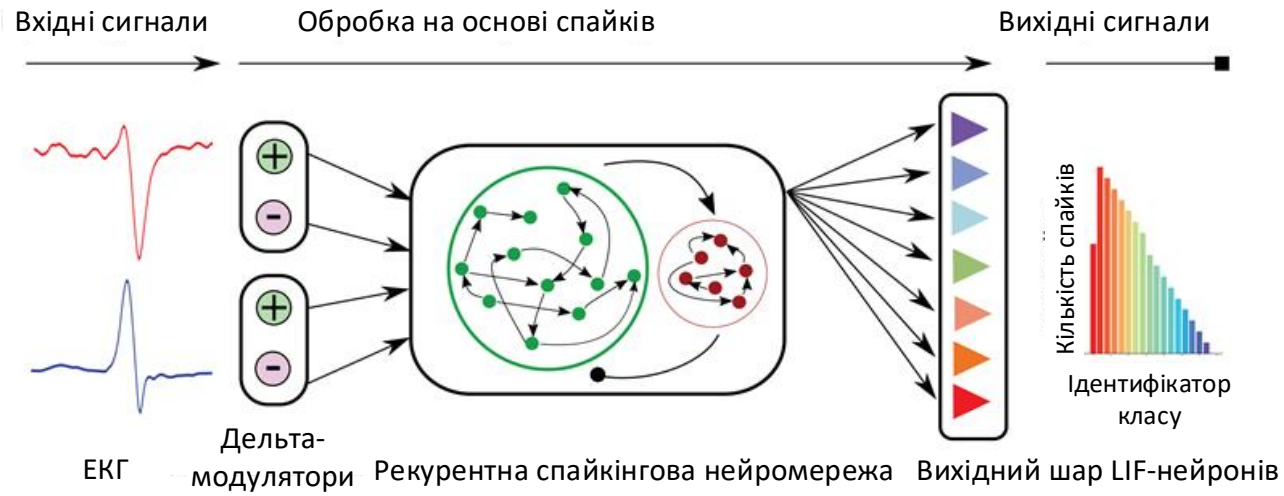


Рисунок В.2– Структура інформаційної технології класифікації кардіограм

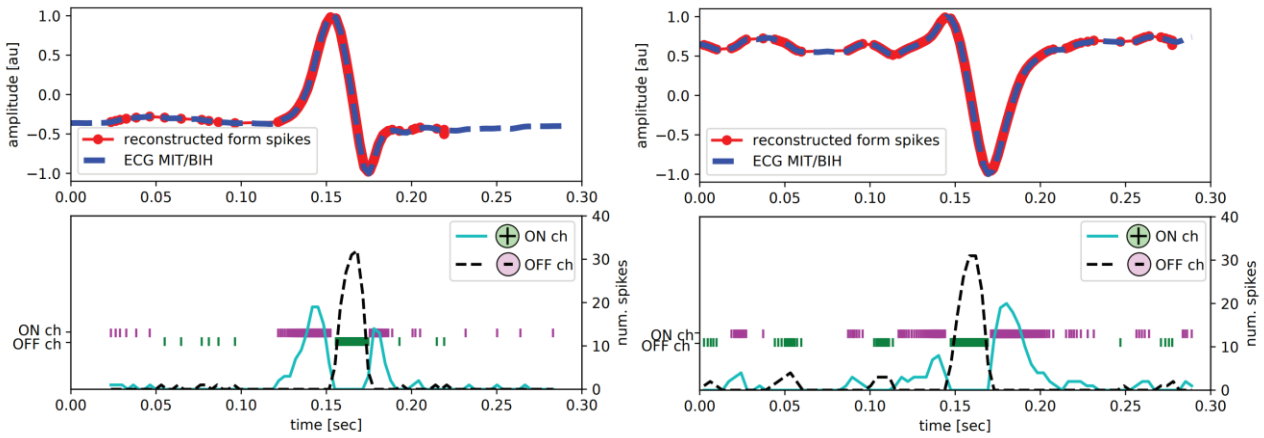


Рисунок В.3– Дельта-модулятор та вхідний сигнал ЕКГ

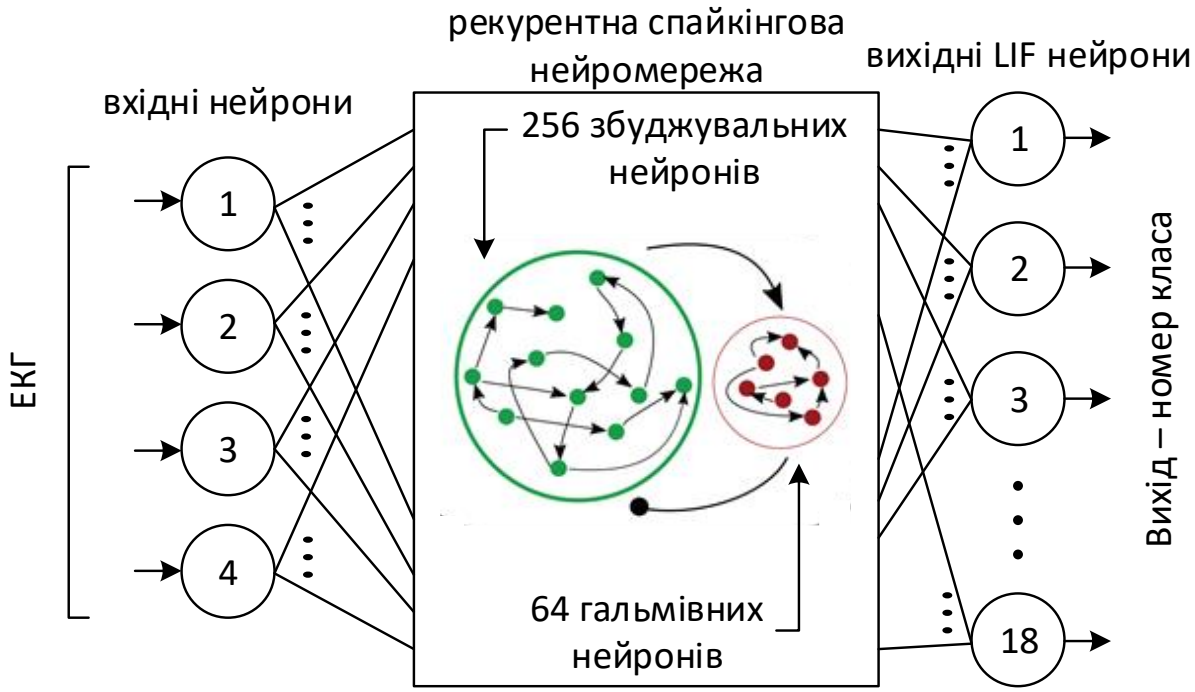


Рисунок В.4 – Структура спайкінгової нейромережі для класифікації кардіограм

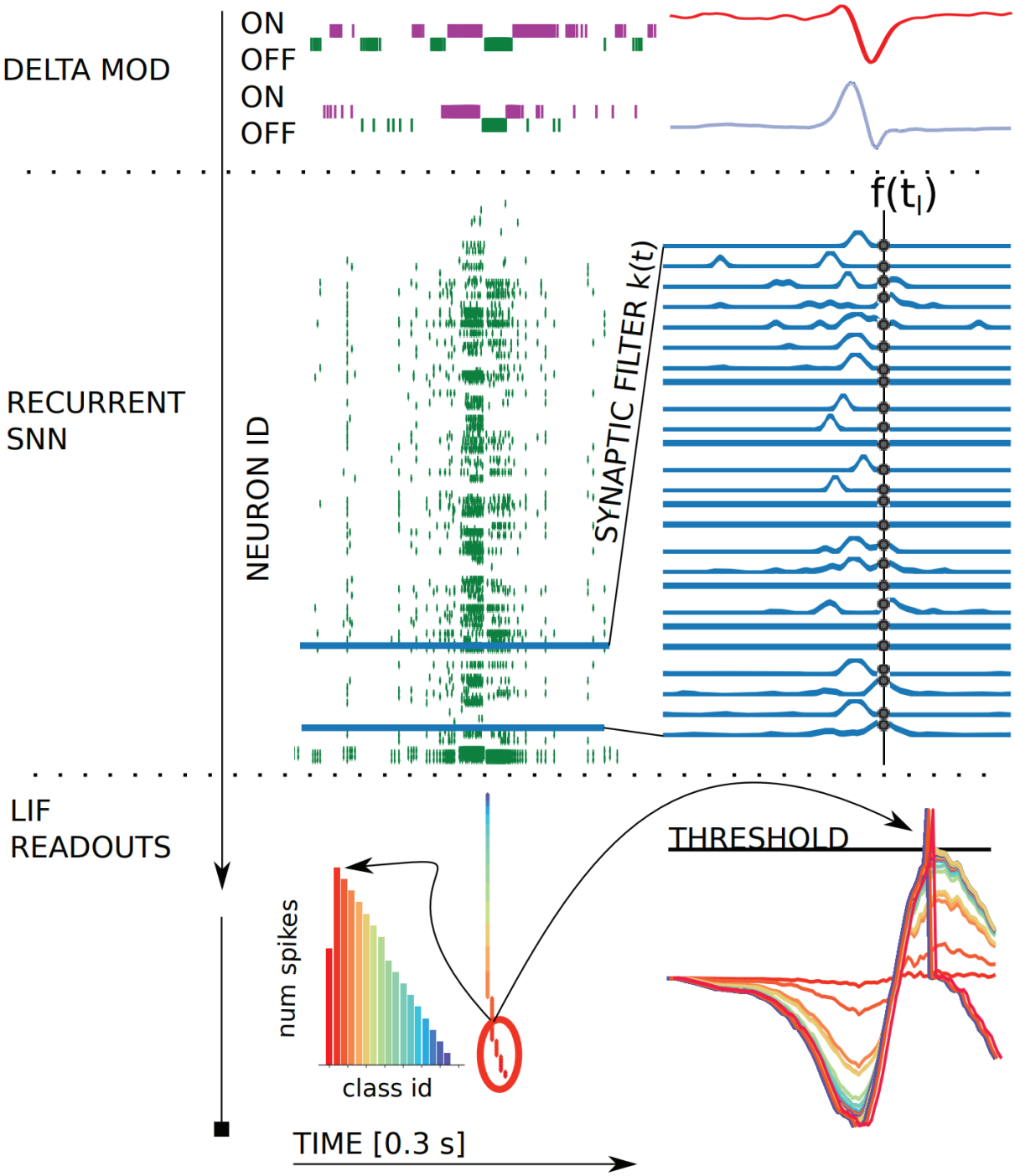


Рисунок В.5 – Повний конвеєр обробки на основі спайків

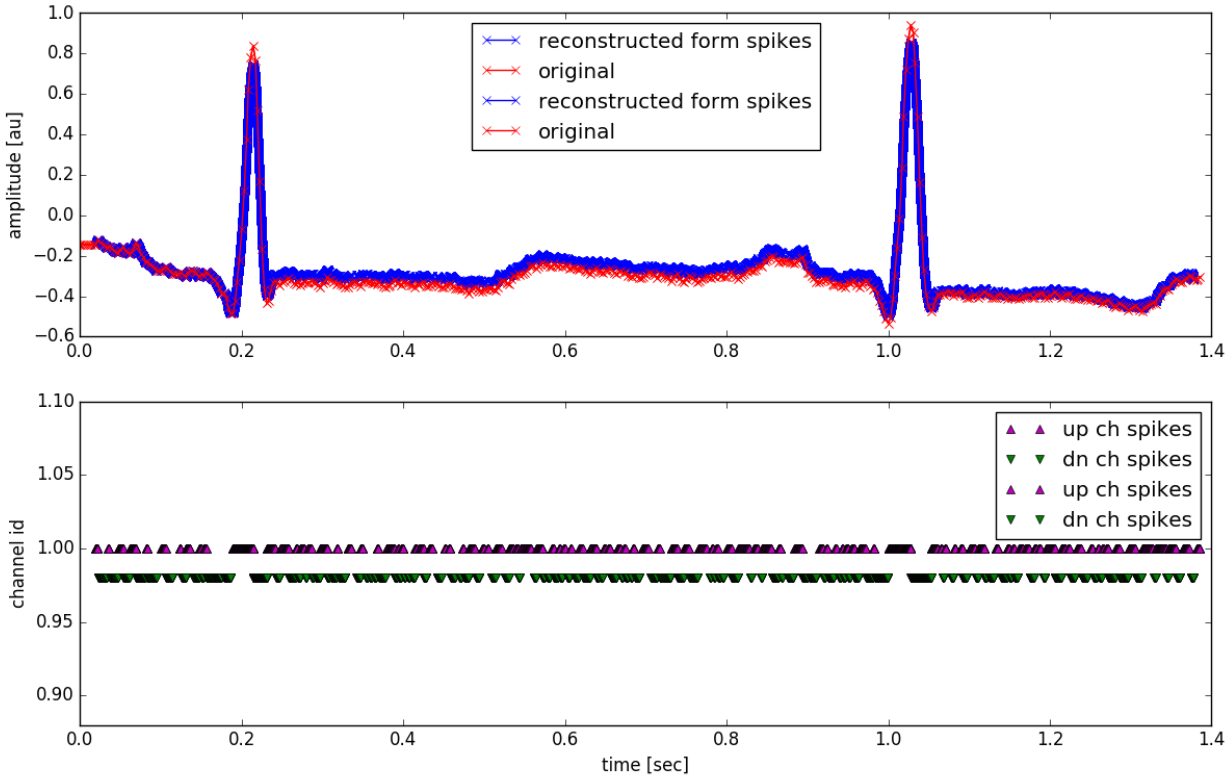


Рисунок В.6 – Вихідне зображення програмного забезпечення класифікації кардіограм на основі спайкінгової нейромережі

Додаток Г (довідниковий)

Інструкція користувача

Виконайте код із папки python і з батьківської папки, як показано нижче на рис. Г.1.

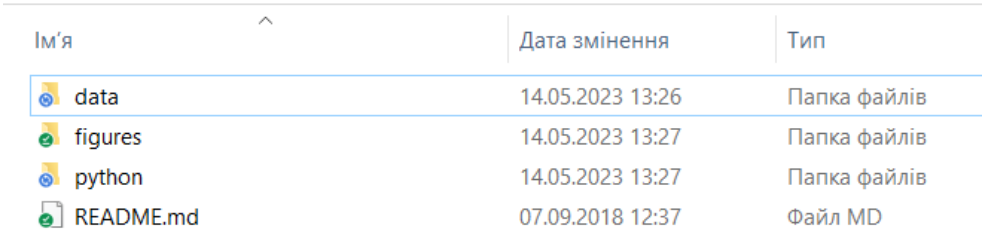


Рисунок Г.1– Структура програми класифікації кардіограм

Після обробки всіх файлів у папці даних буде створено вихідне зображення (рис. Г.2).

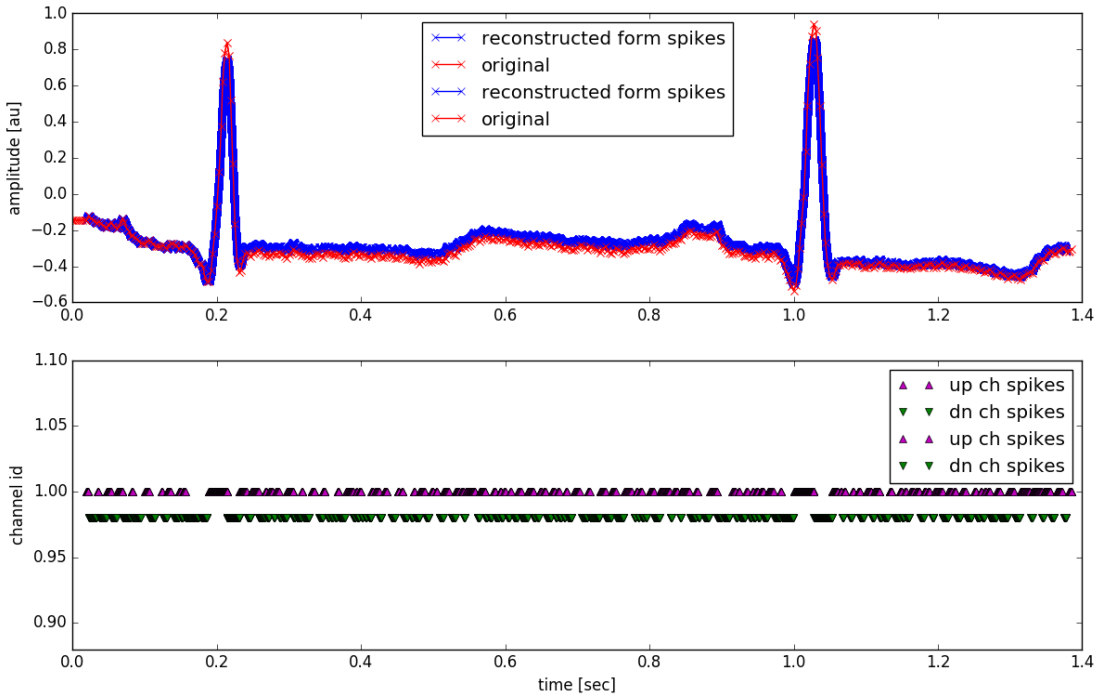


Рисунок Г.2– Вихідне зображення програми класифікації кардіограм

Потім програма створює і проводить навчання спайкінгової нейромережі на навчальній вибірці. Після навчання проводиться тестування роботи спайкінгової нейромережі на тестовій вибірці. І програма виводить результат тестування у вигляді:

```
testing accuracy, avg / total:       0.9631   /   3955
```

Виведена інформація свідчить про те, що розроблена програма має достовірність класифікації кардіограм на тестовій вибірці 96,3%,