Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод обробки даних електроенцефалограми з використанням засобів машинного навчання

(тема)

Виконав:

студент II курсу, групи СПм-20-1

Чернов Д.В.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: доц. Іващенко Г.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Чернову Дмитру Володимировичу
(прізвище, ім’я, по батькові)

1. Тема роботи Метод обробки даних електроенцефалограми з використанням засобів машинного навчання

затверджена наказом по університету від “ 5 ” листопада 2021 р. № 1657 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2021 р.

3. Вхідні дані до роботи 1) необроблені дані електроенцефалограми від апаратної частини; 2) документація про протокол комунікації апаратної частини; 3) засоби для обробки вхідного сигналу; 4) засоби машинного навчання; 5) документація до бібліотек Pandas, NumPy, SciPy; 6) джерела для вивчення з теорії машинного навчання; 7) документація з мови програмування Python та середовища розробки PyCharm.

4. Перелік питань, що потрібно опрацювати у роботі
1) аналіз предметної області;
2) огляд існуючих досліджень діагностики на основі даних електроенцефалограми;
3) вибір технологій розробки та інструментальних засобів;
4) реалізація запропонованого методу у вигляді програмного застосунку;
5) аналіз результатів досліджень;
6) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Слайд-презентація – 14 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих методів	09.11.21-12.11.21	
	обробки електроенцефалограми		
2	Вибір та обґрунтування методики дослідження	13.11.21-18.11.21	
3	Вибір інструментальних засобів	19.11.21-22.11.21	
4	Розробка програмного застосунку	23.11.21-29.11.21	
5	Проведення експериментів	30.11.21-03.12.21	
6	Оформлення матеріалів атестаційної роботи	04.12.21-07.12.21	
7	Подання атестаційної роботи керівникові та	08.12.21-09.12.21	
	її попередній захист		
8	Подання атестаційної роботи на рецензування	10.12.21-11.12.21	

Дата видачі завдання 8 листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Іващенко Г.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 79 с., 25 рис., 1 табл., 3 дод., 37 джерел.

ЕЛЕКТРОЕНЦЕФАЛОГРАМА, НОРМАЛІЗАЦІЯ, ПЕРІОДОГРАМА УЕЛЧА, ПРАВИЛО СІМПСОНА, МЕТОД СЕРЕДНЬОКВАДРАТИЧНОГО ВІДХИЛЕННЯ, МАШИНЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ, KNN, TGAM, NEUROSKY.

Метою кваліфікаційної роботи є реалізація методу обробки даних електроенцефалограми з використанням засобів машинного навчання.

У ході виконання кваліфікаційної роботи було реалізовано метод обробки даних електроенцефалограми для отримання типів хвиль мозку з необробленого сигналу, а також використання засобів машинного навчання з виконаною попередньою обробкою. Для отримання типів хвиль мозку було використано нормалізацію даних, метод Уелча та правило Сімпсона.

В якості засобу машинного навчання було обрано класифікацію за алгоритмом K-найближчих сусідів для прогнозування закритого або відкритого стану очей людини за допомогою наявного стану на момент збору даних завдяки можливостям апаратної частини.

ABSTRACT

Master’s thesis: 79 pages, 25 figures, 1 table, 3 appendices, 37 sources.

ELECTROENCEPHALOGRAM, NORMALIZATION, WELCH’S PERIODOGRAM, SIMPSON’S RULE, STANDARD DEVIATION METHOD, MACHINE LEARNING, CLASSIFICATION, KNN, TGAM, NEUROSKY.

The major goal of this thesis is to implement the method of electroencephalogram data processing using machine learning tools.

During the qualification work, the method of electroencephalogram data processing was implemented to obtain brain wave types from the raw signal, as well as the use of machine learning with pre-processing. Data normalization, Welch’s method, and Simpson’s rule were used to obtain brain wave types.

As a means of machine learning, the classification according to the algorithm of K-nearest neighbors was chosen to predict the closed or open state of the human eye using the current state at the time of data collection due to the capabilities of the hardware.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕЕГ – електроенцефалограма

ANN – штучна нейронна мережа (англ., Artificial Neural Network)

BCI – інтерфейс між мозком людини та комп'ютером (англ., Brain-computer interface)

CSV – тип файлу зі значеннями, які розділені комами (англ., Comma-Separated Values)

DL – глибоке навчання (англ., Deep Learning)

DNN – глибока нейронна мережа (англ., Deep Neural Network)

FFT – швидке перетворення Фур'є (англ., Fast Fourier Transform)

KNN – K-найближчих сусідів (англ., K-nearest neighbours)

ML – машинне навчання (англ., Machine Learning)

SVM – метод опорних векторів (англ., Support Vector Machine)

TGAM – ASIC-модуль мозкових хвиль (англ., ThinkGear ASIC Module)

ВСТУП

Сучасні досягнення в галузі комп’ютерного обладнання та процесорних технологій дозволили дослідникам у всьому світі значно розширити існуючі знання про складність людського мозку і отримати більш глибоке розуміння процесів і структур мозку. Хоча технологія електроенцефалограми (ЕЕГ) існує протягом більшої частини минулого століття, лише порівняно недавно вона стала доступною для широкого загалу [1].

У кінці XX та на початку XXI сторіччя ЕЕГ вийшла на споживчий ринок, коли були створені компанії, що є на сьогоднішній день найбільшими виробниками ЕЕГ-обладнання. Це забезпечило доступ до використання ЕЕГ-пристроїв користувачами у різних сферах, зокрема, в медичних, академічних, споживчих дослідженнях, охороні здоров’я [2].

Основним етапом аналізу ЕЕГ-даних є виявлення різних типів мозкових хвиль: альфа, бета, гамма, дельта і тета хвилі. Кожен з цих типів хвиль може бути виявлений з неопрацьованого ЕЕГ-сигналу та характеризує різний стан людського мозку [3]. Але для подальшого аналізу, наприклад, виявлення патернів, що відповідають тим чи іншим діям користувача, необхідна обробка вхідного сигналу [4].

Вимірювання ЕЕГ-даних може проводитись при наявності у приміщеннях великої кількості електричних пристроїв, які призводять до появи викривлень у вимірюваннях, через це виникає необхідність у попередній обробці вхідного сигналу. Підготовлені вхідні дані можуть бути у подальшому оброблені за допомогою методів машинного навчання для автоматичного аналізу даних електроенцефалограми [4].

Засоби машинного навчання набули широкого використання у ЕЕГ-пристроях, які виконують роль інтерфейса між мозком людини та комп’ютером (Brain-computer interface, BCI). BCI-пристрої можуть виявити складні мозкові хвилі у реальному часі для контролю або управління різними

пристроями, що може допомогти особам з легкими та важкими руховими порушеннями, включаючи тих, хто не може спілкуватися з іншими. При використанні BCI-пристроїв інваліди уявляють виконання діяльності або зосереджуються на об’єкті, не покладаючись на рухи м’язів, завдяки перетворенню специфічної активності мозку на функції управління та команди. Прикладами можуть виступати керування візками або переміщення курсору на екрані [5].

Також однією з популярних сфер застосування засобів машинного навчання для аналізу ЕЕГ даних є виявлення та передбачення можливих артефактів при вимірюванні. У дослідженні ЕЕГ моргання є одним з широко відомих типів артефактів, який найчастіше з’являється при будь-якому вимірюванні ЕЕГ. Дані артефакти можна визначити як сплеск електричних потенціалів, частотно-часові характеристики яких варіюються в залежності від конкретної людини. Їхня присутність може негативно вплинути на різноманітні дослідження або бути корисною при застосуванні в різних ЕЕГ-пристроях. Тому виявлення та передбачення сигналів моргання є важливим етапом аналізу ЕЕГ [6].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз предметної області

Мозок людини складається з мільярдів пов’язаних нейронів, взаємодія між якими створює незначний електричний розряд. Ці заряди неможливо виміряти ззовні черепа, але активність, яка створена сотнями тисяч одночасних розрядів, агрегується у хвилі, які можна виміряти.

Різні стани мозку є результатом різних моделей нейронної взаємодії. Такі закономірності призводять до хвиль, що характеризуються різними амплітудами та частотами, наприклад, хвилі між 12 і 30 герцами, бета-хвилі, пов’язані з концентрацією, тоді як хвилі між 8 і 12 герцами, альфа-хвилі, пов’язані з розслабленням та станом психічного спокою. Скорочення м’язів також асоціюється з унікальними хвильовими моделями, тому що деякі ЕЕГ-пристрої виявляють моргання, ізолюючі ці патерни [1].

Пристрої електроенцефалографії – це електронні пристрої, які можуть вимірювати електричні сигнали мозку. Датчики ЕЕГ зазвичай вимірюють різні електричні сигнали, що створюються активністю великих груп нейронів біля поверхні мозку протягом певного періоду часу. Вони працюють шляхом вимірювання невеликих коливань електричного струму між шкірою та електродом датчика, посилення електричного струму та виконання будь-якої фільтрації, наприклад, смугової [3].

Окремо виділяють пристрої, які виконують роль інтерфейсу між мозком людини та комп’ютером (Brain-computer interface). BCI-пристрої можуть виявити складні мозкові хвилі 30 різних виразів, емоцій та дій за допомогою машинного навчання [5].

Майже всі системи BCI містять в якості основної частини алгоритм машинного навчання, який вивчає дані тренувань і дає функцію, яка може бути використана для розрізнення різних моделей активності мозку. Він адаптує

систему BCI до мозку конкретного суб’єкта. Це зменшує навчальне навантаження на предмет.

Для простоти та з практичних міркувань, алгоритми машинного навчання зазвичай поділяються на дві частини: вилучення та класифікації. Частина вилучення необхідна для перетворення необроблених сигналів мозку в уявлення, яке полегшує класифікацію. Тобто мета виділення функцій полягає у видаленні шуму та іншої непотрібної інформації з вхідних сигналів, одночасно зберігаючи інформацію, важливу для розрізнення різних класів сигналів. За допомогою алгоритмів машинного навчання ці функції перетворюються на сигнал управління [5].

1.2 Різновиди сигналів електроенцефалограми

Електроди пристрою ЕЕГ фіксують електричну активність, виражену в різних частотах ЕЕГ. Використовуючи швидке перетворення Фур’є (Fast Fourier Transform), ці необроблені сигнали ЕЕГ можна ідентифікувати як різні хвилі з різними частотами. Частота, що визначає швидкість електричних коливань, вимірюється в циклах на секунду – один Герц (Гц) дорівнює одному циклу в секунду. Мозкові хвилі класифікуються за частотою на чотири основні типи: бета, альфа, тета і дельта (рисунок 1.1) [7].

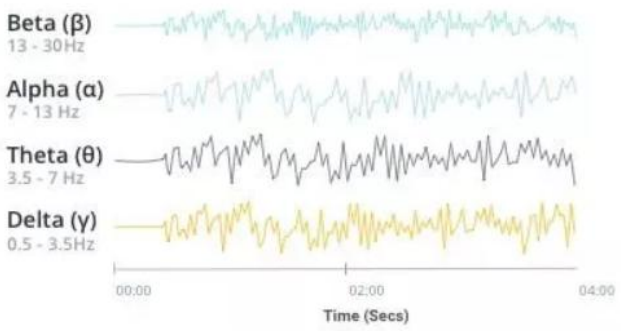

Рисунок 1.1 – Відстеження хвилей ЕЕГ у реальному часі на графіках

Альфа-хвилі (від 7 до 13 Гц) часто асоціюються з розслабленим, спокійним і усвідомленим станом. Альфа-хвилі можна виявити в потиличній і задній частинах мозку. Альфа-хвилі можна викликати, якщо розслабитись та закрити очі. Вони рідко присутні під час інтенсивних пізнавальних процесів, таких як мислення, розумовий розрахунок та вирішення проблем. У більшості дорослих альфа-хвилі мають частоту від 9 до 11 Гц [7].

Бета-хвилі (від 13 до 30 Гц) найтісніше пов’язані зі свідомістю або перебуванням у стані неспання, уважності та настороженості. Бета-хвилі низької амплітуди пов’язані з активною концентрацією, або з зайнятим або тривожним станом душі. Бета-хвилі також пов’язані з моторними рішеннями, наприклад, стримування руху або сенсорний зворотний зв’язок руху [8].

Активність мозку в діапазоні частот від 4 до 7 Гц називається тета-активністю. Тета-ритм, виявлений при вимірюванні ЕЕГ, часто зустрічається у молодих людей, особливо над скроневими областями та під час проблем з диханням. У літніх людей тета-активність з амплітудою більше приблизно 30 мілівольт (мВ) спостерігається рідше, за винятком сонливості [8].

Дельта-хвилі (від 0.5 до 3.5 Гц) є високоамплітудними мозковими хвилями, які переважно виявляються у немовлят, а також асоціюються з глибокими стадіями сну [8].

1.3 Сфери застосування пристроїв електроенцефалограми

ЕЕГ-пристрої можуть надавати цінну інформацію про стан психічного здоров’я людини, її думки та уяву, тому дослідники різних галузей використовують ці пристрої для своїх досліджень. Сфери застосування таких пристроїв (рисунок 1.2) можна розділити на п’ять категорій: біометрія, нейронаука та клінічне застосування, нейромаркетинг, спеціальні рішення, пристрої взаємодії між мозком людини та комп’ютером [5, 11].

ЕЕГ-пристрої використовуються у біометрії для розпізнавання та визначення відмінності людей за допомогою фізіологічних чи поведінкових

особливостей, таких як відбитки пальців, голос, обличчя, оболонка ока, погляд, хода та поза. Навчання показують, що дані ЕЕГ можуть надати інформацію про відмінності людей. Останнім часом когнітивний та емоційний стан мозку використовується для біометрії, тобто дані ЕЕГ використовуються для ідентифікації людей.

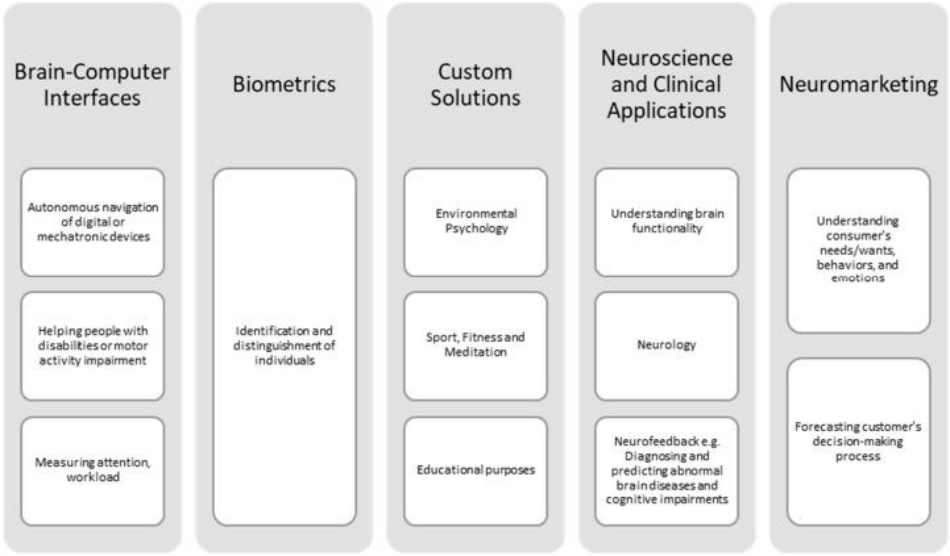

Рисунок 1.2 – Основні сфери застосування електроенцефалограми

Як правило, психологічні дослідження використовують ЕЕГ для вивчення мозкових процесів, що лежать в основі уваги, навчання та пам’яті. На основі масового повторення проб, потенціали, пов’язані з подіями (ERPs), витягуються з безперервного потоку даних ЕЕГ, що дозволяє охарактеризувати мозкові процеси, викликані подіями, у детальній шкалі часу (з періодами, що складають десятки мілісекунд).

ERPs можна охарактеризувати за їх амплітудою (у мілівольтах з позитивною та негативною хвилями, позначеними P та N відповідно), часом (у мілісекундах відносно початку події) та розподілом напруги на всіх електродах (топографія). Визначено конкретні ERP, які можливо використати для обробки виявлення обличь (N170), слів та значення (N400), несподіванки (P300) або відкликання пам’яті (P600) [9].

ЕЕГ-пристрої у спеціальних рішеннях використовуються для створення комфортного середовища, покращення самопочуття та якості життя, пришвидшення процесу навчання.

Спортсмени та будь-які зацікавлені споживачі можуть використовувати ЕЕГ для відстеження своєї діяльності мозку так само, як вони можуть відстежувати кількість кроків, які вони роблять за день. Такі ЕЕГ-пристрої можуть вимірювати когнітивні функції, такі як увага та відволікання, стрес та когнітивне навантаження, тобто загальна здатність мозку до розумової діяльності. Такі дослідження допомагають виявити як мозок реагує на події повсякденного життя. Дані ЕЕГ забезпечують зворотний зв’язок, який може бути використаний для розробки науково обґрунтованих стратегій зменшення стресу чи поліпшення зосередженості [10].

ЕЕГ-пристрої можуть бути використані в освітніх цілях для вимірювання читацьких здібностей учнів, рівня розуміння під час онлайн-лекцій або рівня концентрації та когнітивного навантаження, коли учні намагаються розв’язати математичну головоломку з метою конструювання інтелектуальної репетиторської системи.

У сфері нейромаркетингу економісти використовують дослідження ЕЕГ для виявлення мозкових процесів, які керують споживчими рішеннями, ділянок мозку, які активні, коли споживачі купують який-небудь продукт чи послугу, та психічних станів, у яких перебуває відповідна особа під час вивчення фізичних чи віртуальних магазинів [11].

Порівняно новою сферою є використання електроенцефалограми як інтерфейс між мозком людини та комп’ютером (рисунок 1.3). Такі пристрої використовують ЕЕГ-дані у реальному часі для контролю або управління механічними чи електронними пристроями, завдяки більш розвинутим знанням про активні ділянки мозку при сприйманні подразників, руханні тіла або запам’ятовуванні речі. Дана технологія породжує багато сфер застосування ЕЕГ, наприклад, для допомоги паралізованим пацієнтам керувати інвалідними візками або електронними пристроями [5, 11].

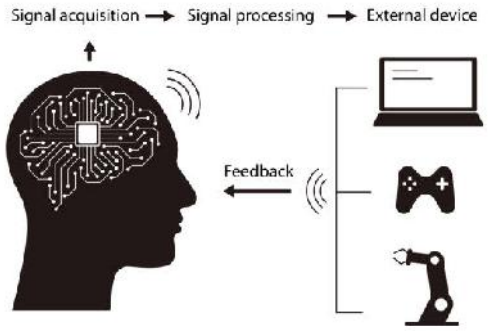

Рисунок 1.3 – Схема роботи нейрокомп’ютерного інтерфейсу

Вимірювання даних ЕЕГ в рамках зазначених сфер може проводитись при наявності у приміщеннях великої кількості електричних пристроїв (наприклад, побутової техніки), які створюють певний рівень навколишнього «шуму». Цей «шум» заважає хвилям, що виходять з мозку, тому більшість пристроїв ЕЕГ можуть збирати показання, якщо вони не на голові людини. У минулому пристрої ЕЕГ обходили цю проблему, вимірюючи ці сигнали в середовищах, де суворо контролюється електрична активність, і збільшували силу сигналу даних, що надходять з мозку, шляхом застосування струмопровідного розчину [4].

Проте більшість людей не мають у своєму будинку приміщень, позбавлених електронних пристроїв, і вони не хочуть наносити на голову струмопровідну рідину щоразу, коли вони користуються пристроєм. Компанії, які є виробниками ЕЕГ-пристроїв для ринку, розробляють власні складні алгоритми, що є вбудованими в їхні продукти, які фільтрують цей «шум» [4].

Існує великий інтерес до використання методів машинного навчання для автоматичного аналізу електроенцефалограми, особливо в області клінічної діагностики на основі ЕЕГ. Наприклад, даний підхід є основою для виявлення та прогнозування епілептичних нападів з метою попередити пацієнтів про майбутні судоми або контролювати стимуляцію мозку для запобігання або припинення судомної активності. Крім того, засоби машинного навчання дозволяють автоматизувати процес виявлення стадій сну на основі ЕЕГ та

неврологічну діагностику специфічних захворювань і розладів, таких як: хвороба Альцгеймера, депресія, черепно-мозкові травми, удари, розлади свідомості або загальні патології ЕЕГ [10].

1.4 Аналіз існуючих досліджень діагностики на основі даних електроенцефалограми

Для діагностики епілептичних припадків на основі електроенцефалограми з використанням засобів машинного навчання необхідна попередня обробка сигналу, яка включає в себе три етапи: видалення шуму, нормалізація і підготовка сигналу для засобів машинного навчання. На етапі видалення шуму зазвичай використовуються фільтри з кінцевою імпульсною характеристикою (FIR) або з нескінченною імпульсною характеристикою (IIR) для усунення додаткового шуму сигналу. Потім виконується нормалізація з використанням різних схем, таких як метод z-оцінки. Для підготовки сигналу для використання у засобах машинного навчання використовуються різні часові інтервали, частотні і частотно-часові методи [12].

Серед алгоритмів машинного навчання виділяють моделі глибокого навчання (Deep Learning), які зазвичай розробляються з використанням мови програмування Python з численними наборами інструментів з відкритим кодом. Мова Python, яка має вільно доступні набори інструментів для DL, допомагає дослідникам розробити нові автоматизовані системи, а завдяки хмарним обчисленням забезпечила доступ до обчислювальних ресурсів для всіх дослідників.

Найбільш популярним засобом DL є бібліотека TensorFlow, а також Keras, яка є однією з високорівневих API для цієї бібліотеки. Дані бібліотеки широко використовуються для виявлення епілептичних припадків за допомогою глибокого навчання в розглянутих роботах через їх універсальність і застосовність [13].

Глибокі нейронні мережі як модель глибокого навчання є галуззю машинного навчання, яка виникла від штучних нейронних мереж, але на відміну від звичайних, глибокі нейронні мережі являють собою структури з більш ніж двома прихованими шарами. Таке збільшення розміру мережі призводить до значного збільшення кількості параметрів мережі, що вимагає відповідних методів навчання, а також заходів щодо запобігання перенавчання вивченої мережі. Згорткові мережі використовують фільтри, які згорнуті з вхідними шаблонами, замість множення вектору ваг (матриці), що значно скорочує кількість параметрів для навчання [14].

Крім того, пропонуються й інші методи для прискорення навчання мережі. Об’єднання шарів зменшує розмір вхідного шаблону до наступного згорткового шару. Для запобіганню переоснащення вивченої мережі та підвищення її здатності до навчання і швидкості використовуються наступні методи: пакетна нормалізація, відсів, рання зупинка, неконтрольоване або частково неконтрольоване навчання і методи регуляризації [15].

Згорткові нейронні мережі (Convolutional Neural Networks) є одними з найпопулярніших мереж DL, яким присвячено більшість досліджень в області машинного навчання. Спочатку вони були представлені для програмних застосунків для обробки зображень, але пізніше були адаптовані до одно та двовимірної архітектури для діагностики та прогнозування захворювань з використанням біологічних сигналів. Цей клас мереж DL широко використовується для виявлення епілептичних припадків за сигналами ЕЕГ.

У двовимірних згорткових нейронних мережах (2D-CNN) одномірні (1D) сигнали ЕЕГ спочатку перетворюються в двовимірні графіки з використанням методів візуалізації, таких як спектрограма, спектр більш високого порядку, та wavelet-перетворення, а потім застосовуються до входу згорткової мережі. В одновимірних архітектурах сигнали ЕЕГ подаються в одновимірній формі на вхід згорткової мережі. У цих мережах зміни вносяться в основну архітектуру 2D-CNN, що робить його здатним обробляти сигнали 1D-EEG.

Оскільки в області виявлення епілептичних припадків використовуються як двовимірні, так і одномірні згорткові нейронні мережі (1D-CNN), вони досліджуються окремо [16, 17].

На даний час глибокі 2D-мережі використовуються для різних медичних програм за допомогою комп’ютерної томографії та рентгенографії і розладів аутистичного спектра за допомогою методів МРТ. У 2012 році було запропоновано мережу для вирішення проблем класифікації зображень, а потім використання аналогічних мереж для різних завдань, таких як класифікація медичних зображень, щоб уникнути труднощів попередніх мереж та вирішити більш складні проблеми з кращою продуктивністю. На рисунку 1.4 показаний загальний вигляд 2D-CNN мережі, яка використовується для виявлення епілептичного нападу [18].

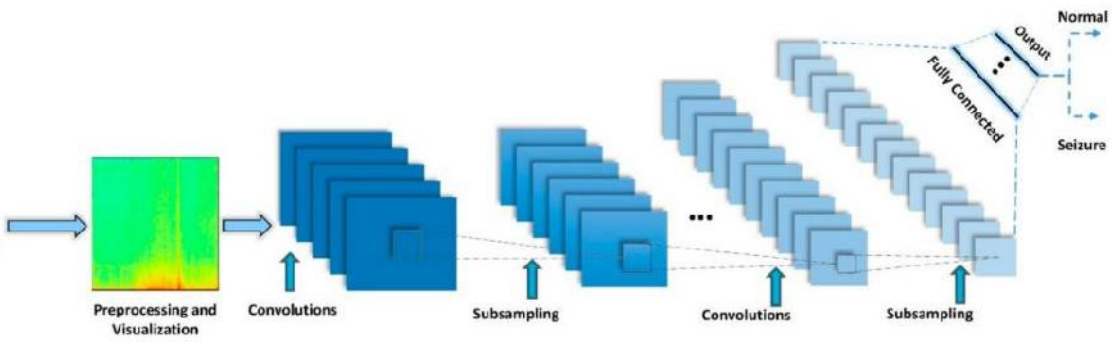

Рисунок 1.4 – Типова глибока 2D-мережа для виявлення епілептичних нападів

1.5 Постановка завдання

Метою роботи є створення та дослідження методу обробки даних електроенцефалограми з використанням засобів машинного навчання. Отримання даних електроенцефалограми необхідно виконувати від апаратної частини, яка представлена у вигляді ЕЕГ-модуля, мікроконтролера, набору електродів, модуля передачі даних, а також акумуляторної батареї.

Для керування процесом роботи ЕЕГ-пристрою необхідно використовувати мікроконтролер, який буде застосовувати ЕЕГ-модуль для обробки даних, які надходять від одного або декілька електродів, після чого мікроконтролер буде відправляти отримані дані використовуючи модуль передачі даних, яким може виступати Bluetooth чи WiFi модуль. Bluetooth-модуль забезпечує більшу автономність та компактність пристрою, тому слід надати перевагу саме цьому рішенню.

Для забезпечення автономності пристрою та можливостей до більш зручного збору даних можна використати стандартні AAA батарейки. Необхідно також передбачити корпус для зберігання всіх компонентів пристрою та ізолювання їх від зовнішнього середовища.

У якості програмної частини будуть виступати скрипти, написані на мові програмування Python, які будуть виконувати прийом даних від апаратної частини по Serial порту по підключеному Bluetooth, тому необхідно передбачити налаштування імені COM-порта, як вхідну зміну до скриптів. Передбачити виконання Python-коду як на операційних системах Windows, так і Linux.

Використання скриптів на мобільних пристроях не буде передбачено для простоти тестування, однак може бути використано завдяки тому, що Python є кросплатформеною інтерпретованою мовою.

Виконання Python-скриптів буде виконуватись через командну строку для більш спрощеного використання. Необхідно передбачити можливість налаштування вхідних змінних через командну строку, таких як ім’я COM-порта, а також ім’я команди для виконання. Останнє необхідно для виконання тільки частини логіки роботи з усієї можливої роботи, наприклад, список імен команд може бути наступним: збір даних від пристрою та їх збереження, обробка збережених даних від пристрою та їх збереження, вивід оброблених даних у вигляді графіків. Для останнього необхідно визначити засоби для відображення графіків з використанням мови Python. Для цього можна використати Jupyter Notebook на базі Anaconda або простий експорт графіків у

вигляді PNG-зображень. Останній підхід дозволить більш гнучку роботу для якої буде потребувати меньше зовнішніх залежностей.

Етап збору даних по Bluetooth від пристрою буде потребувати додатковий код для обробки потоку байт, його розбору відповідно до протоколу комунікації пристрою та збереженню з використанням додаткових програмних бібліотек на Python. Забезпечити можливий інтервал збору даних, а також можливе відсіювання викривлених даних згідно до можливостей протоколу комунікації пристрою.

Етап обробки вхідних даних передбачає можливу попередню обробку використовуючи фільтрацію та нормалізацію даних. Фільтрація дозволить позбутись непотрібних «шумів», однак часто деяка фільтрація шумів виконується вже на стороні пристрою. Нормалізація даних є необхідною для роботи з алгоритмами машинного навчання.

Метою використання засобів машинного навчання є прогнозування закритого або відкритого стану очей людини на основі зібраних апаратною частиною даних електроенцефалограми, а також наявності в ЕЕГ-модулів можливість визначати стан очей на момент збору даних.

Етап виводу результату обробки включає можливість показу вихідних графіків з необробленими чи обробленими даними у вигляді зображень, але також і можливість перегляду отриманих даних у вигляді CSV файлів з вказаними проміжками часу.

2 ТЕХНОЛОГІЇ ТА МЕТОДИ ОБРОБКИ ДАНИХ

2.1 Апаратні засоби отримання електроенцефалограми

Модуль ThinkGear ASIC (TGAM) від компанії NeuroSky є основою апаратної частини розпізнавання мозкових хвиль. Розробки на базі модуля TGAM дозволяють партнерам NeuroSky швидко та ефективно виводити на ринок споживчі технології на основі аналізу електроенцефалограми. Дані модулі є одними з найпопулярніших у світі рішень для отримання даних електроенцефалограми. При використанні сухого електрода він сприймає сигнал людського мозку, відфільтровує сторонні шуми та поміхи від електричних приборів, перетворює його у цифровий сигнал.

У модуль TGAM є вбудований чіп TGAT – повністю інтегрований однокристальний датчик ЕЕГ (рисунок 2.1). Чіп поставляється з запрограмованими функціями NeuroSky eSense, A/D, посиленням при накладені на голову та фільтрацією шуму для електроміографії та перешкод від мережі змінного струму 50 або 60 Гц. Ціна TGAM розрахована на масове впровадження мобільних пристроїв, освітнього обладнання та інших товарів [20]. Основні характеристики модулю TGAM наведені у додатку Б.

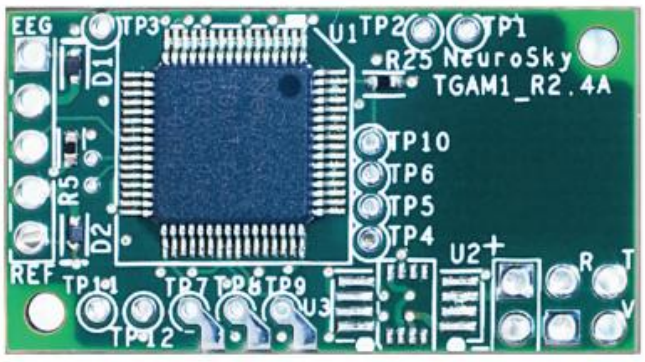

Рисунок 2.1 – Модуль ThinkGear ASIC від компанії NeuroSky

2.2 Типи пристроїв отримання та обробки електроенцефалограми

Дротові та бездротові гарнітури ЕЕГ передають дані на комп’ютер за допомогою кабелю, бездротового або з’єднання по Bluetooth відповідно. Струмопровідні з’єднання ЕЕГ є більш стабільними і часто можуть передавати більше даних у форматі даного часу, але не забезпечують свободу пересування, як при випадку використання бездротових з’єднань. Одним з основних недоліків бездротових гарнітур ЕЕГ є те, що під час збору даних мозку гарнітура може втратити бездротове з’єднання і не записувати дані. Незалежно від типу з’єднання, рух кабелів та електродів може спричинити артефакти у сигналі ЕЕГ, оскільки може порушити зв’язок між електродами та шкірою голови.

ЕЕГ-пристрої вимагають цілісного електричного з’єднання між шкірою голови особи, яка носить пристрій. Цього можна досягти різними способами (рисунок 2.2) [22].

Рисунок 2.2 – ЕЕГ-пристрої зі сухим з’єднанням (а), на основі гелю (б), фізрозчину (в)

За допомогою з’єднання на основі м’якого гелю електроди з’єднуються зі шкірою голови шляхом нанесення струмопровідного гелю на поверхню кожного електрода. Після завершення експерименту необхідно очистити гарнітуру, видаливши гель та очистивши електроди, використовуючи розчини зі спиртом у складі.

Окремі гарнітури ЕЕГ вимагають використання струмопровідного гелю для створення електричного контакту з низьким опором між шкірою людини та електродом. Для цього на кожен електрод ЕЕГ пристрою наноситься фізрозчин.

Сухі ЕЕГ-пристрої не використовують гелю або фізіологічного розчину для з'єднання електродів зі шкірою голови, що полегшує запис даних ЕЕГ без допомоги кваліфікованого фахівця. Крім того, час його налаштування значно коротший за гарнітури з розчинами.

Деякі типи з'єднань датчиків ЕЕГ не вписуються чітко в одну з цих двох категорій. Струмопровідні тверді гелеві матеріали виробництва компанії Enobio, також успішно використовуються в ЕЕГ-пристроях [22, 23].

2.3 Аналіз пристроїв електроенцефалограми від компанії NeuroSky

Компанія NeuroSky, яка заснована в 2004 році у Кремнієвій долині, Каліфорнія, є виробником технологій мозкового комп'ютерного інтерфейсу (BCI) для споживчих продуктів. Компанія адаптує технології електроенцефалографії та електроміографії, щоб відповідати споживчому ринку в ряді сфер, таких як розваги, освіта, автомобілебудування та охорона здоров'я.

Технологія NeuroSky дозволяє проводити недорогі дослідження та вироби, пов'язані з ЕЕГ, за допомогою недорогих сухих датчиків; більш старі ЕЕГ вимагають нанесення струмопровідного гелю між датчиками та головою. Системи також включають вбудоване програмне та апаратне забезпечення для зменшення електричного «шуму» та використовують вбудовані на рівні мікросхеми рішення для обробки та виведення сигналу.

Neurosky в першу чергу працює як виробник оригінального обладнання або OEM, співпрацюючи з промисловими партнерами, розробниками та науково-дослідними установами для впровадження технологій у власні продукти та системи. Коли NeuroSky випускає продукти для споживачів, такі

як MindSet та MindWave, вони, як правило, розроблені для максимальної гнучкості використання стороннього та відкритого коду. NeuroSky MindWave Mobile (рисунок 2.3) є першою найпоширенішою версією універсальної чорної гарнітури з Bluetooth 2.1, яка підтримується всіма популярними мобільними застосунками для роботи з BCI, не рахуючи нових для iOS з Bluetooth low energy, з’явилася у 2011 році, коштувала близько 100 доларів у США та зникла з продажу приблизно в наприкінці 2016 року в користь NeuroSky MindWave Mobile+. Дана модель з’явилася в середині 2016 року та коштувала близько 100 доларів у США після чого зникла з продажу приблизно в середині 2017 року на користь BrainLink Pro. Не має третього положення перемикача на відміну від старої версії MindWave Mobile, бо підключення з пристроями можливо виконати у будь-який момент при включеній гарнітурі.

Рисунок 2.3 – Нейрогарнітура NeuroSky MindWave Mobile

У версії MindWave Mobile+ покращено підключення до iPhone завдяки використанню BLE, який вперше з’явився у протоколі Bluetooth 4, для чого потрібен як мінімум iPhone 5 або iPad, замість старого Bluetooth третьої або нижче версій. Деяке старе програмне забезпечення не може підключитись через BLE навіть на нових пристроях. Застосунок Brainwave Visualizer є доступним для всіх чотирьох платформ при використанні гарнітури NeuroSky MindWave Mobile+. Пристрій є також сумісним з операційними системами Windows 7/8/10 [24].

2.4 Перетворення даних при стандартизації або нормалізації

Перетворення даних є одним з основних етапів обробки даних, для цього використовується масштабування, стандартизація та нормалізація.

Результатом стандартизації (або нормалізації Z-балів) є те, що характеристики будуть змінені, щоб забезпечити середнє та стандартне відхилення 0 та 1 відповідно.

$$x_{s} = \frac{x - m(x)}{s \quad d(x)}, \tag{2.1}$$

Цей метод полягає у повторному масштабуванні значень об’єктів із значенням розподілу від 0 до 1, що є корисним для алгоритмів оптимізації, таких як градієнтний спуск, які використовуються в засобах машинного навчання, які зважують вхідні дані (наприклад, нейронні мережі). Масштабування також використовується для алгоритмів, які використовують вимірювання відстані, наприклад, K-найближчих сусідів (KNN).

Іншим поширеним підходом є так звана Min-Max нормалізація (Min-Max масштабування). Ця техніка полягає в повторному масштабуванні об’єктів зі значенням розподілу від 0 до 1. Для кожної ознаки мінімальне значення цієї функції перетворюється на 0, а максимальне – на 1.

$$x_{n} = \frac{x - m(x)}{m(x) - m(x)}, \tag{2.2}$$

Деякі моделі машинного навчання фундаментально базуються на матриці відстані, також відомі як класифікатор на основі відстані, наприклад, K-найближчих сусідів, SVM та штучні нейронні мережі. Масштабування функцій надзвичайно важливо для цих моделей, особливо у випадку, коли діапазон функцій дуже різний. В іншому випадку функції з великим діапазоном матимуть великий вплив на обчислення відстані.

Нормалізація Max-Min зазвичай дозволяє перетворювати дані у різних масштабах, таким чином, що жоден конкретний вимір не буде домінувати в статистиці, але це не вирішує проблему з викидами. У той же час стандартизація дозволяє користувачам краще обробляти викиди та полегшує сходження для деяких обчислювальних алгоритмів, таких як градієнтний спуск. Тому в таких випадках зазвичай надається перевага стандартизації, ніж Min-Max нормалізації.

У таблиці 2.1 наведені алгоритми, які потребують масштабування значень [25].

Таблиця 2.1 – Алгоритми, які потребують масштабування значень

Алгоритм	Причина використання масштабування значень
Метод кластеризації k-середніх	Використання Евклідової відстані
K-найближчих сусідів	Вимірювання відстаней між парами прикладів
Метод головних компонент (PCA)	Отримання об’єкта з максимальною дисперсією
Штучна нейронна мережа	Застосування градієнтного спуску
Градієнтний спуск	Розрахунок порогу стає швидшим після масштабування функцій, а швидкість навчання у рівнянні оновлення стохастичного спуску є однаковою для кожного параметра

Якщо алгоритм не ґрунтується на відстані, масштабування функцій не має значення, використовується наївний класифікатор Байєса, лінійний дискримінантний аналіз та моделі на основі дерева (збільшення градієнта, випадковий ліс тощо) [25, 26].

2.5 Використання машинного навчання

Машинне навчання (ML) використовує принципи статистики та інформатики та систематично використовує алгоритми для розкриття прихованого характерного та внутрішнього зв’язку даних. Поширення використання машинного навчання спостерігається у багатьох областях, наприклад, у розпізнаванні мовлення, образів зображення, веб-пошуку, фільтрації спаму, автопілоті. У медицині машинне навчання підвищує точність прогнозування та виявлення певного захворювання, а також оцінки прогнозу. Для біомедичних даних засоби машинного навчання відображають принципові, автоматичні та об’єктивні алгоритми для високорозмірних та складних даних. Наприклад, машинне навчання показує свої переваги у виборі генів порівняно із застосуванням класичного методу відбору ознак [19].

Згідно з наведеним у розділі 1.5 алгоритмом визначення засобу машинного навчання відповідно до вхідних даних та вимог, необхідно використовувати метод класифікації через наявність вхідних даних для навчання, а також відомі стани для прогнозування на момент обробки даних (рисунок 2.4).

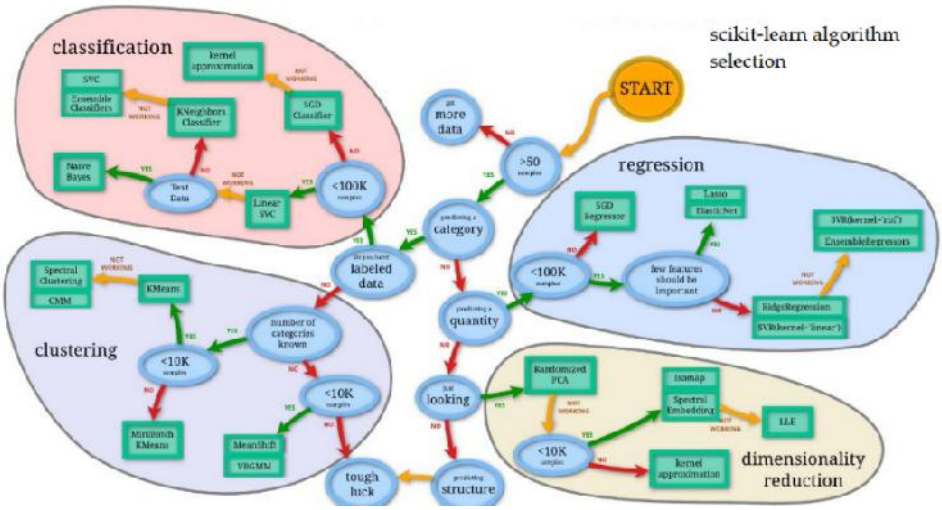

Рисунок 2.4 – Визначення засобу машинного навчання

Одними з направлень використання ML є розпізнавання об’єктів та формування прогнозу щодо майбутніх подій. Обчислювальною особливістю ML є виведення гіпотез через досвід навчання або прикладів за допомогою математичних алгоритмів. Завдання з машинного навчання можуть бути загалом організовані у двох основних галузях: із залученням або без вчителя.

Метод із залученням вчителя отримує попередньо позначені вхідні дані та прагне перейти до найбільш оптимізованого класифікатора за допомогою алгоритму, який є навченим для класифікації немаркованих даних. Порівняно з методом із залученням вчителя, алгоритми без вчителя, наприклад, алгоритми кластеризації та зменшення розмірності відносяться до процесу встановлення математичних моделей після аналізу подібності між немаркованими вхідними даними для виявлення тенденцій, підгруп чи викидів.

Навчання з частковим залученням вчителя працює між поняттями контрольованого та неконтрольованого навчання, вивчаючи невелику кількість маркованих наборів даних із поєднанням великої кількості немаркованих наборів даних, для створення класифікатора (або модельної функції).

Навчання з підкріпленням є одним з найсучасніших прийомів у динамічних системах, наприклад, в системі, що змінюється за часом, енергетичних системах, шляхом автоматичного вивчення оптимальних стратегій управління. Підсилення означає, що хороші дії позитивно підсилюються, а погані – деградують, щоб наблизитися роботу системи до оптимальної стратегії.

Поглиблене навчання, як одна категорія методів ML, побудовано на репрезентативному навчанні, система якого полягає в автоматичному вивченні та виявленні шаблонів для класифікатора з кількох шарів вхідних даних. Ансамблеве навчання об’єднує декілька моделей навчання (таких як нейронні мережі, метод опорних векторів та методи Байєса), а також навчає самостійно або спільно вхідні дані [29].

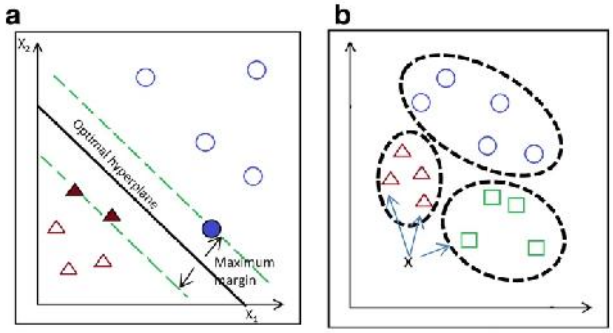

Рисунок 2.5 – Метод опорних векторів (а), K-найближчих сусідів (б)

За останні кілька десятиліть було розроблено велику кількість методів класифікації. Серед найбільш широко використовуваних можна виділити: метод K-найближчих сусідів (KNN), штучні нейронні мережі (ANN), метод опорних векторів (SVM) та ансамблі дерев класифікації, такі як випадковий ліс (RF).

Метод опорних векторів (SVM), широко використовуваний контрольований метод навчання, генерує гіперплощину у просторі особливостей високого розміру, щоб максимізувати найбільшу мінімальну відстань між окремими позначеними опорними векторами [30].

Класифікація K-найближчих сусідів (KNN) є навчанням на основі екземплярів, класифікує об’єкти на основі тренувальних даних у просторі функцій, призначаючи мітку відповідно до найбільш домінуючого класу її найближчих сусідів (в даному прикладі, k дорівнює 3).

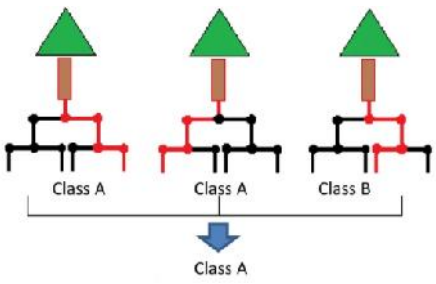

Рисунок 2.6 – Алгоритм випадкового лісу

Алгоритм випадкового лісу (рисунок 2.6) представляє собою множину дерев рішень, які в завданні класифікації приймають рішення шляхом голосування за більшістю. Для незалежної побудови всіх дерев використовується наступний алгоритм. Вибирається розмір підвибірки від навчальної вибірки, по якій будується дерево. Для побудови кожного розщеплення в дереві переглядається максимальна кількість випадкових ознак. Вибираються найкращі ознаки та розщеплення в дереві по цим ознакам по заздалегідь заданому критерію. Дерево будується, як правило, до вичерпання вибірки поки в листі не залишаться представники тільки одного класу.

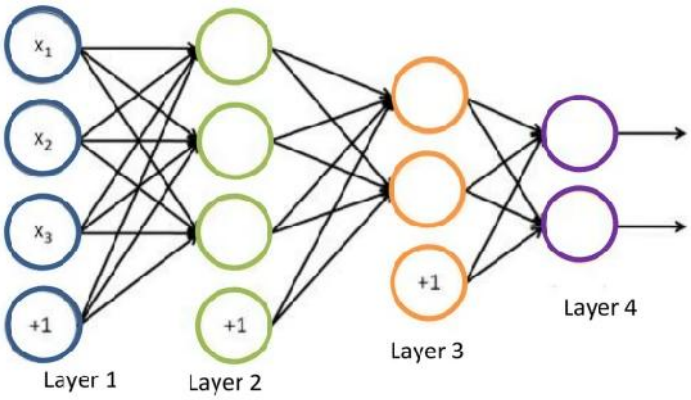

Рисунок 2.7 – Штучні нейронні мережі

Штучні нейронні мережі (ANN), вхід пов’язаний зі «штучним нейроном» за допомогою зважених зв’язків (рисунок 2.7). Після процесу підсумовування, зміщення та функції активації вхід поширюється на вихідний вузол для класифікації [31].

3 РЕАЛІЗАЦІЯ МЕТОДУ ОБРОБКИ ДАНИХ

3.1 Апаратна частина для отримання вхідних даних

Для реалізації запропонованого методу обробки даних енцефалограми було використано існуюче апаратне рішення на базі модуля ThinkGear ASIC (TGAM) від компанії NeuroSky для розпізнавання мозкових хвиль, який підключений до джерела живлення у вигляді акумулятора, що представлений блоком з двох батарейок типу AAA для забезпечення автономності (рисунок 3.1).

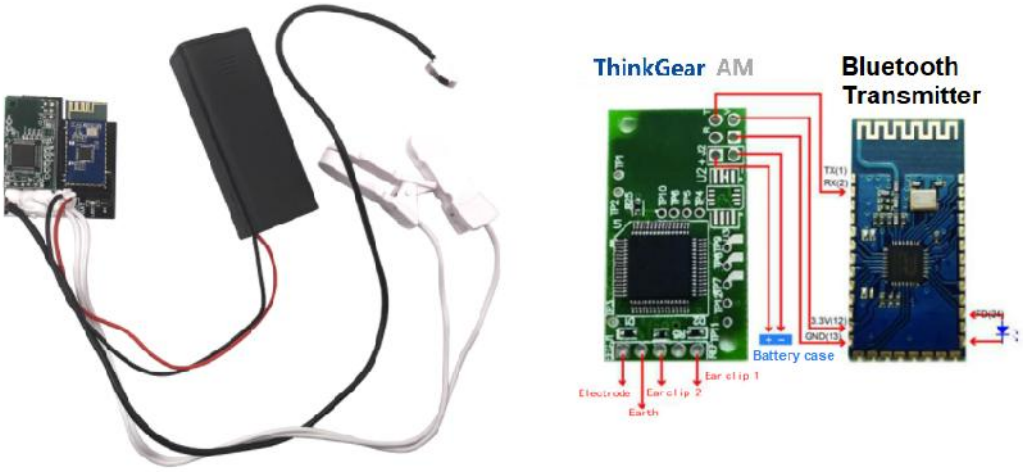

Рисунок 3.1 – Апаратна частина та схема підключення її компонентів

У якості модуля передачі даних для взаємодії з програмною частиною виступає Bluetooth-передавач, який підключено до TGAM-модуля за допомогою інтерфейсу UART (Serial).

Важливим компонентом для збору ЕЕГ виступають електроди. Особливістю зазначеного апаратного рішення є наявність лише одного сухого електроду, який розміщують на лобі, без використання струмопровідного гелю.

Таке рішення обумовлено відносно невеликою ціною модуля мозкових хвиль у той час як дорогі модулі мають кілька електродів, тому чутливість модуля порівняно невелика, що може накладати обмеження на подальший аналіз.

3.2 Отримання необроблених даних від апаратної частини

Модуль TGAM передає свої цифрові дані у вигляді асинхронного послідовного потоку байтів, який передається у вигляді ThinkGear-пакетів, кожен з яких необхідно розібрати та проаналізувати для правильного вилучення даних. Формат кожного пакету складається з трьох частин: заголовок пакета, дані, контрольна сума даних (рисунок 3.2).

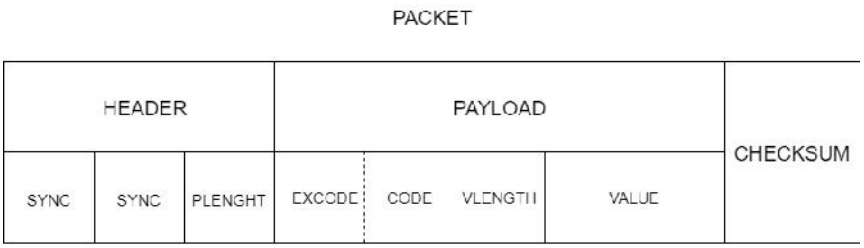

Рисунок 3.2 – Структура пакета у комунікаційному протоколі ThinkGear

Заголовок пакету складається з трьох байтів: два байти синхронізації, за якими слідує байт розміру корисних даних PLENGTH. Два байта синхронізації використовуються для сигналізації про початок нового пакету і представляють собою байти зі значенням 0xAA (десятичне 170).

Довжина синхронізації становить два байти замість одного, щоб зменшити ймовірність того, що байти синхронізації, які зустрічаються в пакеті, можуть бути помилково прийняті у якості початку пакету. Хоча два наступних байта все ще можуть з’явитися в пакеті, що призводить до спроби парсера надати синтаксичний аналіз середини пакету як початку пакету. Таким чином, PLENGTH та CHKSUM разом гарантують, що неправильно

синхронізований пакет ніколи не буде випадково інтерпретовано як дійсний пакет. Байт PLENGTH вказує довжину в байтах розділу пакета даних і може мати будь-яке значення від 0 до 169. Байт CHKSUM необхідно використовувати для перевірки цілісності безпосередньо даних пакета. Контрольна сума даних визначається як підсумовування всіх байтів пакетних даних, вибірка найнижчих 8 бітів від попереднього результату та виконання бітової інверсії для цих найнижчих 8 біт.

Після перевірки контрольної суми пакета байти даних можна проаналізувати. Вони складаються з безперервної серії значень даних, кожен з яких міститься в серії байтів, які називають DataRow. Кожен рядок даних містить інформацію про те, що представляє значення даних, довжину значення даних і байти самого значення даних. Таким чином, щоб проаналізувати дані, потрібно проаналізувати кожен рядок даних з нього, поки не будуть проаналізовані всі байти даних.

Кожен DataRow складається з байтів коду, довжини, значення. DataRow може починатися з нуля або більше байтів, маючи при цьому розширений код, який представлений байтами зі значенням 0x55. Кількість байтів EXCODE вказує на рівень розширеного коду. Рівень розширеного коду, у свою чергу, використовується разом із байтом CODE, щоб визначити, який тип значення даних містить цей рядок даних. Для отримання необроблених даних виконується збір DataRow з кодом 0x80, який складається з двох байтів, що являє собою одну необроблену вибірку хвилі. Його значення є 16-бітовим цілим числом зі знаком, яке коливається від -32768 до 32767.

Перший байт значення представляє старші біти значення двійкового компліменту, а другий байт представляє молодші біти. Щоб відновити повне необроблене значення хвилі, виконується зсув першого байту вліво на 8 біт та побітово АБО на другий байт [32].

Отриманий алгоритм для збору необроблених даних наведено нижче:

Крок перший. Виконання читання байтів з потоку, поки не зустрінеться байт SYNC, який дорівнює 0xAA.

Крок другий. Читання наступного байту та перевірка, що він також є байтом SYNC. Якщо це не байт SYNC, необхідно повернутись до кроку 1. В іншому випадку перейти до кроку 3.

Крок третій. Читання наступного байту із потоку як PLENGTH. Якщо PLENGTH дорівнює 0xAA (десятичне 170), то повторення кроку 3. Якщо значення PLENGTH перевищує 170, повернення до кроку 1. В іншому випадку перейти до кроку 4.

Крок четвертий. Читання PLENGTH байту з PAYLOAD потоку байтів, зберігаючи їх у сховищі. Підсумування кожного байту під час його читання шляхом збільшення акумулятора контрольної суми.

Крок п’ятий. Взяття найнижчих 8 біт акумулятора контрольної суми та їх інвертування.

Крок шостий. Читання наступного байту із потоку як байт CHKSUM. Якщо CHKSUM не відповідає обчисленій сумі, то ігнорувати дані, вивести попереджуюче повідомлення, перейти до кроку 1. В іншому випадку зберегти отримані дані. Якщо необхідна кількість даних зібрана, то повернути результат, у іншому випадку – перейти до кроку 1.

Завдяки загальному алгоритму для збору необроблених даних було виконано його реалізацію на мові Python з використанням бібліотеки PySerial (лістинг 3.1).

Лістинг 3.1 – Реалізований алгоритм читання потоку байт

```
input_bytes = serial_obj.read(bytes_to_read)
bytes_iterator = iter(input_bytes)
while values_limit > 0:
    first_byte = next(bytes_iterator)
    second_byte = next(bytes_iterator)
    if first_byte != BYTE_SYNC or second_byte != BYTE_SYNC:
        continue
    packet_length = next(bytes_iterator)
    while packet_length == ByteCodes.SYNC:
        packet_length = next(bytes_iterator)
    if packet_length > ByteCodes.SYNC:
        continue
```

```
packet_data = []
packet_checksum = 0x00
while packet_length > 0:
    packet_code = next(bytes_iterator)
    packet_checksum += packet_code
    packet_length -= 1
    if packet_code == ByteCodes.RAW_VALUE:
        row_len = next(bytes_iterator)
        low_byte = next(bytes_iterator)
        high_byte = next(bytes_iterator)
        packet_data.append(_raw_value(low_byte, high_byte))
        packet_checksum += row_len + low_byte + high_byte
        packet_length -= 3
packet_checksum &= 0xFF
packet_checksum = ~packet_checksum & 0xFF
received_checksum = next(bytes_iterator)
if received_checksum == packet_checksum:
    result_data.extend(packet_data)
values_limit -= len(packet_data)
```

Для читання лише обмеженої кількості байт було передбачено окрему змінну, яка була визначена за рамками наведеного прикладу. Результат читання збережено у змінну result_data, значення якої зберігається у вигляді csv файлів.

3.3 Цифрова фільтрація сигналу

Цифрова фільтрація є поширеним кроком попередньої обробки при аналізі даних ЕЕГ. Типова практика обробки сигналів ЕЕГ полягає у застосуванні високочастотного фільтра для фільтрації повільних частот менше 0,1 Гц або навіть 1 Гц та фільтра низьких частот для фільтрації частот вище 40 або 50 Гц. Хоча фільтрація може бути надзвичайно корисною для зменшення або відокремлення шуму від потрібного сигналу, вона також змінює вхідний сигнал. Численні дослідження демонструють, що фільтрація може впливати та змінювати форму або часову структуру сигналів ЕЕГ або дані, пов’язані з потенційними подіями [27].

Фільтри можуть бути сконструйовані так, щоб вони мали кінцеву імпульсну характеристику (FIR) або нескінченну імпульсну характеристику

(IIR). Імпульсна характеристика – це те, як фільтр обробляє одиничний імпульсний сигнал у часовій області (те, що він робить для структури сигналу), а його перетворення Фур'є відома як частотна характеристика.

FIR-фільтр має імпульсну характеристику протягом кінцевої тривалості, після чого вихідний сигнал стає нульовим та виконує рівні затримки на всіх частотах (також відомий як лінійна фазова характеристика). На відміну від цього, IIR-фільтри (також відомі як рекурсивні фільтри) мають нескінченну імпульсну характеристику, де частина виходу фільтра використовується як зворотний зв'язок. Це призводить до неоднакових затримок на різних частотах (нелінійні фазові характеристики). Це означає, що вихідний сигнал зміщується в часі щодо входу, при цьому деякі частотні складові зміщуються більше, ніж інші. Основною перевагою IIR-фільтрів є те, що вони обчислювально ефективніші [27].

Інший аспект у проектуванні фільтра – це напрямок сигналу, який використовується як вхідний. Фільтри, які містять лише минулу та теперішню інформацію, відомі як причинно-наслідкові фільтри, тоді як фільтри, які залежать від минулого та майбутнього введення, відомі як непричинні чи неказуальні фільтри. Причинно-наслідковий фільтр дає відповідь після початку (t = 0). Непричинні фільтри викликають реакцію ще до початку через зворотну фільтрацію, а також спричиняють більші бокові всплески.

Причинна фільтрація досягається на практиці шляхом фільтрації даних один раз вперед, а потім знову назад, і може допомогти мінімізувати введення фазових затримок у сигнал. Оскільки акаузальний або причинний фільтр вимагає повних даних, це не можна робити у реальному часі (наприклад, причинно-наслідковий фільтр), а є операцією в автономному режимі, яка виконується після збору даних.

Рисунок 3.3 демонструє, як причинно-наслідковий фільтр викривляє форму сигналу, тоді як акаузальний чи непричинний фільтр зберігає форму сигналу, але він вносить відмінності у сигнал ще до його початку при t = 0, через зворотну фільтрацію [28].

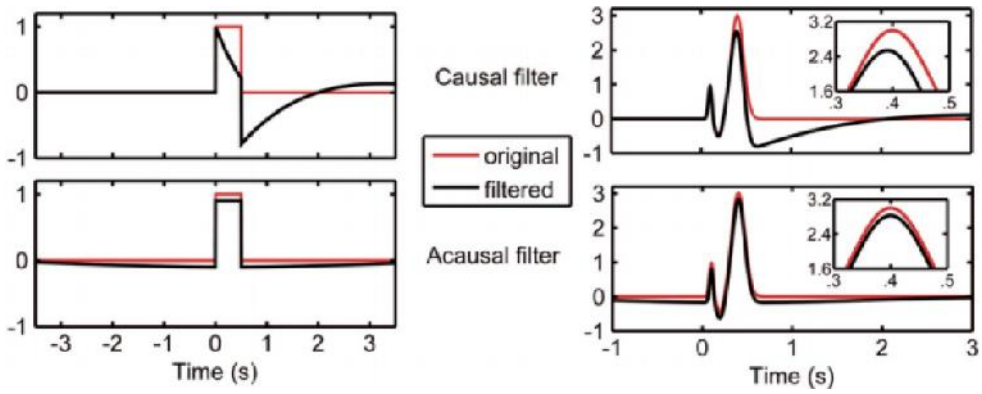

Рисунок 3.3 – Порівняння двох фільтрів

3.4 Визначення типів мозкових хвиль

Основним етапом аналізу ЕЕГ-даних є виявлення різних типів мозкових хвиль: альфа, бета, гамма, дельта і тета хвилі. Кожен з цих типів хвиль має власний діапазон частот, може бути виявлений з неопрацьованого ЕЕГ-сигналу та характеризує різний стан людського мозку. Після отримання необроблених даних від апаратної частини виконується їх нормалізація, отримання періодограми Уелча для підрахунку спектральної густини потужності, визначення сили діапазону хвиль мозку за допомогою правила Сімпсона для отримання площі під кривою графіка спектральної густини потужності.

3.4.1 Оцінка спектральної густини потужності за методом Уелча

Оцінка спектральної потужності є однією з найважливіших областей досліджень і застосувань у цифровій обробці сигналів. Методи оцінки спектральної потужності є непараметричними або класичними методами, які засновані на періодограмі. Оцінка спектральної потужності є спектральною характеристикою сигналів, які характеризуються як випадкові процеси. Автокореляційна функція випадкового сигналу є відповідним статистичним середнім, яке буде використовуватися для характеристики випадкових

сигналів у часовій області, а спектральна густина потужності є перетворенням Фур’є автокореляційної функції, яке забезпечує перетворення з часової області в частотну область.

Спектральна потужність описує розподіл енергії часового ряду в частотній області. Енергія є величиною, що має реальну величину, тому спектральна потужність не містить фазової інформації. Оскільки часовий ряд може містити неперіодичні або асинхронні періодичні компоненти сигналу, спектральна потужність часового ряду зазвичай вважається неперервною функцією частоти.

Непараметричний метод Уелча, в якому потужність будь-якого вхідного сигналу є оцінкою на різних частотах, є вдосконаленим методом оцінки спектру періодограми (метод оцінки автокореляції кінцевої довжини сигналу), коли відношення сигналу до шуму є високим і зменшує шум в обмін на зниження роздільної здатності по частоті. Періодограму легко обчислити і вона має обмежені можливості для точної оцінки спектру потужності. Даний метод упереджений при роботі з кінцевими вікнами. Зі збільшенням довжини даних швидкість коливань у цьому також збільшується. Покращеним варіантом періодограми є метод Уелча.

Метою оцінки спектральної потужності є опис розподілу (по частоті) потужності, що міститься в сигналі, на основі кінцевого набору даних. Оцінка спектральної потужності корисна для різноманітних застосувань, включаючи виявлення сигналів у широкосмуговому шумі. Аналіз спектру знайшов застосування при вивченні інженерних сигналів зв’язку, пов’язаних з подіями або стимульованих реакцій електроенцефалограми людини в діагностиці захворювань мозку, інших біологічних сигналів, метеорологічного контролю промислових процесів і вимірювання спектрів шуму для оптимальної лінійної фільтрації.

Шкідливий вплив спектральних викривлень можна мінімізувати, якщо переглядати дані за допомогою відповідної віконної функції. Вибіркове значення даних множиться від точки до точки на вибіркове значення обраної

віконної функції. Для вибору відповідного вікна враховуються еквівалентна смуга пропускання шуму, посилення обробки, втрати обробки в найгіршому випадку та мінімальна роздільна здатність. У кореляції перекриття усереднення спектру віконних даних безпосередньо призводить до значно покращеної оцінки спектру.

Методи оцінки спектральної потужності можна розділити на два типи: параметричний і непараметричний. Параметричні або некласичні методи аналізований процес замінюють відповідною моделлю з відомим спектром.

Непараметричні або класичні методи не роблять жодних припущень щодо процесу генерування даних. Параметричні методи базуються на параметричних моделях часового ряду, таких як моделі AR, моделі ковзного середнього та моделі авторегресійного ковзного середнього. Тому параметричні методи також відомі як методи на основі моделі.

Щоб оцінити спектральну густину потужності часового ряду за допомогою параметричних методів, спочатку потрібно отримати параметри моделі часового ряду.

У непараметричних методах, які включають метод періодограми, метод Уелча та методи Блекмана-Тьюкі, ґрунтуються на дискретному перетворенні Фур’є. У цих методах не потрібно отримувати параметри часового ряду. Всі ці методи мають перевагу можливої реалізації з використанням швидкого перетворення Фур’є, але з недоліком у разі малої довжини даних з обмеженою роздільною здатністю по частоті.

Параметричні методи, з іншого боку, можуть забезпечити високу роздільну здатність на додаток до обчислювальної ефективності. Найбільш поширеним параметричним підходом є виведення спектру з параметрів авторегресійної моделі сигналу.

Метод Уелча є покращеною оцінкою, отриманою шляхом зміни методу Бартлетта у двох аспектах. По-перше, сегментам даних у методі Уелча дозволяється перекриватися. По-друге, перед обчисленням кожного сегмента даних застосовується віконна функція для обчислення періодограми.

Метод складається з поділу даних часового ряду на сегменти (можливо, що перекриваються), обчислення модифікованої періодограми кожного сегмента, а потім усереднення оцінок спектральної густини потужності [33].

При реалізації методу Уелча було використано бібліотеку scipy у модулі signal, яка містить в собі рішення з обробки сигналів (лістинг 3.2).

Лістинг 3.2 – Отримання періодограми Уелча

```
def get_welch_periodogram(input_data):
          seg_length = (input_data['Time'].max()
              * input_data['Time'].size)

          freq_values, psd_values = signal.welch(
              normalized_data['Value'],
              normalized_data['Time'].size,
              nperseg=seg_length)

          return pd.DataFrame(list(zip(freq_values,
                                       psd_values)),
              columns=['Frequency', 'PSD'])
```

Перед використанням функції welch виконується обчислення розміру сегмента. Зменшення розміру nperseg дозволяє розділити вхідні дані на більше сегментів, що дозволяє отримати більш надійну оцінку. Збільшення розміру nperseg покращує частотну роздільну здатність результату. У поточній реалізації розмір сегмента дорівнює розміру вхідного масиву даних, тобто дорівнює 512.

Вхідні дані та збереження періодограми представлено у вигляді об’єкта DataFrame з бібліотеки Pandas. Кожен DataFrame є двовимірною структуру даних, що є схожою на звичайну таблицю. Дані, отримані з періодограми Уелча, будуть у подальшому використані для визначення конкретних діапазонів мозкових хвиль, через виконання обчислення площі під кривою графіка.

Для обчислення площі під кривою графіка доцільним є використання правила Сімпсона.

3.4.2 Використання правила Сімпсона

Інтегрування функцій f(x) за правилами трапецій можна інтерпретувати як заміну первісної функції f(x) деякою кусочно-лінійною функцією (після розбиття загального інтервалу інтеграції на безліч відрізків, на кожному з яких змінюється пряма лінія), від якої і вираховується близьке значення інтеграла. Помилка методу в цьому випадку визначається грубістю запропонованого способу апроксимації функцій. Наближення первісної функції f(x) на відрізках не лінійними функціями, а поліномами більш високих порядків, дозволяє зменшити помилку методу інтегрування. Для правила Сімпсона в якості функції обрана парабола, за допомогою якої забезпечується приближення вихідної функції на частинках відрізків інтегрування [34].

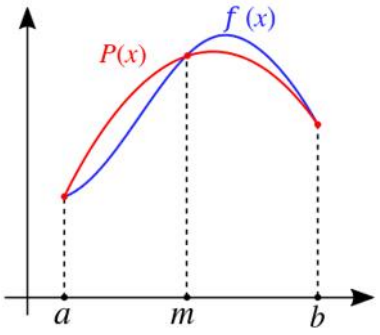

Рисунок 3.4 – Апроксимація функції f(x) (синій графік) квадратичним поліномом P(x) (червоний)

Формулою Сімпсона (3.1) називається інтеграл від інтерполяційного багаточлена другого ступеня на відрізку [a,b]:

$$\int_a^b f(x)d \ = \int_a^b p_2(x)d \ = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right), \qquad (3.1)$$

де $f(a)$, $f\left(\frac{a+b}{2}\right)$ і $f(b)$ значення функції у відповідних точках (на кінцях відрізка та у його середині).

Застосування правила Сімпсона було виконано за допомогою бібліотеки scipy та функції simpson для обчислення площі під кривою графіка (лістинг 3.3).

Лістинг 3.3 – Застосування правила Сімпсона

```
def get_band_strength(welch_periodogram):
    psd_values = welch_periodogram['PSD']
    freq_values = welch_periodogram['Frequency']
    band_values = []
    auc_values = []
    for band_name, band_range in BAND_RANGE.items():
        low_value, high_value = band_range
        band_result = (frequency_values >= low_value)
            & (frequency_values < high_value)
        band_psd_values = psd_values[band_result]
        band_freq_values = freq_values[band_result]
        auc = 0.0
        if (not band_psd_values.empty
                and not band_frequency_values.empty):
            auc =integrate.simpson(band_psd_values,
                                   band_freq_values)
        band_values.append(band_name)
        auc_values.append(auc)
    total_auc = sum(auc_values)
    normalized_auc_values = np.array(auc_values) / total_auc
    return pd.DataFrame(list(zip(
        band_values, normalized_auc_values)),
        columns=['Band', 'AUC'])
```

Дані, отримані з періодограми Уелча, були додатково проаналізовані для визначення конкретних діапазонів мозкових хвиль, шляхом розрахунку площі під кривою для кожного заданого діапазону частот для хвиль (альфа, бета, гамма, дельта, тета), після чого отримані значення були нормовані.

3.5 Видалення аномальних викидів у ході попередньої обробки даних

Аномальні значення в деякому наборі даних є зразками, які значно відрізняються від основного потоку даних. Викиди можуть мати безліч причин, наприклад: помилка виміру або введення, пошкодження даних,

справжнє спостереження за викидами. Немає точного способу визначення та ідентифікації викидів загалом через специфіку кожного набору даних. Натомість існує необхідність додаткової обробки вхідних даних з метою вирішити, чи є те чи інше значення викидом, чи ні, але може бути проблема у визначенні таких значень, що може з необережності призвести до видалення чи зміни звичайних даних, особливо якщо розмір вибірки невеликий.

Але можна використовувати статистичні методи виявлення спостережень, які здаються рідкісними чи малоймовірними з урахуванням наявних даних. Виявлення викидів та невірних даних у наборі даних є однією з найскладніших частин фільтрації даних.

Якщо відомо, що розподіл значень у вибірці є гаусовим (нормальний розподіл) або є подібним до гаусового, тоді можна використовувати середньоквадратичне відхилення вибірки як порогового значення для виявлення викидів. Розподіл Гауса має властивість, що стандартне відхилення від середнього може використовуватися для надійного підсумовування відсотка значень у вибірці.

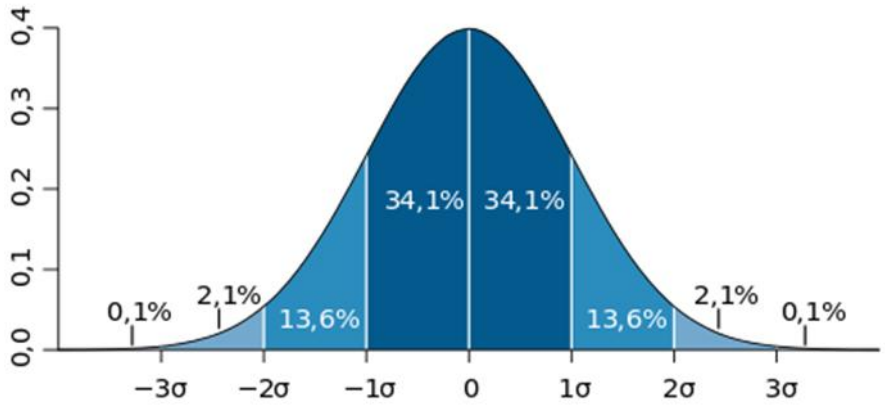

Рисунок 3.5 – Графік густини ймовірності нормального розподілу і відсоток попадання випадкової величини на відрізки

Наприклад, при генерації 10 тисяч випадкових чисел, взятих з розподілу Гауса із середнім значенням 50 і стандартним відхиленням 5, в межах одного

стандартного відхилення середнє значення охоплюватиме 68% даних. Таким чином, якщо середнє значення дорівнює 50, а стандартне відхилення дорівнює 5, то всі дані у вибірці між 45 і 55 будуть становити близько 68% вибірки даних. Можна охопити більше вибірки даних, якщо розширити діапазон наступним чином:

- одне середньоквадратичне відхилення від середнього: 68%;
- два середньоквадратичні відхилення від середнього: 95%;
- три середньоквадратичні відхилення від середнього: 99,7%.

Значення, що виходить за межі трьох стандартних відхилень, є частиною розподілу, але це малоймовірна або рідкісна подія, що становить приблизно 1 на 370 зразків. Три стандартні відхилення від середнього – це звичайне обмеження на практиці для виявлення викидів у гауссівському або подібно до гауссівського розподілу. Для менших вибірок даних можна використовувати значення двох стандартних відхилень (95%), а для більших вибірок, можна використовувати значення чотирьох стандартних відхилень (99,9%) [35].

3.6 Прогнозування за допомогою алгоритму K-найближчих сусідів

KNN робить прогнозування безпосередньо за допомогою навчального набору даних. Передбачення для нового екземпляру здійснюються шляхом пошуку в усьому навчальному наборі для K найбільш подібних екземплярів (сусідів) і підсумовування вихідної змінної для цих K екземплярів. Для регресії це може бути середня вихідна змінна, у класифікації це може бути значенням класу режиму (або найбільш поширеним).

Щоб визначити, які з K екземплярів у наборі навчальних даних найбільш схожі на новий вхід, використовується міра відстані. Для вхідних змінних з дійсним значенням найпопулярнішою мірою відстані є евклідова відстань. Евклідова відстань обчислюється як квадратний корінь із суми квадратів різниць між новою точкою (x) та існуючою точкою (x_i) за всіма вхідними атрибутами j.

$$E(x, x_i) = \sqrt{\sum_{j=1}^{n}(x_j - x_i)^2}, \quad (3.2)$$

Інші популярні міри дистанції включають:

- відстань Хеммінга: обчислення відстані між двійковими векторами (або більше);
- відстань Манхеттена: обчислення відстані між реальними векторами, використовуючи суму їх абсолютної різниці;
- відстань Мінковського: узагальнення евклідової та манхеттенської відстані.

Існують також інші міри відстаней, які можна використовувати, наприклад, Танімото, Жаккард, Махаланобіса та косинусна відстань. Найкращий показник відстані визначається на основі властивостей вхідних даних. Якщо немає впевненості у використанні конкретного методу відстані, то можна поекспериментувати з різними показниками відстані та різними значеннями K та визначити, який метод розрахунку відстані призводить до найбільш точних моделей.

Обрання евклідової міри відстані є хорошим вибором, якщо вхідні змінні подібні за типом (наприклад, ширина, довжина, висота). Використання відстані Манхеттена є придатним, якщо вхідні змінні не схожі за типом (наприклад, вік, стать, зріст тощо).

Значення для K можна знайти шляхом налаштування алгоритму. Бажано спробувати багато різних значень для K (наприклад, значення від 1 до 21) і побачити, що найкраще підходить для конкретних даних. Складність обчислень KNN зростає із збільшенням розміру навчального набору даних. Для великих навчальних наборів KNN можна зробити стохастичним, взявши вибірку з навчального набору даних, з якої обчислюються K-найбільш подібні екземпляри. Через широку сферу використання алгоритму KNN, різні дисципліни мають різні назви для нього, наприклад:

- навчання на основі екземплярів: необроблені навчальні екземпляри використовуються для прогнозування. Тому KNN часто називають навчанням на основі екземплярів або навчанням на основі випадків (де кожен навчальний екземпляр є випадком із проблемної області);

- ліниве навчання: вивчення моделі не потрібно, і вся робота виконується в той час, коли запитується прогноз. Тому KNN часто називають алгоритмом відкладеного навчання;

- непараметричний: KNN не робить припущень щодо функціональної форми проблеми, що вирішується. Тому KNN називають непараметричним алгоритмом машинного навчання [36].

Підготовка даних для KNN потребує зміну масштабу даних: KNN працює набагато краще, якщо всі дані мають однаковий масштаб, тому нормалізація даних до діапазону [0, 1] є доречним кроком попередньої обробки. Також можна стандартизувати дані, якщо вони мають розподіл по Гауссу.

Відсутність даних свідчить про те, що відстань між вибірками не може бути розрахована. Ці зразки можна було б виключити або врахувати відсутні значення.

KNN підходить для даних меншої розмірності. Можливе його використання на даних великої розмірності (сотні чи тисячі вхідних змінних), але при цьому даний метод може не працювати так добре, як інші методи. KNN може отримати вигоду від вибору ознак, який зменшує розмірність вхідного набору даних.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

4.1 Отримання необроблених даних від апаратної частини

Для читання даних від апаратної частини було використано алгоритм, який було наведено у розділі 3.2, а також Python-бібліотеку PySerial, яка реалізує доступ до послідовного порту на Windows, OSX, Linux, BSD, можливо, будь-яку POSIX-сумісну систему. При цьому необхідне попереднє з’єднання між двома пристроями по Bluetooth.

Зберігання отриманих даних під час виконання Python-скрипту проводилось за допомогою бібліотеки Pandas, яка підтримує збереження отриманого об’єкту DataFrame у вигляді csv файла, який містить в собі дві колонки, Time і Value. Згідно зі специфікацією апаратної частини, видача необроблених даних ЕЕГ виконується зі швидкістю 512 біт у секунду, тому для спрощення подальшого аналізу DataFrame з отриманим результатом при збереженні даних ділить їх на 512 значень, таким чином отримуючи набір csv файлів, яка дорівнює кількості секунд протягом яких виконується збір даних. При цьому це значення передбачено програмною частиною, тому можна динамічно змінювати кількість отриманих фрагментів даних для більш зручного дослідження.

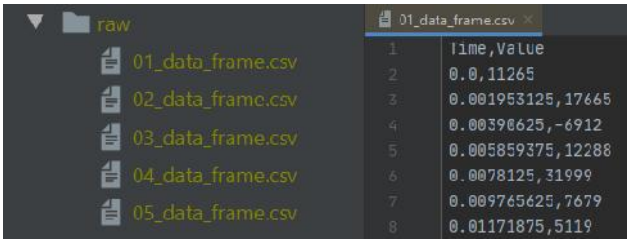

Рисунок 4.1 – Фрагмент одного з отриманих CSV файлів з необробленими даними

Для візуалізації отриманих даних на графіках та гістограмах була використана бібліотека Matplotlib, яка є гнучкою, легко конфігурованою та разом із NumPy, SciPy та IPython надає можливості, подібні до MATLAB. На сьогоднішній день бібліотека працює з декількома графічними бібліотеками, включаючи wxWindows та PyGTK. На рисунку 4.2 наведено графік одного з фрагментів необробленого файлу.

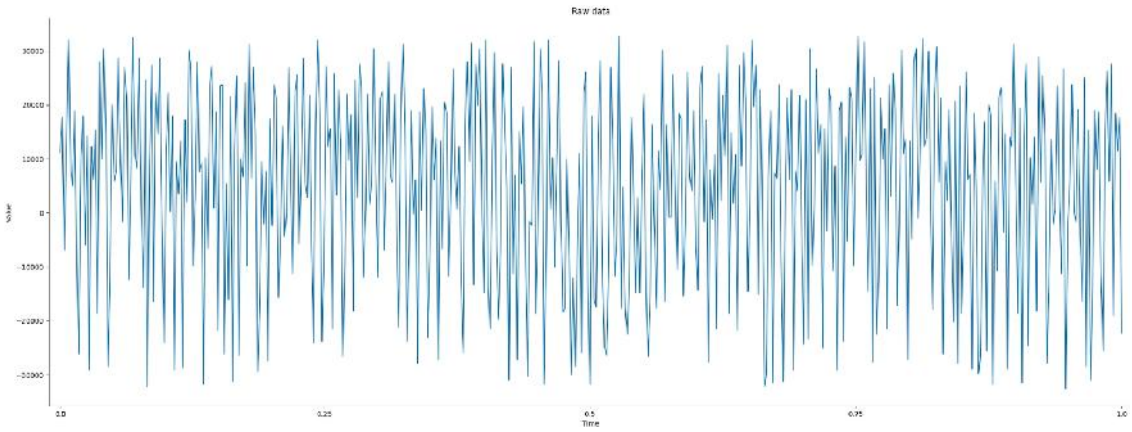

Рисунок 4.2 – Отримані необроблені дані на графіку

Результатом даного етапу є підготовлені для обробки дані, які будуть використані у подальшому.

4.2 Аналіз отриманих даних про типи мозкових хвиль

Першим кроком обробки даних енцефалограми було виконання їх перетворення шляхом нормалізації (4.1):

$$n \qquad \_v \qquad = 2 \cdot \frac{in \quad \_v \quad -m \quad \_v}{m \quad \_v \quad -m \quad \_v} - 1, \qquad (4.1)$$

де min_value, max_value є мінімальними і максимальними значеннями зі всього набору даних. Таким чином було отримано нормалізовані дані у діапазоні [-1:1] (рисунок 4.3).

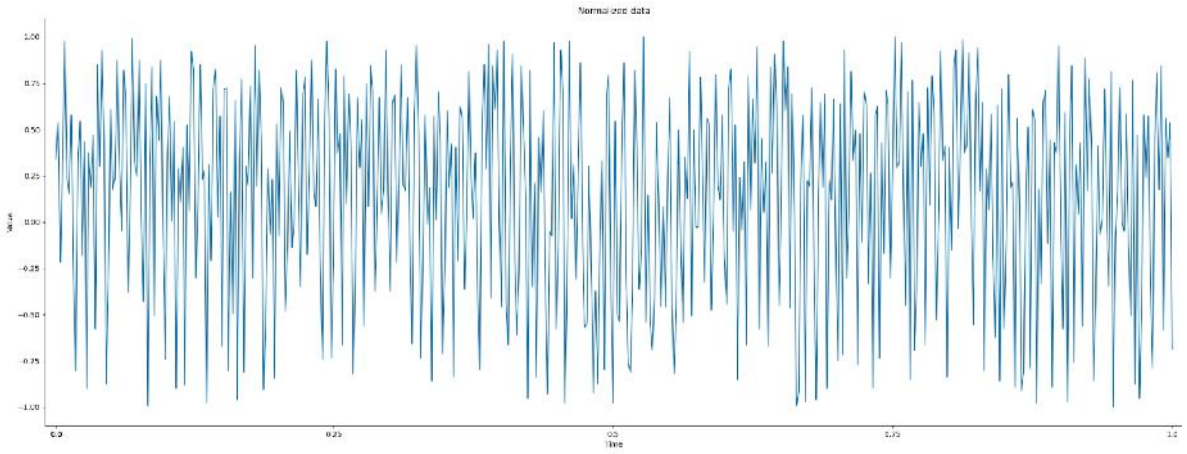

Рисунок 4.3 – Графік з нормалізованими даними

За допомогою бібліотеки SciPy для мови програмування Python були застосовані відповідні методи обробки сигналів. Дана бібліотека використовується для вирішення математичних, наукових, інженерних та технічних завдань. Це дозволяє користувачам маніпулювати даними та візуалізовувати їх за допомогою широкого спектру команд високого рівня Python. SciPy побудовано на основі бібліотеки NumPy.

Для застосування методу Уелча було використано функцію welch з зазначеної бібліотеки, а її результатом є періодограма яка наведена на графіку нижче (рисунок 4.4).

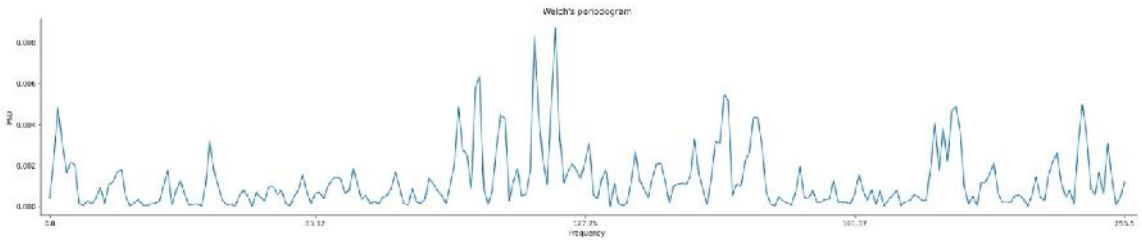

Рисунок 4.4 – Отримана періодограма Уелча

Наступним етапом є визначення сили діапазону альфа, бета, гамма, дельта і тетта хвиль мозку за допомогою правила Сімпсона для отримання площі під кривою графіка спектральної густини потужності. Для цього була

використана функція simpson із модуля integrate з бібліотеки SciPy. Кожен з зазначених типів хвиль має власний діапазон частот, тому їх необхідно було окремо визначити (лістинг 4.1).

Лістинг 4.1 – Типи хвиль мозку з вказаним діапазоном частот

```
BAND_RANGE = {
    'Delta': (0.05, 3),
    'Theta': (3.05, 8),
    'Alpha': (8.05, 12),
    'Low-Beta': (12.05, 15),
    'Mid-Beta': (15.05, 18),
    'High-Beta': (18.05, 30),
    'Low-Gamma': (30.05, 39.75),
    'Mid-Gamma': (39.80, 49.75),
}
```

Маючи періодограму Уелча з частотами та визначений діапазон частот для кожного з типів хвиль можна силу кожного діапазону на зобразити її на гістограмі за допомогою зазначеної функції hist з бібліотеки Matplotlib, результат роботи якої наведено на графіку нижче (рисунок 4.5).

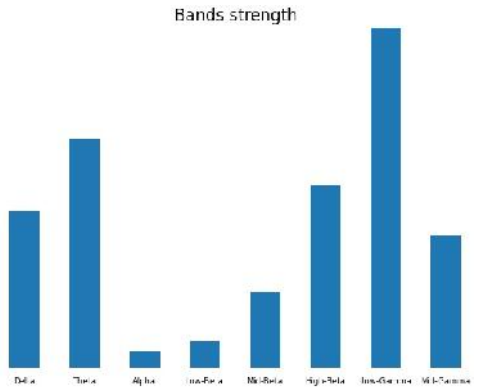

Рисунок 4.5 – Гістограма з визначеними силами для кожного діапазону

Результатом даного аналізу є виявлення різних типів мозкових хвиль: альфа, бета, гамма, дельта і тета хвилі, які були виявлені з неопрацьованого ЕЕГ-сигналу, характеризують різний стан людського мозку.

4.3 Аналіз результату прогнозування стану очей людини

Через відносну обмеженість апаратної частини, яку було зазначено у розділі 3.1, для аналізу прогнозування стану очей було використано вхідні необроблені ЕЕГ-дані з відкритих джерел від апаратної частини, яка містить в собі також і визначений закритий чи відкритий стан очей людини. Апаратна частина представлена у вигляді нейрогарнітури від компанії Emotiv під назвою Emotiv EEG Neuroheadset, яка містить в собі 14 електродів, завдяки чому надає більш точні дані порівняно з використаними апаратними засобами.

Необроблені апаратні дані представлені у вигляді файла з ARFF форматом (рисунок 4.6), який можна легко перевести у CSV формат видалив вказані заголовки з файлу. Безпосередньо самі дані представлені у вигляді 14 значень відповідні до кількості електродів та стану очей у вигляді 0 чи 1.

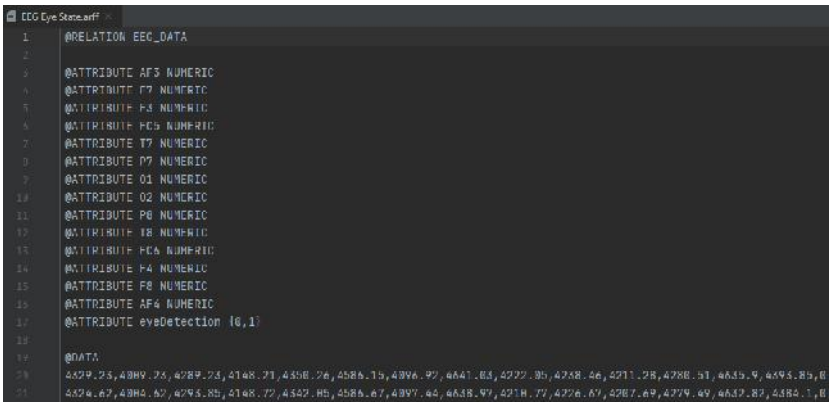

Рисунок 4.6 – Вхідний ARFF файл з даними

Використовуючи вже зазначену бібліотеку Matplotlib, було виконано візуалізацію вхідних даних на 15 відповідних графіках (рисунок 4.7).

Завдяки вказаній візуалізації можна побачити аномальні викиди, які перешкоджають подальшому аналізу з використанням засобів машинного навчання, тому необхідно передбачити відповідну фільтрацію. Для цього було використано метод середньоквадратичного відхилення для відсіювання значень, які не відповідають обчисленому діапазону.

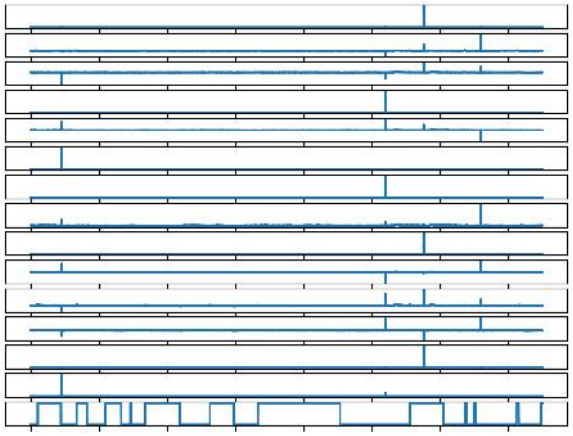

Рисунок 4.7 – Візуалізація вхідних даних разом зі станом очей

Для обчислення середньоквадратичного відхилення було використано функцію std з бібліотеки numpy, а функцію mean – для знаходження середнього арифметичного значення з відповідного масиву значень з кожного з електродів. Відповідний код наведено у лістингу 4.2.

Лістинг 4.2 – Запропонована фільтрація даних

```
data_mean, data_std = mean(data), std(data)
cut_off = data_std * 3
lower, upper = data_mean - cut_off, data_mean + cut_off
filtered_data = [x for x in data if x > lower and x < upper]
```

Виконавши метод середньоквадратичного відхилення можна побачити результат на рисунку 4.8.

Наявність лише одного файлу з ЕЕГ-даними призводить до вивчення параметрів функції прогнозування та тестування її на тих самих даних, що є методологічною помилкою: модель, яка буде повторювати мітки зразків, які вона щойно побачила, буде мати ідеальну оцінку, але не зможе виконати прогноз на невидимих даних. Така ситуація свідчить про перенавчання системи.

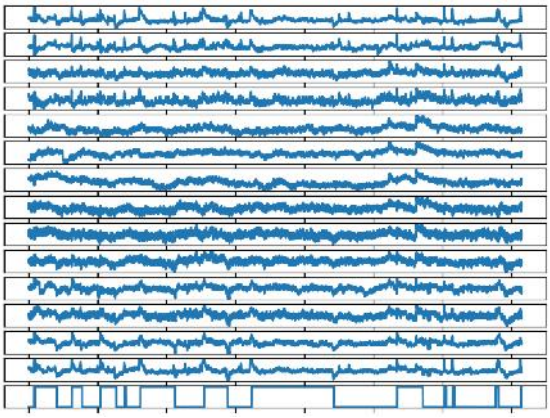

Рисунок 4.8 – Візуалізація даних після відповідної фільтрації

Щоб уникнути перенавчання, під час проведення експерименту з машинним навчанням зазвичай використовується частина наявних даних у вигляді тестового набору [37].

Для цього було використано модель KFold з бібліотеки sklearn для розділення вхідних даних на навчальні та тестові вибірки для вирішення завдання класифікації з даними, які не були використані під час навчання.

Для прогнозування стану очей було використано класифікацію на основі алгоритму K-найближчих сусідів при K=3, яка приймає дані для тренування та прогнозування, використовуючи програмні засоби бібліотеки sklearn. Результати прогнозування наведені на рисунку 4.9.

```
che@DESKTOP-7I8GH9I:/mnt/c/Users/Che/Projects/eeg_prediction$ python3 prediction_test.py
>0.970
>0.975
>0.978
>0.977
>0.973
>0.979
>0.978
>0.976
>0.974
>0.969
Final Score: 0.975
```

Рисунок 4.9 – Результати точності прогнозування

Для оцінки точності прогнозування було використано функцію accuracy_score (лістинг 4.3).

У класифікації за кількома мітками ця функція обчислює точність підмножини: набір міток, передбачених для вибірки, повинен точно відповідати до очікуваного набору міток.

Лістинг 4.3 – Прогнозування за алгоритмом K-найближчих сусідів

```
data = pd.read_csv('eeg_data.csv', header=None)
input_values = data.values
scores = []
kfold = KFold(10, shuffle=True, random_state=1)
for train_ix, test_ix in kfold.split(input_values):
    train_x = input_values[train_ix, :-1]
    train_y = input_values[train_ix, -1]
    test_x = input_values[test_ix, :-1]
    test_y = input_values[test_ix, -1]
    model = KNeighborsClassifier(n_neighbors=3)
    model.fit(train_x, train_y)
    pred_res = model.predict(test_x)
    score = accuracy_score(test_y, pred_res)
    scores.append(score)
    print('>%.3f' % score)
print('Final Score: %.3f' % (mean(scores)))
```

Змінні train_ix та test_ix являють собою масив індексів з вхідного набору даних input_values та завдяки цим змінних отримується набір даних зі значеннями електроенцефалограми (train_x, test_x відповідно) та значення стану очей (train_y, test_y відповідно). Значення стану очей дорівнюють останньому стовпцю з вхідних значень, а всі інші значення є відповідно ЕЕГ-даними.

Таким чином, було виконано аналіз результату прогнозування стану очей використовуючи вхідні необроблені ЕЕГ-дані з відкритого джерела, які було попередньо оброблено використовуючи метод середньоквадратичного відхилення та виконано прогнозування за допомогою алгоритму машинного навчання K-найближчих сусідів.

ВИСНОВКИ

Результатом роботи є створення методу обробки даних електроенцефалограми з використанням засобів машинного навчання. Отримання даних електроенцефалограми було забезпечено завдяки апаратним засобам ЕЕГ-модуля TGAM від компанії NeuroSky, який надає неопрацьований сигнал мозкових хвиль по Bluetooth для подальшої обробки програмною частиною.

У вигляді програмної частини виступають Python-скрипти, які надають можливість приймати необроблені дані по Bluetooth від апаратної частини та виконувати їх збереження у вигляді CSV файлів з обраним проміжком часу для збору.

Наступним етапом роботи виступає обробка збережених даних у вигляді Min-Max нормалізації, отримання періодограми Уелча для отримання спектральної густини потужності, визначення сили діапазону хвиль мозку за допомогою правила Сімпсона для отримання площі під кривою графіка спектральної густини потужності. Кожен тип хвиль мозку має свій діапазон та має відповідні назви: альфа, бета, дельта і тета.

Для прогнозування закритого або відкритого стану очей людини на основі зібраних апаратною частиною даних електроенцефалограми в якості засобу машинного навчання було використано класифікацію за допомогою алгоритму K-найближчих сусідів. Через відносну обмеженість визначеної апаратної частини було використано вхідні дані з відкритих джерел.

Останній етап роботи програмної частини виконує експорт отриманих необроблених та оброблених даних на графіках у вигляді PNG-зображень для аналізу отриманих результатів.

Отримане рішення є актуальним на поточний час та має широку сферу застосування: використовується у біометрії, сфері охорони здоров’я, в освітніх цілях, нейромаркетингу тощо.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Electroencephalogram [Електронний ресурс]. – Режим доступу : www/ URL: https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/electroencephalogram-eeg/ – 17.11.2021 р. – Загол. з екрану.

2. Top 14 EEG Hardware Companies [Електронний ресурс]. – Режим доступу : www/ URL: https://imotions.com/blog/top-14-eeg-hardware-companies-ranked/ – 16.11.2021 р. – Загол. з екрану.

3. What is EEG and How Does it Work? [Електронний ресурс]. – Режим доступу : www/ URL: https://imotions.com/blog/what-is-eeg/ – 11.11.2021 р. – Загол. з екрану.

4. Repovš, G. Dealing with Noise in EEG Recording and Data Analysis [Текст] / G. Repovš // Infor Med Slov. – 2010. – №15(1). – С. 18–25.

5. Kübler, A. Brain-computer interface technology: A review of the second international meeting [Текст] / A. Kübler // IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society. – 2003. – №11(2). – С. 94–109.

6. Chambayil, B., EEG Eye Blink Classification Using Neural Network [Текст] / B. Chambayil, R. Singla, R. Jha // Proceedings of the World Congress on Engineering. – 2010. – №1. – С. 2–5.

7. Teplan, M. Measurement science review [Текст] / M. Teplan // Fundamentals of EEG measurement. – 2002. – №2(2). – С. 1–11.

8. The Introductory Guide to EEG (Electroencephalography) [Електронний ресурс]. – Режим доступу : www/ URL: https://www.emotiv.com/eeg-guide/ – 15.11.2021 р. – Загол. з екрану.

9. Shravani, Sur. Event-related potential: An overview [Текст] / S. Shravani, V. K. Sinha // Industrial Psychiatry Journal. – 2009. – №18(1). – С. 70–73.

10. Thompson, T. EEG applications for sport and performance [Текст] / T. Thompson, T. Steffert, T. Ros, J. Leach, J. Gruzelier // Journal Methods. – 2008. –

№45(4). – С. 279–288.

11. Huang, S. The Application of EEG related [Текст] / S. Huang, H. Xiao // Procedia Environmental Sciences. – 2011. – №10. – С.1338-1342.

12. Gemein, L. A. W. Machine-learning-based diagnostics of EEG pathology [Текст] / L. A.W.Gemein, R. T.Schirrmeister, P. Chrabąszcz // NeuroImage Journal. – 2020. – №220(10). – С. 1–48.

13. Raschka, S. Python Machine Learning: Machine Learning and Deep Learning with Python. In Scikit-Learn and TensorFlow [Текст] / S. Raschka, V. Mirjalili // Packt Publishing Ltd. – 2017.

14. Bizopoulos, P. Signal2image modules in deep neural networks for eeg classification [Текст] / P. Bizopoulos, P. Lambrou, D. Koutsouris // Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). – 2019. – С. 702–705.

15. I. Goodfellow, Y. Courville. Deep Learning [Текст] / I. Goodfellow, Y. Bengio, A. Courville // MIT Press. – 2016. – №1(2). – С. 773.

16. Faust, O. Deep learning for healthcare applications based on physiological signals: A review [Текст] / O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, U.R. Acharya // Comput. Methods Programs Biomed. – 2018. – №161. – С. 1–13.

17. Yildirim, O. Automated detection of diabetic subject using pre-trained 2D-CNN models with frequency spectrum images extracted from heart rate signals [Текст] / O. Yildirim, M. Talo, B. Ay, U. B. Baloglu, G. Aydin, U. R. Acharya // Comput. Biol. Med. 2019. – №113. – С. 1–33.

18. Shoeibi, A. Epileptic Seizures Detection Using Deep Learning Techniques: A Review [Текст] / A. Shoeibi, M. Khodatars // Int J Environ Res Public Health. – 2021. – №18(11).

19. Choosing the right estimator [Електронний ресурс] – Режим доступу : www/ URL: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html – 11.11.2021 р. – Загол. з екрану.

20. TGAM Documentation [Електронний ресурс]. – Режим доступу : www/ URL: https://cdn.hackaday.io/files/11146476870464/TGAM Datasheet.pdf –

11.11.2021 р. – Загол. з екрану.

21. TGAM1 Spec Sheet [Електронний ресурс]. – Режим доступу : www/ URL: https://cdn.sparkfun.com/datasheets/Sensors/Biometric/tgam1.pdf – 10.11.2021 р. – Загол. з екрану.

22. Soufineyestani, M. Electroencephalography (EEG). Technology Applications and Available Devices [Текст] / M. Soufineyestani, D. Dowling, A. Khan // Applied Sciences. – 2020. – №10(21) – С. 1–23.

23. Main Features of the EEG Sensor Layer Explained [Електронний ресурс]. – Режим доступу : www/ URL: https://www.bitbrain.com/blog/eeg-sensor-layer – 10.11.2021 р. – Загол. з екрану.

24. Версии гарнитур NeuroSky [Електронний ресурс]. – Режим доступу : www/ URL: https://scriptures.ru/yoga/eeg_meditation_neurosky_headsets.htm – 09.11.2021 р. – Загол. з екрану.

25. Data transformation: standardization vs normalization [Електронний ресурс]. – Режим доступу : www/ URL: https://www.kdnuggets.com/ 2020/04/data-transformation-standardization-normalization.html – 18.11.2021 р. – Загол. з екрану.

26. Understanding the Difference Between Normalization vs. Standardization [Електронний ресурс]. – Режим доступу : www/ URL: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/ – 24.11.2021 р. – Загол. з екрану.

27. Widmann, A. Filter Effects and Filter Artifacts in the Analysis of Electrophysiological Data [Текст] / A. Widmann, E. Schröger // Frontiers in Psychology – 2012. – №3(233) – С. 233.

28. Lyons, R. Understanding Digital Signal Processing [Текст] / R. Lyons // Pearson, 3rd edition – 2004. – С. 954.

29. Pitfalls of Filtering the EEG Signal [Електронний ресурс]. – Режим доступу : www/ URL: https://sapienlabs.org/pitfalls-of-filtering-the-eeg-signal/ – 17.11.2021 р. – Загол. з екрану.

30. Doma, V. Comparative analysis of machine learning methods for emotion

recognition using EEG and peripheral physiological signals [Текст] / V. Doma, M. Pirouz // Journal of Big Data – 2020. – №7(1) – С. 1–21.

31. Support Vector Machine. Introduction to Machine Learning Algorithms. [Електронний ресурс]. – Режим доступу : www/ URL: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 – 27.11.2021 р. – Загол. з екрану.

32. ThinkGear Serial Stream Guide [Електронний ресурс]. – Режим доступу : www/ URL: http://developer.neurosky.com/docs/doku.php?id=thinkgear_communications_ protocol – 27.11.2021 р. – Загол. з екрану.

33. Hossein, M. Simplified Welch Algorithm for Spectrum Monitoring [Текст] / M. Hossein, G. Gandubert, G. Gleeton, P. Ivanov, R. Landry // Appl. Sci. – 2021. – №11(1). – С. 1–86.

34. Calculus. Integration of Functions. Simpson’s Rule [Електронний ресурс]. – Режим доступу : www/ URL: https://math24.net/simpsons-rule.html – 27.11.2021 р. – Загол. з екрану.

35. Pritha Bhandari. Understanding and calculating standard deviation [Електронний ресурс]. – Режим доступу : www/ URL: https://www.scribbr.com/statistics/standard-deviation/ – 29.11.2021 р. – Загол. з екрану.

36. K-Nearest Neighbors for Machine Learning [Електронний ресурс]. – Режим доступу : www/ URL: https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/ – 29.11.2021 р. – Загол. з екрану.

37. Cross-validation: evaluating estimator performance [Електронний ресурс]. – Режим доступу : www/ URL: https://scikit-learn.org/stable/modules/cross_validation.html – 29.11.2021 р. – Загол. з екрану.