

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

(повна назва)

Кафедра прикладної математики

(повна назва)

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Застосування методів розпізнавання зображень

для класифікації часових рядів

(тема)

Виконав:

студент 2 курсу, групи САУМ-19-1

(прізвище, ініціали)

Спеціальність 124 Системний аналіз

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системний аналіз і управління

(повна назва освітньої програми)

Керівник проф. Кіріченко Л.О.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ

(підпис)

Тевяшев А.Д.

(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 124 Системний аналіз

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системний аналіз і управління

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____

(підпис)

“ _____ ” _____ 2020 р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові Воронову Глібу Георгійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Застосування методів розпізнавання зображень для класифікації часових рядів

затверджена наказом по університету від 23 жовтня 2020 р. № 1420 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 грудня 2020 р.

3. Вихідні дані до роботи Часові медичні ряди

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Системний аналіз проблеми вибору середовища проведення дослідження часових рядів за допомогою вейвлетів

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Результати обчислювального експерименту

5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій _____

1. Актуальність теми роботи _____

2. Постановка задачі _____

3. Системний аналіз проблеми _____

4. Метод чисельного аналізу _____

5. Результати обчислювального експерименту _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	вересень 2020 р.	виконано
2	Вибір та обґрунтування методу	жовтень – листопад 2020 р.	виконано
3	Розробка алгоритму і програми	листопад – грудень 2020 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	листопад – грудень 2020 р.	виконано
5	Робота над текстом пояснювальної записки	грудень 2020 р.	виконано
6	Представлення роботи на рецензію в ЕК	грудень 2020 р.	виконано

Дата видачі завдання 1 вересня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Кіріченко Л.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 57 с., 9 табл., 13 рис., 1 дод., 10 джерела.

ЧАСОВИЙ РЯД, ВЕЙВЛЕТ, СКАЛОГРАМА, НЕЙРОННА МЕРЕЖА, МАШИННЕ НАВЧАННЯ

Об'єкт дослідження – часові ряди, що були отримані під час проведення електроенцефалографії.

Мета роботи – провести класифікацію часових рядів з допомогою методів розпізнавання зображень.

Методи дослідження – методи вейвлет-аналізу часових рядів, методи розпізнавання зображень за допомогою глибинних нейронних мереж.

Атестаційна робота присвячена дослідженню застосування методів класифікації для аналізу часових рядів за допомогою зображень вейвлет спектрів. Як класифікатор зображень застосовано глибоку нейронну мережу. Обчислювальний експеримент було проведено для часових рядів електроенцефалограми здорових людей та людей з деякими захворюваннями.

ABSTRACT

Introductory note: 57 pages, 9 tables, 13 figures, 1 appendix, 10 sources.

TIME SERIES, WAVELET, SCALOGRAM, NEURAL NETWORK, MACHINE LEARNING.

Object of research – time series obtained during electroencephalography.

Purpose of work – to classify time series using image recognition methods.

Methods of research – methods of wavelet analysis of time series, methods of image recognition by means of deep neural networks.

This thesis is devoted to the study of the application of classification methods for the analysis of time series using images of wavelet spectra. A deep neural network is used as an image classifier. A computational experiment was performed for electroencephalograms time series of healthy people and people with certain diseases.

ЗМІСТ

	С.
Вступ	8
1 Системний аналіз проблеми вибору середовища проведення дослідження часових рядів за допомогою вейвлетів та постановка задач дослідження	10
1.1 Системний аналіз проблеми вибору середовища для проведення дослідження	10
1.1.1 Вербальна модель системи	10
1.1.2 Морфологічний опис системи	11
1.1.3 Функціональна модель системи	11
1.1.4 Інформаційна модель системи	15
1.2 Аналіз сценаріїв вирішення проблеми вибору середовища для проведення дослідження	17
1.2.1 Модель аналізу проблеми	17
1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій	19
1.2.3 Модель вирішення проблеми	23
1.3 Змістовна та формальна постановка задачі	24
1.3.1 Змістовна постановка задачі	24
1.3.2 Формальна постановка задачі	24
1.4 Постановка задач дослідження	26
2 Вибір та обґрунтування методу розв’язання	27
2.1 Вейвлет-перетворення	27
2.2 Принцип вейвлет-перетворення	28
2.2.1 Свойства вейвлета	31
2.2.2 Основні функції вейвлет-перетворення	33
2.3 Глибокі нейронні мережі	34
3 Програмна реалізація	38
3.1 Python як потужний аналітичний інструмент	38

	7
3.2 Нейронні мережі для класифікації зображень	40
3.3 Опис програми.....	41
4 Результати обчислювального експерименту	43
Висновки	47
Перелік джерел посилання	48
Додаток А Listing program.....	49

ВСТУП

Одне з найскладніших завдань класифікації – класифікація часових рядів. Часовий ряд можна представити як реалізацію часу деякої динамічної системи, яка відображає безліч внутрішніх і зовнішніх зв'язків цієї системи. До складних динамічних систем належать біологічні, інформаційні, технічні та соціальні. Часові ряди таких складних систем частково або повністю випадкові, мають зворотну нелінійну залежність і тривалу кореляційну залежність. Відомим прикладом біомедичних часових рядів є реалізації електроенцефалограми (ЕЕГ).

Мозок людини має електричне поле, яке характеризується електричними стрибками від декількох міліампер до декількох сотень міліампер. Кожен нейрон мозку генерує зміну електричного потенціалу, цей процес можна виміряти. Аналіз електричного поля мозку широко використовується при діагностиці різних захворювань. Таким чином, постає завдання класифікації реалізацій ЕЕГ.

В даний час машинне навчання широко використовується в класифікації часових рядів. Найчастіше після попередньої обробки часового ряду з нього витягується набір деяких ознак, які є вхідними даними класифікатора. В якості ознак можуть бути використані значення часового ряду або міри відстані часового ряду. Популярним методом виділення ознак є перетворення часових рядів у частотну область за допомогою перетворення Фур'є або вейвлет-перетворення. Вищезазначені підходи широко використовуються для класифікації реалізацій ЕЕГ.

Одним з популярних методів аналізу часових рядів є вейвлет-аналіз, в даний час широко поширений і розроблений. Вейвлет-спектр часового ряду можна візуалізувати у вигляді геометричних структур, а динаміку часових рядів можна наочно продемонструвати.

Таким чином, візуалізація вейвлет-спектру перетворює часові ряди в зображення, які можна класифікувати за допомогою методів комп'ютерного зору. Однак такі дослідження є досить новими і досі не набули широкого поширення.

Метою даної роботи є проведення класифікації реалізації ЕЕГ на основі візуалізації вейвлетів.

1 СИСТЕМНИЙ АНАЛІЗ ПРОБЛЕМИ ВИБОРУ СЕРЕДОВИЩА ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ ЧАСОВИХ РЯДІВ ЗА ДОПОМОГУЮ ВЕЙВЛЕТІВ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Системний аналіз проблеми вибору середовища для проведення дослідження

1.1.1 Вербальна модель системи

Об'єкт аналізу – «Аналіз часових рядів за допомогою вейвлетів».

Предмет аналізу – «Вибір середовища проведення аналізу».

Точка зору: дослідник.

Ціль: вибрати найбільш оптимальне середовище для програмної реалізації побудови вейвлетів та аналізу їх властивостей.

Модель типу «чорний ящик» акцентує увагу на взаємодії системи із зовнішнім середовищем. Тут виходами системи є її цільові продукти, а входи - це вплив середовища на систему. Зміст «чорного ящика» не розкривається, тому що увага звертається тільки на межу системи. Межа, в свою чергу, підкреслює цілісність системи, відокремленість її від зовнішнього середовища і взаємодія системи і середовища.

Для опису функціонування моделі із зовнішнім середовищем на рис. 1.1 представлена модель типу «чорний ящик».

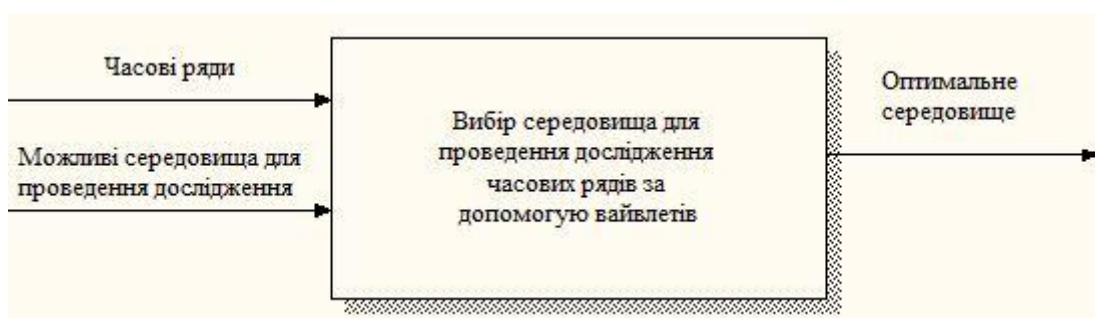


Рисунок 1.1 – Модель типу «чорний ящик»

Система, що представлена як «чорний ящик», має певний «вхід» для введення інформації і «вихід» для відображення результатів роботи. Процеси, що відбуваються в ході роботи системи, досліднику невідомі, але стан виходів функціонально залежить від стану входів.

У даній системі «входом» є часові ряди та можливі середовища для проведення їх аналізу. На «вихід» ми отримуємо інформацію про найоптимальніше середовище для проведення такого аналізу і порівняння кожного з них.

1.1.2 Морфологічний опис системи

Зовнішнє середовище – це умови і фактори, які існують в навколишньому середовищі незалежно від діяльності конкретного підприємства. Вони можуть як впливати на функціонування організації, так і відчувати на собі вплив організації.

Всі ресурси, необхідні для функціонування, організація отримує із зовнішнього середовища, з якої знаходиться в стані постійного обміну.

На рисунку 1.2 бачимо, що поставлена задача порівняльного аналізу безпосередньо виконується дослідником за допомогою ЕОМ. Причому використовуються такі інформаційні ресурси як теорія аналізу часових рядів та теорія вейвлетів. Підприємством являється центр Data Mining.

Зовнішнє середовище – сукупність всіх об'єктів за межі системи, зміна властивостей яких впливають на систему, а також тих об'єктів, чії властивості змінюються в результаті поведінки системи.

1.1.3 Функціональна модель системи

Процес моделювання будь-якої системи в IDEF0 починається з визначення контексту, тобто найбільш абстрактного рівня опису системи в цілому. В

контекст входить визначення суб'єкта моделювання, цілі і точка зору на модель.

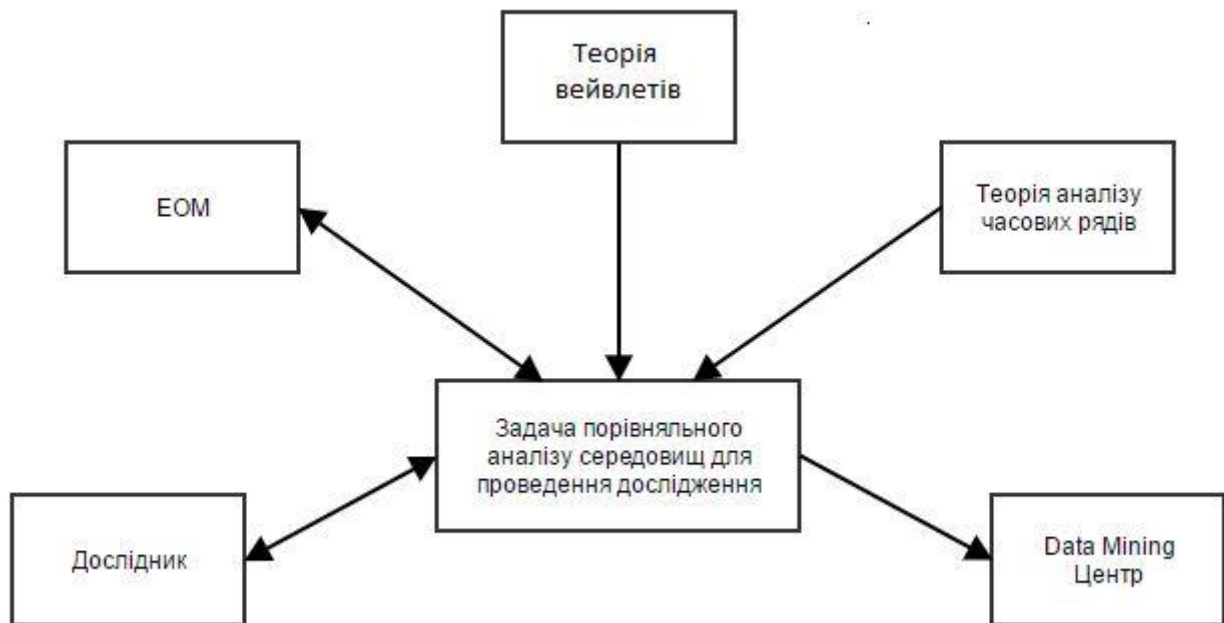


Рисунок 1.2 – Модель зовнішнього середовища системи

Для даної задачі:

- точка зору – дослідник;
- суб'єкт – проведення дослідження часових рядів за допомогою вейвлетів;
- ціль – порівняльний аналіз середовищ для проведення аналізу.

Контекстна діаграма IDEF0 зображує функціонування системи в цілому (рисунок 1.3). Основною задачею являється порівняльний середовищ для проведення дослідження, яка виконується дослідником за допомогою ЕОМ. Інформаційним ресурсом являються програмні середовища та алгоритми побудови вейвлетів, які застосовуються до вхідних даних, часових рядів. В результаті виконання роботи очікується отримання порівняльного аналізу середовищ.

Щоб розглянути функціональну частину більш детально, виконується декомпозиція системи. Декомпозиція роботи «Порівняльний аналіз середовищ для проведення аналізу часових рядів за допомогою вейвлетів» зображена на

рисунку 1.4. Даний процес розділено дві задачі:

- проведення аналізу;
- якісний аналіз результатів аналізу, в результаті якого ми маємо отримати порівняльний аналіз проходження процесів аналізу у різних середовищах.

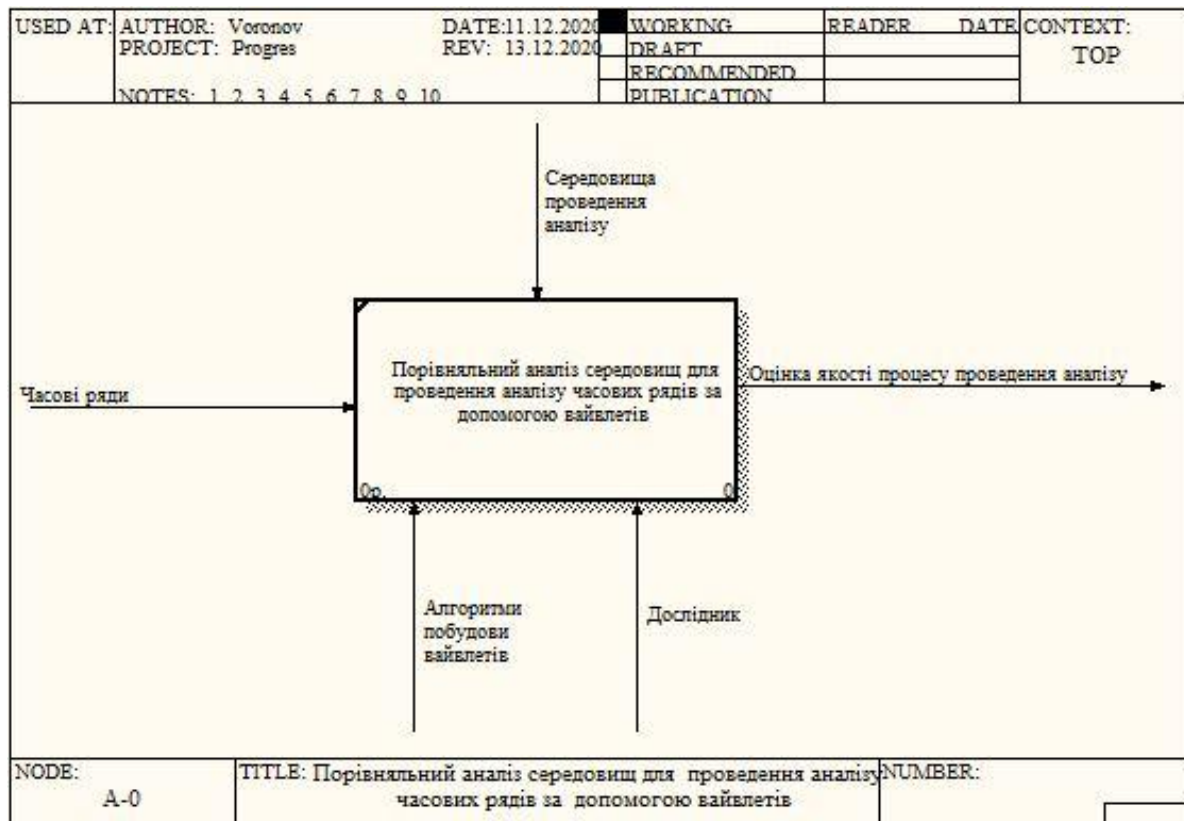


Рисунок 1.3 – Контекстна діаграма IDEF0

Далі більш детально розглянемо процес програмної реалізації алгоритмів. На рисунку 1.5 представлена декомпозиція роботи «Проведення аналізу». Основними підзадачами є:

- вивчення алгоритмів побудови вейвлетів, які можуть бути використані для аналізу часових рядів;
- вибір середовища для проведення аналізу вейвлетів часових рядів;
- розробка програми: визначення типів програмних об'єктів та, їх призначення та їх взаємозв'язок, та їх програмна реалізація;
- аналіз властивостей: деякі властивості вейвлетів можуть давати суттєву інформацію щодо властивостей вхідного часового ряду.

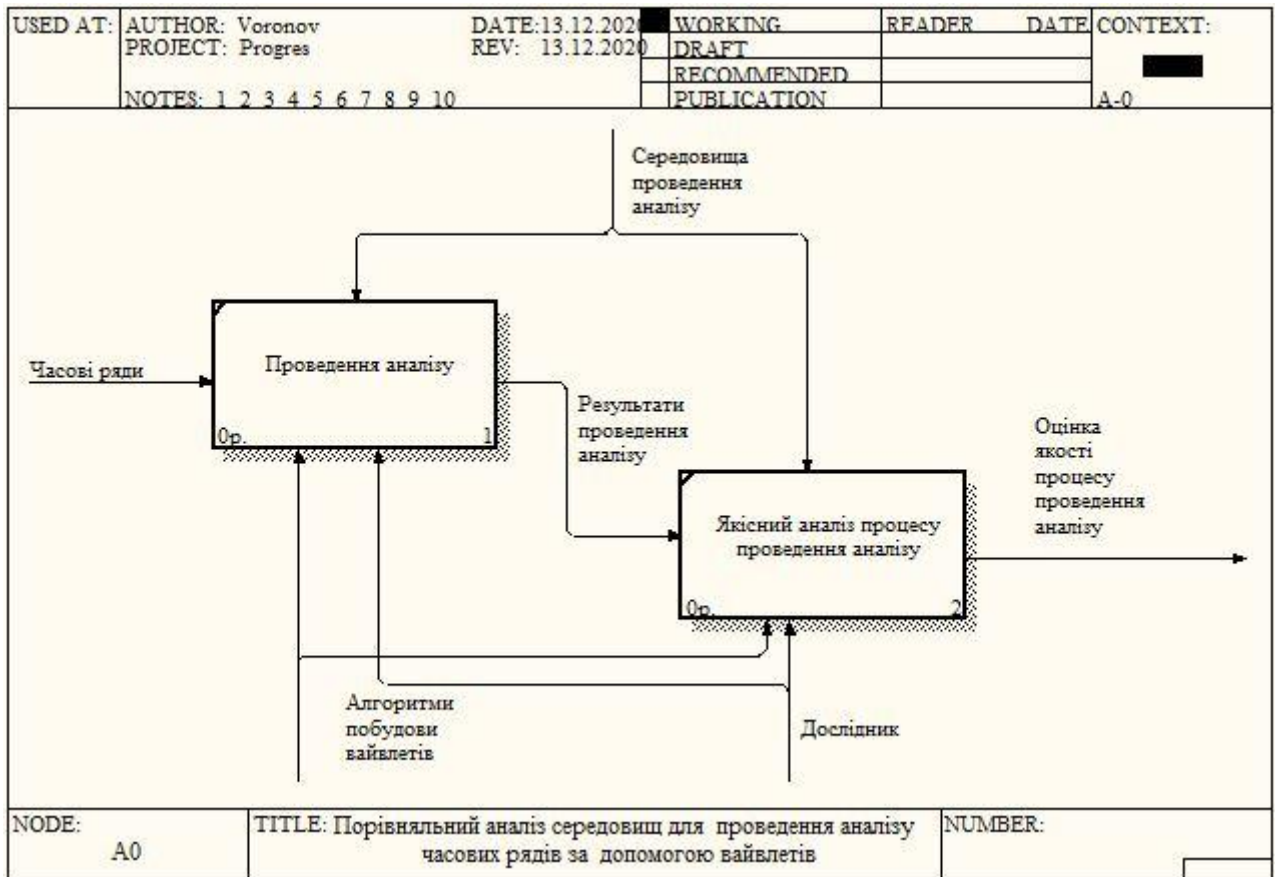


Рисунок 1.4 – Декомпозиція роботи «Порівняльний аналіз середовищ для проведення аналізу часових рядів за допомогою вейвлетів»

В результаті програмної реалізації буде отримано вейвлет та його властивості, по яким можна буде зробити висновки щодо властивостей часового ряду. На рисунку 1.6 представлена декомпозиція аналізу якості процесу проведення аналізу. Для якісного аналізу виділені такі основні підзадачі:

- вивчення методів оцінки, які можуть бути застосовані для обраних середовищ;
- застосування обраних функціоналів якості для оцінки обраних середовищ;
- аналіз переваг і недоліків програмної реалізації алгоритмів та аналізу у різних середовищах, їх застосування для достатньо великих вибірок.

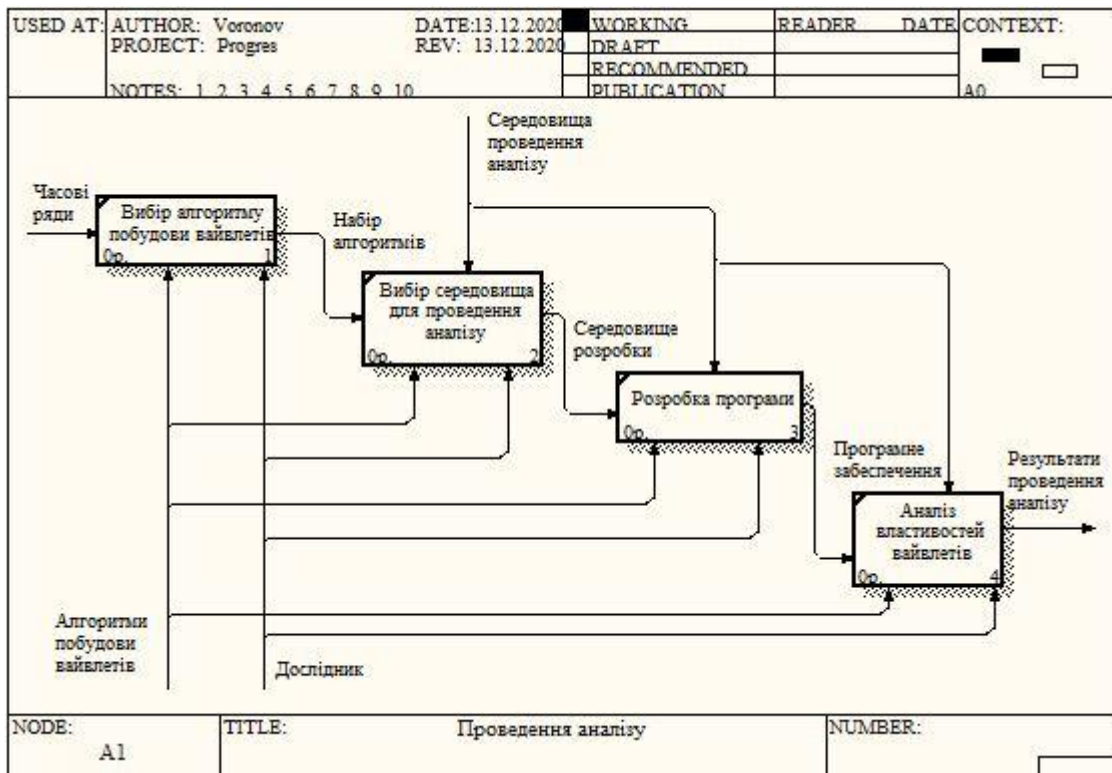


Рисунок 1.5 – Декомпозиція роботи «Програмна реалізація алгоритмів»

1.1.4 Інформаційна модель

Інформаційна модель системи представлена у вигляді DFD діаграм.

DFD-модель (діаграма) – (Data Flow Diagrams) – діаграма потоків даних – є засобом для моделювання функціональних вимог до проектованої системи. Вимоги представляють у вигляді ієрархії функціональних процесів, пов'язаних потоками даних. Основне завдання DFD-моделі (діаграми) показати перетворення вхідних даних у вихідні кожного процесу і позначити відносини між процесами.

На рисунку 1.7 представлена інформаційна модель системи, яка зображує потік даних між дослідником та підприємством (центром Data Mining). Вхідні дані, над якими будуть проводитися дослідження – часові ряди. Дослідження, які будуть проводитися – проведення аналізу часових рядів за допомогою вейвлетів у різних середовищах. На виході буде отримано порівняльний аналіз середовищ.

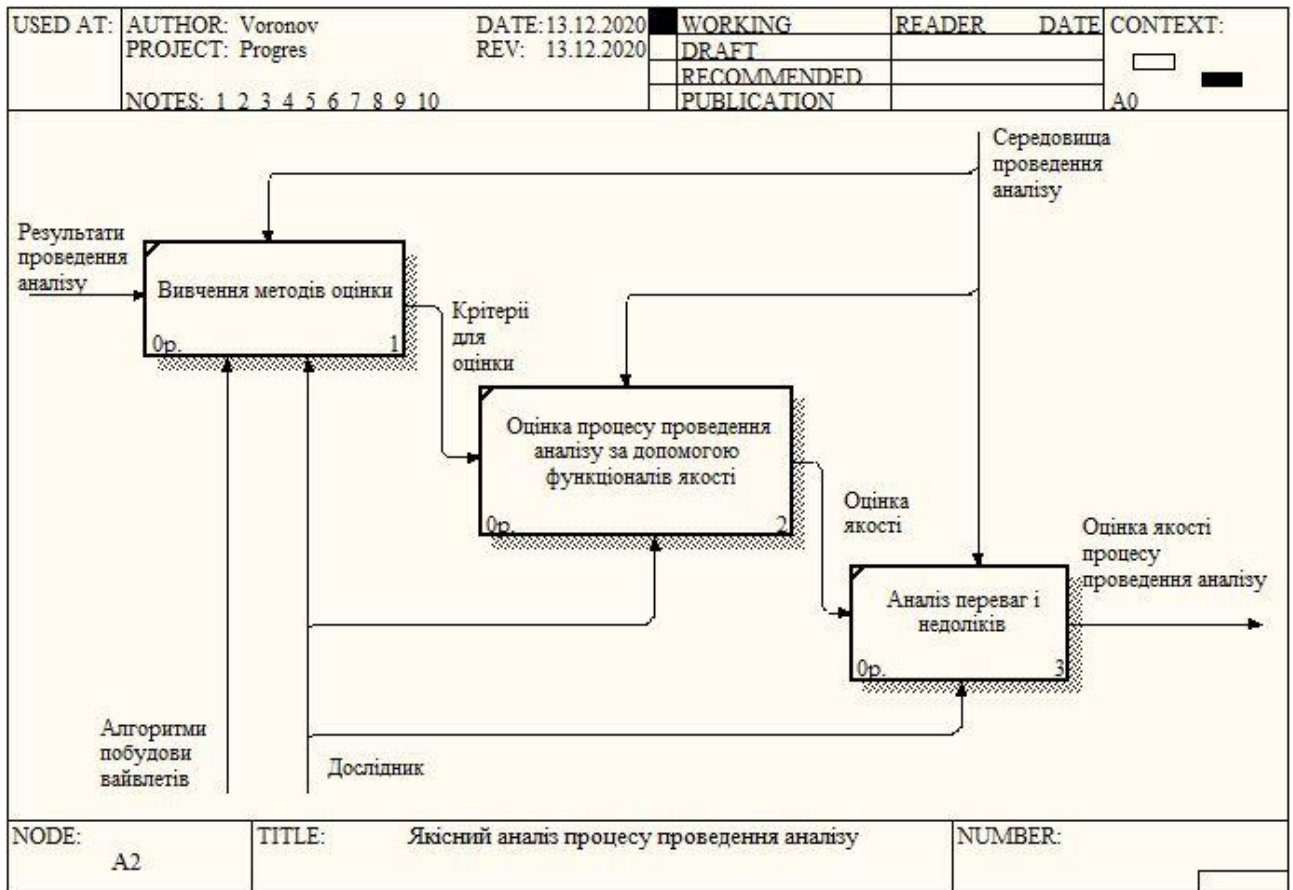


Рисунок 1.6 – Декомпозиція роботи «Якісний аналіз результатів розподілу на кластери»

Отже, процес проведення дослідження та аналізу середовищ розподілено на такі під задачі:

- вивчення алгоритмів побудови вейвлетів, в результаті якої буде вирішено які саме алгоритми мають бути реалізовані;
- вибір середовищ для розробки програмного продукту та проведення аналізу, в результаті якого буде вирішено якою мовою та з застосуванням яких бібліотек має бути виконана програмна реалізація;
- розробка програми, в результаті дослідник матиме програмний продукт;
- аналіз властивостей вейвлету, в результаті якого буде отримана інформація про властивості вхідного часового ряду;
- оцінка та порівняльний аналіз, в результаті чого буде отримано порівняльна таблиця середовищ по вказаним вище критеріям.

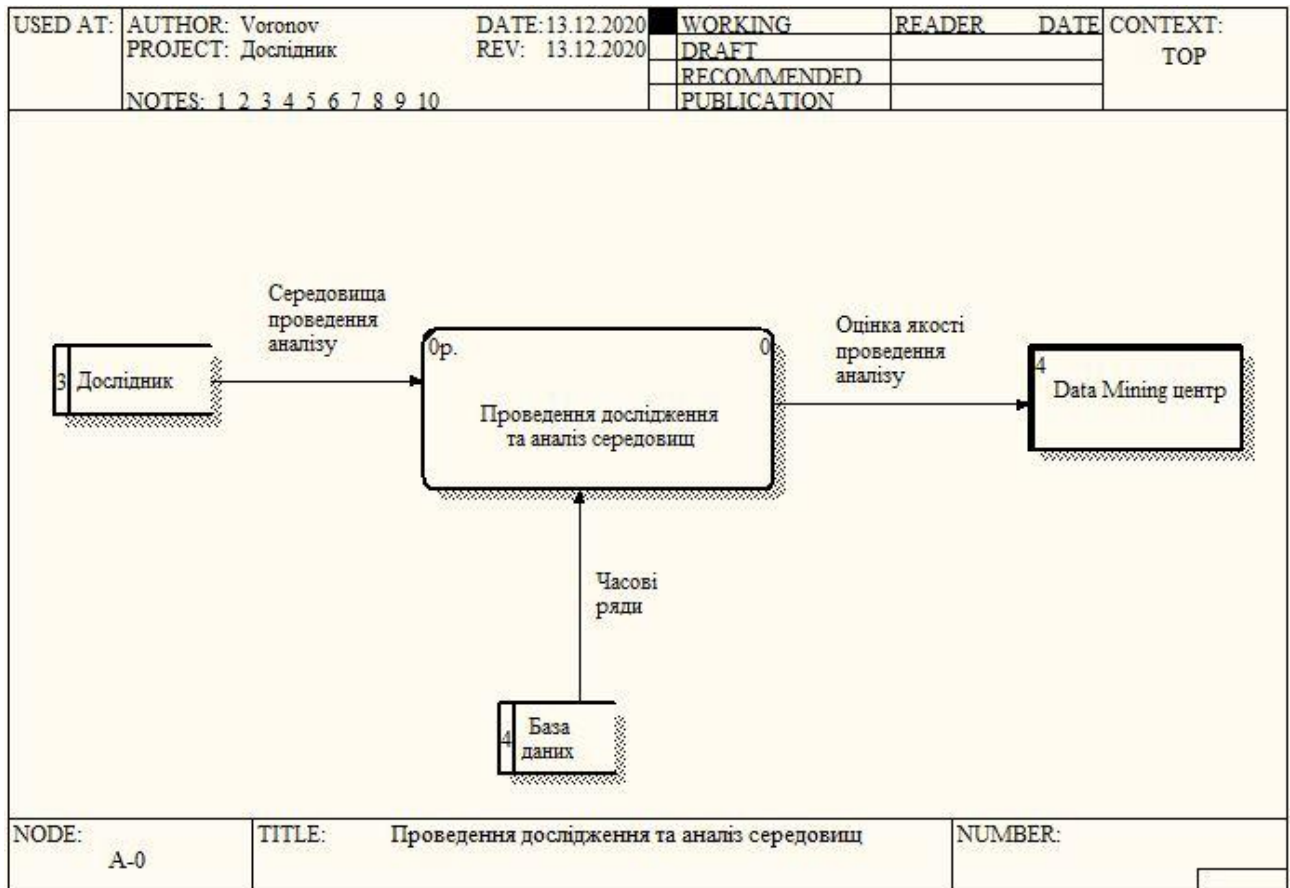


Рисунок 1.7 – DFD-діаграма

1.2 Аналіз сценаріїв вирішення проблеми вибору середовища для проведення дослідження

1.2.1 Модель аналізу проблеми

Досліджуємо проблему вибору середовища, в якому буде проводитися дослідження властивостей вейвлетів для різних часових рядів. Для цього необхідно визначити альтернативи, та критерії, за якими будемо їх відбирати. Проблема буде вирішуватися за наступними критеріями:

- критерій 1: швидкість виконання розрахунків (продуктивність);
- критерій 2: швидкість розробки програми;
- критерій 3: можливості візуалізації;
- критерій 4: наявність готових необхідних методів та бібліотек;

– критерій 5: популярність середовища (наявність добре розвиненої спільноти для вирішення проблем, що виникають під час розробки).

Обирати будемо з 3 альтернатив:

– альтернатива 1: R;

– альтернатива 2: Python;

– альтернатива 3: Wolfram Mathematica.

Відомо, що Python є доволі популярним середовищем для розв'язання задач, пов'язаних з аналізом даних та статистикою. Дане середовище має гарно розвинену спільноту, що допомагає при вирішенні нетривіальних проблем під час розробки. Слабкою стороною цього середовища являється її продуктивність, але цей недолік гарно компенсується наявністю великої кількості бібліотек, що значно розширюють стандартний функціонал. До того ж, Python помітно виділяється з поміж інших своїм зрозумілим синтаксисом, що помітно прискорює швидкість розробки на ньому.

Мову R можна охарактеризувати як середовище, що розроблялося з причілом на вирішення проблем, пов'язаних зі статистикою. З цього випливає, що сильною стороною R можна вважати гарні можливості з візуалізації даних. Показники продуктивності не дуже високі, проте це можна виправити за допомогою сторонніх бібліотек та ін. R гарно підходить далеко не для всіх задач, наявні бібліотеки не здатні покрити всі вимоги дослідника, а спільнота цього середовища, хоч і непогано розвинена, проте недостатньо для швидкого виправлення цього недоліку. Синтаксис R є доволі непростим в освоєнні та використанні, що затрудняє процес розробки.

Wolfram Mathematica – це середовище, що розроблялося для задоволення потреб наукових досліджень, пов'язаних з математичними розрахунками. До його слабких сторін також можна і віднести доволі погану продуктивність, яку важко покращити. Синтаксис краще за конкурентів підходить для математичних розрахунків, проте погано підходить для написання великих програм з чіткою структурою. Проте його сильною стороною можна вважати доволі непогані можливості візуалізації даних.

Побудуємо ієрархічну структуру, використовуючи метод парних порівнянь. Ієрархічна структура приведена на рисунку 1.8.

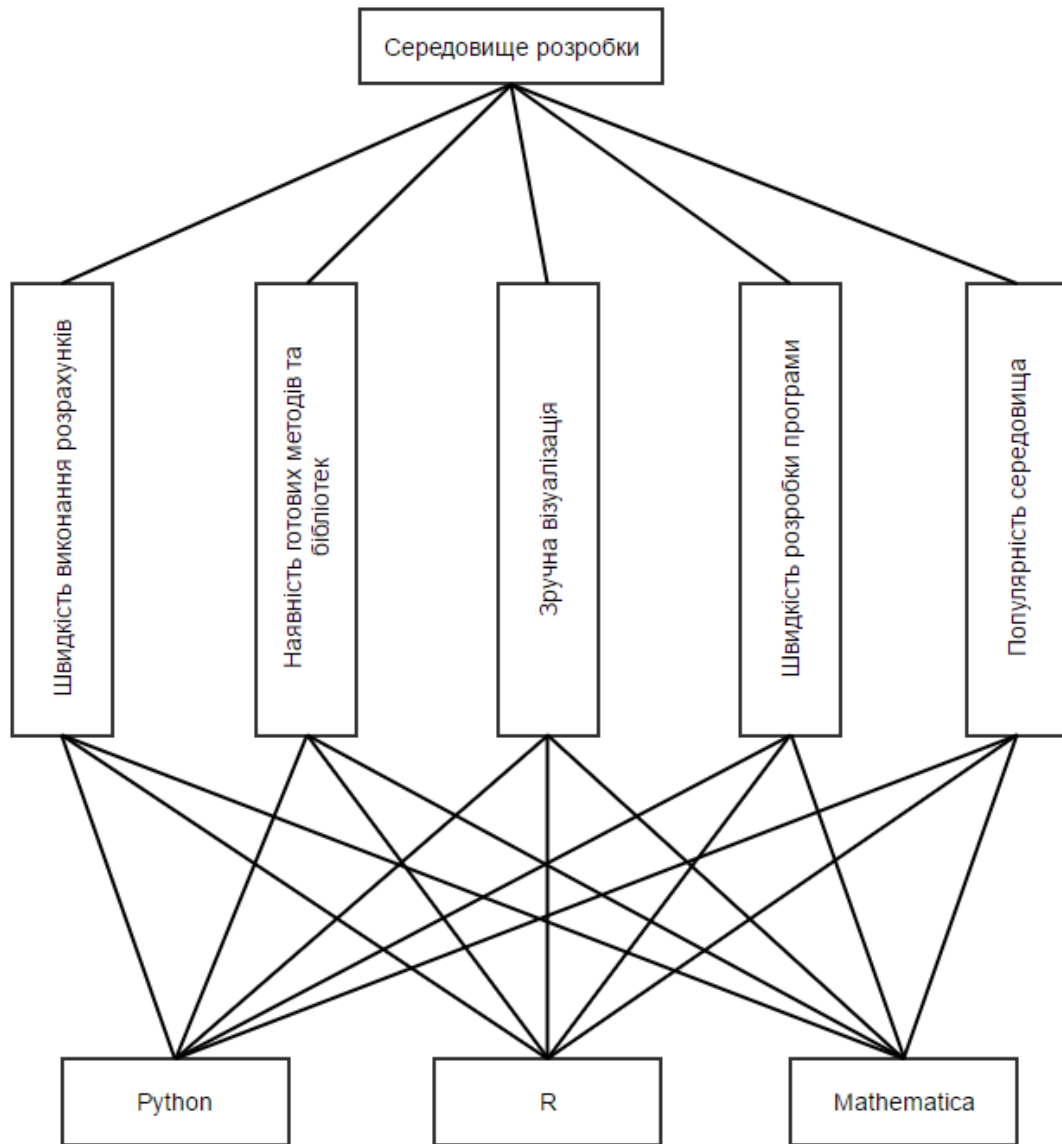


Рисунок 1.8 – Ієрархічна модель процесу аналізу рішення задачі

1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій

Для продовження аналізу необхідно побудувати матриці парних порівнянь моделі, а також критеріїв системи. Матриця попарних порівнянь наведена

у таблиці 1.1.

Таблиця 1.1 – Матриця парних порівнянь критеріїв

Критерії оцінювання	K1	K2	K3	K4	K5	Оцінки компонентів	Вектор пріоритетів	Величина значущості
K1	1	4	7	8	4	3,89	0,5	0,886
K2	$\frac{1}{4}$	1	9	8	5	2,45	0,31	1,721
K3	$\frac{1}{7}$	$\frac{1}{9}$	1	$\frac{1}{2}$	3	0,47	0,06	1,178
K4	$\frac{1}{8}$	$\frac{1}{8}$	2	1	2	0,5	0,064	1,191
K5	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{3}$	1	1	0,44	0,056	0,795
Усього						7,75		5,68

$$\text{Індекс узгодженості (ІУ)} = \frac{5,68 - 5}{5 - 1} = 0,17.$$

Випадкова узгодженість = 1,12.

$$\text{Відносна узгодженість (ВУ)} = \frac{0,17}{0,12} = 0,15 = 15\%.$$

Проведемо порівняльний аналіз альтернатив за кожним з п'яти критеріїв, результати наведено в таблицях 1.2 – 1.6. Випадкова узгодженість для матриць є рівною 0,58.

Таблиця 1.2 – Порівняння за першим критерієм

Критерій 1	A1	A2	A3	Власний вектор	Вектор пріоритетів	Величина значущості
A1	1	$\frac{1}{5}$	$\frac{1}{3}$	0,4	0,09	0,824
A2	5	1	7	3,27	0,73	0,991
A3	3	$\frac{1}{7}$	1	0,75	0,169	1,418
Всього				4,42		3,19

$$IY = 0,09; BY = 0,1 = 10\%.$$

Таблиця 1.3 – Порівняння за другим критерієм

Критерій 2	A1	A2	A3	Власний вектор	Вектор пріоритетів	Величина значущості
A1	1	$\frac{1}{3}$	2	0,87	0,27	1,073
A2	3	1	4	2,28	0,7	0,99
A3	$\frac{1}{2}$	$\frac{1}{4}$	1	0,066	0,02	0,955
Всього				3,216		3,01

$$IY = 0,005; BY = 0,008 = 0,8\%.$$

Таблиця 1.4 Порівняння за третім критерієм

Критерій 3	A1	A2	A3	Власний вектор	Вектор пріоритетів	Величина значущості
A1	1	4	$\frac{1}{6}$	0,874	0,187	1,354
A2	$\frac{1}{4}$	1	$\frac{1}{7}$	0,329	0,070	0,845
A3	6	7	1	3,476	0,743	0,973
Всього				4,679		3,171

$$IU = 0,0855; BU = 0,086 = 8,6\%.$$

Таблиця 1.5 – Порівняння за четвертим критерієм

Критерій 4	A1	A2	A3	Власний вектор	Вектор пріоритетів	Величина значимості
A1	1	$\frac{1}{3}$	4	1,01	0,280	1,189
A2	3	1	5	2,466	0,627	0,961
A3	$\frac{1}{4}$	$\frac{1}{5}$	1	0,368	0,094	0,936
Усього				4,25		3,086

$$IU = 0,043; BU = 0,074 = 7,4\%.$$

Таблиця 1.6 – Порівняння за п'ятим критерієм

Критерій 5	A1	A2	A3	Власний вектор	Вектор пріоритетів	Величина значущості
A1	1	$\frac{1}{3}$	4	1,101	0,28	1,1887
A2	3	1	5	2,466	0,627	0,9609
A3	$\frac{1}{4}$	$\frac{1}{5}$	1	0,368	0,094	0,9362
Всього				3,935		3,086

$$IU = 0,0429; BU = 0,0739 = 7,39\%.$$

1.2.3 Модель вирішення проблеми

Таблиця 1.7 – Кінцеві дані

Критерій /Альтернатива	K1	K2	K3	K4	K5	Узагальнені пріоритети
A1	0,5	0,09	0,27	0,187	0,28	0,361
A2	0,31	0,73	0,7	0,070	0,627	0,406
A3	0,06	0,169	0,02	0,743	0,094	0,233

З усіх отриманих результатів ми, як особа, що приймає рішення, можемо зробити кінцеві підрахунки та підсумувати висновки. У таблиці 1.7 наведені результати

Отримані данні дозволяють нам зробити висновок, що кращою для нас буде друга альтернатива, а саме Python, оскільки у векторі узагальнених пріоритетів їй належить найбільше значення. Перша альтернатива – R отримала

лише трохи менше значення, що свідчить про те, що загалом дане середовище гарно відходить для проведення дослідження часових рядів за допомогою побудови графів видимості.

1.3 Змістовна та формальна постановка задачі

1.3.1 Змістовна постановка задачі

Провести класифікацію часових рядів ЕЕГ за допомогою неперервного вейвлет-перетворення. Часові ряди поділені на п'ять класів, які відповідають різним станам здоров'я. Перевести часові ряди у вейвлет спектрограму, вейвлет-спектр часового ряду візуалізувати у вигляді геометричних структур та за допомогою глибинної нейронної мережі класифікувати на основі зображень.

1.3.2 Формальна постановка задачі

Часовий ряд – зібраний в різні моменти часу статистичний матеріал про значення будь-яких параметрів (в найпростішому випадку одного) досліджуваного процесу.

Вейвлет-перетворення одномерного сигналу – це його представлення у вигляді узагальненого ряду або інтеграла по системі базисних функцій

$$\Psi_{ab}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right), \quad (1.1)$$

отриманих з материнського вейвлета $\psi(t)$, що володіє певними властивостями за рахунок операцій зсуву в часі b і зміни тимчасового масштабу a . Множник

$\frac{1}{\sqrt{a}}$ забезпечує незалежність норми цих функцій від масштабуючого числа

а.

Функція $\psi: \mathbb{R} \rightarrow L^2$ називається вейвлетом, якщо задовольняє наступним умовам:

$$\begin{aligned} \psi &\in L^2, \\ \|\psi\| &= 1, \\ C_\psi &= 2\pi \int_{\mathbb{R} \setminus \{0\}} \frac{|S_\psi(w)|^2}{|w|} dw < \infty, \end{aligned} \quad (1.2)$$

де $S_\psi(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(t) \cdot e^{-iwt} dt$. Для ортонормованих вейвлетів $C_\psi = 1$.

Умова (1.2) еквівалентна наступному:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \text{ або } S_\psi(0) = 0.$$

Неперервне пряме і зворотне вейвлет-перетворення сигналу $X(t)$ має вигляд

$$\begin{aligned} W(a, b) &= \langle X(t), \psi_{ab}(t) \rangle = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} X(t) \bar{\psi}_{ab}(t) dt, \\ X(t) &= \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W(a, b) \psi_{ab}(t) \frac{dadb}{a^2}, \end{aligned}$$

де $\langle \cdot, \cdot \rangle$ операція скалярного твору.

Необхідно визначати ступінь приналежності об'єкта до кожного класу. У

роботі розглядається бінарна класифікація.

В формальному виді постановка задачі класифікації має такий вигляд. Нехай $x_i \in X$, $i = \overline{1, n}$, – множина об'єктів ознак, входів моделі, $y_i \in Y$, $i = \overline{1, n}$, – множина об'єктів відповідей, виходів моделі. Пара $(x_i, y_i) \in X \times Y$ називається розмічений об'єкт, або прецедент. Кінцева множина $\{x_i\}, i = \overline{1, n}$ представляє собою матрицю $\{x_{i,j}\}, i = \overline{1, n}, j = \overline{1, m}$, розміром $n \times m$, де рядок матриці – це масив ознак одного об'єкта, $\{y_i\}, i = \overline{1, n}$ – вектор відповідей, елемент якого є значення номеру класу. Комбінація $\{x_i\}, i = \overline{1, n}$, та $\{y_i\}, i = \overline{1, n}$, називається навчальною вибіркою. Задача класифікації полягає у визначенні функції залежності $f : X \rightarrow Y$ котра проорокує по $x \in X$ відповіді $y \in Y$.

1.4 Постановка задач дослідження

Метою дослідження є проведення класифікації часових рядів на основі розпізнавання зображень вейвлетів часових рядів за допомогою глибоких нейронних мереж.

Сформулюємо задачі для дослідження в рамках даної атестаційної роботи:

- вивчити методи вейвлет-аналізу часових рядів та побудови вейвлет-спектру часового ряду;
- ознайомитися з типами нейронних мереж для розпізнавання зображень;
- розробити програмну реалізацію для побудови вейвлет-спектру часового ряду та його зображення;
- побудувати нейронну мережу для розпізнання зображень;
- провести класифікацію часових рядів на прикладі рядів ЕЕГ;
- на основі отриманих даних зробити висновок про проведену роботу.

2 ВИБІР ТА ОБГРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Вейвлет-перетворення

Вейвлет-перетворення сигналів є узагальненням спектрального аналізу, типовий представник якого – класичне перетворення Фур'є. Термін «вейвлет» (wavelet) в перекладі з англійської означає «маленька (коротка) хвиля». Вейвлети – це узагальнена назва сімейств математичних функцій певної форми, які локальні в часі і по частоті, і в яких всі функції виходять з однієї базової (породжує) за допомогою її зрушень і розтягувань по осі часу. Вейвлет-перетворення розглядають аналізовані тимчасові функції в термінах коливань, локалізованих за часом і частоті. Як правило, вейвлет-перетворення (WT) підрозділяють на дискретне (DWT) і безперервне (CWT). DWT використовується для перетворень і кодування сигналів, CWT – для аналізу сигналів [1].

Вейвлет-аналіз являє собою особливий тип лінійного перетворення сигналів та відображаються цими сигналами фізичних даних про процеси і фізичні властивості природних середовищ і об'єктів. Базис власних функцій, за яким проводиться вейвлетного розкладання сигналів, володіє багатьма специфічними властивостями і можливостями. Вейвлетного функції базису дозволяють сконцентрувати увагу на тих чи інших локальних особливостях аналізованих процесів, які не можуть бути виявлені за допомогою традиційних перетворень Фур'є і Лапласа. До таких процесів в геофізики відносяться поля різних фізичних параметрів природних середовищ. В першу чергу, це стосується полів температури, тиску, профілів сейсмічних трас та інших фізичних величин. Принципове значення має можливість вейвлетов аналізувати нестационарні сигнали зі зміною компонентного змісту в часі або в просторі.

Вейвлети мають вигляд коротких хвильових пакетів з нульовим інтегральним значенням, локалізованих по осі аргументів (незалежних змінних), інваріантних до зсуву і лінійних до операції масштабування (стиснення-розтягування). За локалізацією в тимчасовому і частотному поданні вейвлети

займають проміжне положення між гармонійними (синусоїдальними) функціями, локалізованими по частоті, і функцією Дірака, локалізованої в часі [2].

Теорія вейвлетів не є фундаментальною фізичною теорією, але вона дає зручний і ефективний інструмент для вирішення багатьох практичних завдань. Основна область застосування вейвлетного перетворень – аналіз і обробка сигналів і функцій, нестационарних в часі або неоднорідних в просторі, коли результати аналізу повинні містити не тільки загальну частотну характеристику сигналу (розподіл енергії сигналу по частотним складовим), але і відомості про певні локальні координатах, на яких виявляють себе ті чи інші групи частотних складових, або на яких відбуваються швидкі зміни частотних складових сигналу. У порівнянні з розкладанням сигналів на ряди Фур'є, вейвлети здатні з набагато більш високою точністю представляти локальні особливості сигналів, аж до розривів 1-го роду (стрибків). На відміну від перетворень Фур'є, вейвлет-перетворення одновимірних сигналів забезпечує двовимірну розгортку, при цьому частота і координата розглядаються як незалежні змінні, що дає можливість аналізу сигналів відразу в двох просторах.

Одна з головних і особливо плідних ідей вейвлетного уявлення сигналів на різних рівнях декомпозиції (розкладання) полягає в поділі функцій наближення до сигналу на дві групи: апроксимующу – грубу, з досить повільної тимчасової динамікою змін, і деталізує – з локальної та швидкою динамікою змін на тлі плавної динаміки, з подальшим їх дробленням і деталізацією на інших рівнях декомпозиції сигналів. Це можливо як в тимчасовій, так і в частотній областях уявлення сигналів вейвлетами [3].

2.2 Принцип вейвлет-преображення

Гармонійні базисні функції перетворення Фур'є гранично локалізовані в частотній області (до імпульсних функцій Дірака при $T \rightarrow \infty$) і не локалізовані в тимчасовій (визначені в усьому часовому інтервалі від $-\infty$ до ∞). Їх проти-

лежністю є імпульсні базисні функції типу імпульсів Кронекера, які гранично локалізовані в тимчасовій області і «розмиті» по всьому частотному діапазону. Вейвлети по локалізації в цих двох поданнях можна розглядати як функції, що займають проміжне положення між гармонійними і імпульсними функціями. Вони повинні бути локалізованими як в тимчасовій, так і в частотній області уявлення. Однак при проектуванні таких функцій ми неминуче зіткнемося з принципом невизначеності, що зв'язує ефективні значення тривалості функцій і ширини їх спектру. Чим точніше ми будемо здійснювати локалізацію тимчасового положення функції, тим ширше буде ставати її спектр, і навпаки, що наочно видно на рисунку 2.1

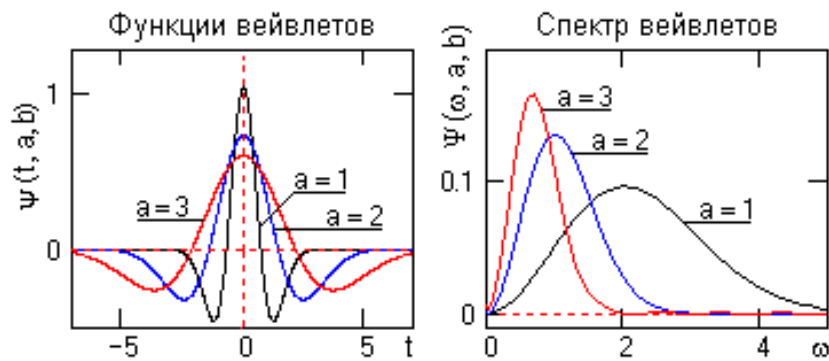


Рисунок 2.1 – Залежність ширини спектра вейвлета від його локалізації

Відмінною особливістю вейвлет-аналізу є те, що в ньому можна використовувати сімейства функцій, що реалізують різні варіанти співвідношення невизначеності. Відповідно, дослідник має можливість гнучкого вибору між ними і застосування тих вейвлетних функцій, які найбільш ефективно вирішують поставлені завдання.

Вейвлетного базис простору $L^2(\mathbb{R})$, $\mathbb{R}(-\infty, \infty)$, доцільно конструювати з фінітних функцій, що належать цьому ж простору, які повинні прагнути до нуля на нескінченності. Чим швидше ці функції прагнуть до нуля, тим зручніше використовувати їх в якості базису перетворення при аналізі реальних сигналів. Припустимо, що такою функцією є функція $\psi(t)$, що дорівнює нулю за межами деякого кінцевого інтервалу і нульове середнє значення по інтервалу завдання.

Останнє необхідно для завдання певної локалізації спектра вейвлета в частотній області. На основі цієї функції сконструюємо базис в просторі $L^2(R)$ за допомогою масштабних перетворень незалежної змінної [4].

Функція зміни частотної незалежної змінної в спектральному поданні сигналів відображається в тимчасовому поданні розтягуванням-стисненням сигналу. Для вейвлетного базису це можна виконати функцією типу $\psi(t) = \psi(a^m t)$, $a = \text{const}$, $m = 0, 1, \dots, M$, т.е. шляхом лінійної операції розтягування-стиснення, що забезпечує самоподоба функції на різних масштабах уявлення. Однак кінцівку (локальність) функції $\psi(t)$ на тимчасовій осі вимагає додаткової незалежної змінної послідовних переносів (зрушень) функції $\psi(t)$ уздовж осі (параметра локалізації), типу $\psi(t) = \psi(t + k)$, для покриття всієї числової осі простору $R(-\infty, \infty)$. С урахуванням обох умов одночасно структура базисної функції може бути прийнята наступного:

$$\psi(t) = \psi(a^m t + k). \quad (2.1)$$

Для спрощення подальших викладок значення змінних m і k приймемо цілочисельними. При приведенні функції (2.1) до одиничної нормі, отримуємо:

$$\Psi_{mk}(t) = a^{m/2} \psi(a^m t + k).$$

Якщо для сімейства функцій $\Psi_{mk}(t)$ виконується умова ортогональності

$$\langle \Psi_{nk}(t), \Psi_{lk}(t) \rangle = \int_{-\infty}^{\infty} \Psi_{nk}(t) \cdot \Psi_{lk}^*(t) dt = \delta_{nl} \delta_{km},$$

то сімейство $\Psi_{mk}(t)$ може використовуватися в якості ортонормированного базису простору $L^2(R)$.

Звідси випливає, що довільна функція цього простору може бути представлена у вигляді ряду (розкладання по базису $\psi_{mk}(t)$):

$$s(t) = \sum_{m,k=-\infty}^{\infty} S_{mk} \psi_{mk}(t),$$

де коефіцієнти уявлення сигналу – проєкції сигналу на новий ортогональний базис функцій, які визначаються скалярним твором

$$S_{mk} = \langle s(t), \psi_{mk}(t) \rangle = \int_{-\infty}^{\infty} s(t) \psi_{mk}(t) dt,$$

при цьому ряд рівномірно сходиться, тобто

$$\lim_{M,K \rightarrow \infty} \left\| s(t) - \sum_{m=-M}^M \sum_{k=-K}^K S_{mk} \psi_{mk}(t) \right\| = 0.$$

При виконанні цих умов базисна функція перетворення $\psi(t)$ називається ортогональним вейвлетом [5].

2.2.1 Свойства вейвлета

Перерахуємо властивості вейвлета, які принципово важливі для реалізації вейвлет-перетворень.

1. Локалізація. Вейвлет повинен бути безперервним, інтегрованим, мати компактний носій і бути локалізованим як в часі (в просторі), так і по частоті. Якщо вейвлет в просторі звужується, то його «середня» (домінуюча) частота підвищується, спектр вейвлета переміщається в область більш високих частот і розширюється. Цей процес повинен бути лінійним – звуження вейвлета вдвічі

повинно підвищувати його домінуючу частоту і ширину спектра також удвічі.

Вейвлетною функцією можна вважати досить добре локалізованої при виконанні умов:

$$\psi(t) \leq C / (1 + |t|)^{1+\xi}, \quad \Psi(f) \leq C / (1 + |f|)^{1+\xi},$$

причому $\xi > 0$.

2. Нульове середнє значення, т. е. виконання умови для нульового моменту:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0,$$

що забезпечує виділення локальних особливостей сигналів в межах вейвлетного носія на рівні регіональних змін і тренда, нульове посилення постійної складової сигналів з нульовим значенням частотного спектра вейвлета при $\omega = 0$, і локалізацію спектра вейвлета у вигляді смугового фільтра з центром на певній (домінуючою) частоті ω_0 вейвлетною функції. Для ігнорування регулярних поліноміальних складових сигналу і аналізу дрібномасштабних флуктуацій і особливостей високого порядку, як правило, потрібні і нульові значення певної кількості наступних моментів:

$$\int_{-\infty}^{\infty} t^m \psi(t) dt = 0.$$

Такі вейвлети називаються вейвлетами m -го порядку.

3. Обмеженість. Необхідна і достатня умова:

$$\|\Psi(t)\|^2 = \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty.$$

Оцінка хорошою обмеженості і локалізації може виконуватися з використанням виразів:

$$|\psi(t)| < 1 / (1 + |t|^n),$$

або

$$|\Psi(\omega)| < 1 / (1 + |\omega_0|^n)$$

де ω_0 – середня частота вейвлета. Число n повинно бути якомога більше.

4. Автомодельного базису або самоподоба. Форма всіх базисних вейвлетів $\psi_{ab}(t)$ повинна бути подібна материнському вейвлету $\psi(t)$, т. е. Має залишатися однією і тією ж при зрушеннях і масштабування (розтягуванні-стисненні), мати одне і те ж число осциляцій [6].

2.2.2 Основні функції вейвлет-перетворення

В основі вейвлет-перетворень, в загальному випадку, лежить використання двох безперервних, взаємозалежних і інтегруються з незалежної змінної функцій:

а) вейвлет-функції $\psi(t)$ як функції часу з нульовим значенням інтеграла і частотним фур'є-образом $\Psi(\omega)$. Цією функцією, яку зазвичай і називають вейвлетом, виділяються деталі сигналу і його локальні особливості. Як аналізують вейвлетов зазвичай вибираються функції, добре локалізовані і в тимчасовій, і в

частотній області. Приклад тимчасового і частотного способу функції наведено на рисунку 2.2

б) масштабується функції $\varphi(t)$, як тимчасової скейлінг-функції з одним значенням інтеграла, за допомогою якої виконується грубе наближення (апроксимація) сигналу.

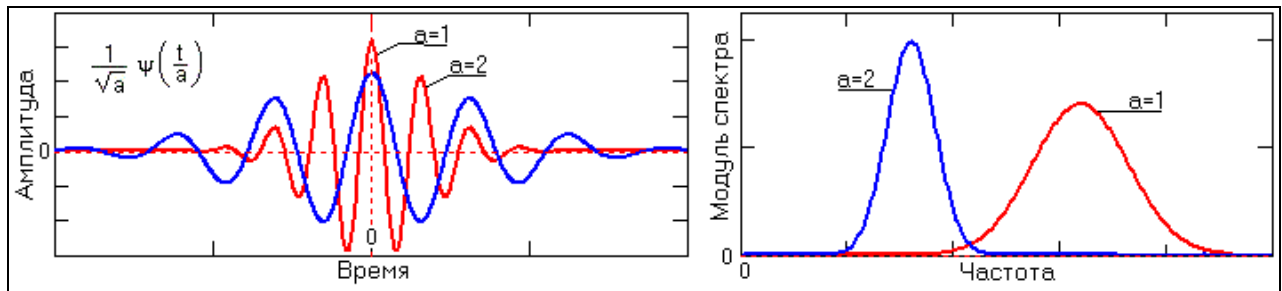


Рисунок 2.2 – Вейвлетные функции в двух масштабах

Функції $\varphi(t)$ притаманні не всім, а, як правило, тільки ортогональним вейвлет. Вони необхідні для перетворення нецентрованого і досить протяжних сигналів при роздільному аналізі низькочастотних і високочастотних складових [7, 8].

2.3 Глибокі нейронні мережі

Мережі часто реалізуються у вигляді комп'ютерних програм, хоча випускається все більша і більша кількість мікросхем, що реалізують нейронні мережі апаратним шляхом. Головна властивість мереж – здатність до навчання. Глибоке навчання – набір алгоритмів, які намагаються моделювати високорівневі абстракції у даних, використовуючи архітектури, що складаються з безлічі нелінійних трансформацій. Глибока нейронна мережа (DNN – Deep Neural Network) це штучна нейронна мережа з декількома прихованими шарами. Подібно до звичайних нейронних мереж, глибокі нейронні мережі можуть моделювати складні нелінійні відносини між елементами. У процесі навчання глибокої

нейронної мережі отримувана модель намагається представити об'єкт у вигляді комбінації простих примітивів (наприклад, у задачі розпізнавання осіб такими примітивами можуть бути частини обличчя: ніс, очі, рот і так далі). Додаткові шари дозволяють будувати абстракції все більш високих рівнів, що і дозволяє будувати моделі для розпізнавання складних об'єктів реального світу.

Як правило, глибинні мережі будуються як мережі прямого поширення. Однак останні дослідження показали, як можна застосувати техніку глибинного навчання для рекурентних нейронних мереж. Згорткові нейронні мережі використовуються в області машинного зору, де цей підхід показав себе як ефективний. Також згорткові нейронні мережі були застосовані для розпізнавання мови.

Навчання глибинних нейронних мереж може бути здійснено за допомогою звичайного алгоритму зворотного поширення помилки. Існує велика кількість модифікацій даного алгоритму. Таким чином, може бути використано кілька правил налаштування ваг. Наприклад, навчання вагових коефіцієнтів $\omega_{ij}(t)$ алгоритмом стохастичного градієнтного спуску

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \frac{\partial C}{\partial \omega_{ij}},$$

де η – стала для регулювання величини поточного кроку; C – функція втрат.

Вибір функції втрат може бути обумовлений класом завдання машинного навчання (з учителем, без учителя, з підкріпленням) і функції активації.

До двох головних проблем глибоких нейронних мереж відносять ті ж проблеми, що виникають і при навчанні звичайних нейронних мереж: час навчання та перенавчання.

Глибокі структури сильніше схильні до перенавчання, оскільки, маючи більше шарів, що дозволяють моделювати високорівневі абстракції, мережа може "вивчити" рідкісні ситуації. У цьому випадку можуть допомогти різні види регуляризації. Один з можливих методів регуляризації (dropout) припускає

випадковим чином виключені вузли під час навчання. У деяких випадках це допомагає менше запам'ятовувати рідкісні залежності в тренувальних даних.

Через простоту реалізації і хорошу збіжність для навчання глибоких нейронних мереж часто використовується метод зворотного поширення помилки і градієнтний спуск. Однак, при навчанні глибоких структур виникає кілька проблем, які особливо важливі при оптимізації функцій у просторі великої розмірності: кількість обчислювальних елементів, початкові умови для ваг мережі, а також описана вище константа регулювання величини кроку [9].

Крім того, алгоритм стохастичного градієнтного спуску відомий своєю проблемою зникаючого градієнта (*vanishing gradient*), яка полягає в ослабленні градієнта, а значить і швидкості навчання в міру поглиблення від останніх шарів мережі до початку мережі. Через це глибокі шари мережі дуже погано навчаються. Проте останнім часом є тенденція замість функції активації вузла мережі виду сигмоїда в глибоких мережах використовувати нелінійність виду ReLU (*Rectified Linear Unit*), функцію якої можна описати як $\max(0, x)$. Глибока мережа з таким видом функції активації не має проблеми ослаблення градієнта і добре навчається градієнтним спуском. За умов великих розмірностей повний перебір всіх комбінацій значень параметрів непрактичний.

Для прискорення обчислень використовується паралелізм, який закладений у саму суть алгоритму навчання нейромережі при прямому і зворотному проході. Розпаралелювання алгоритму на T потоків можливо на рівні:

- фази навчання з одночасним навчанням мережі при різних налаштуваннях її параметрів: числа шарів, нейронів у шарах, початкових установках ваг і алгоритмом управління кроком їх зміни ($T = 2 - 20$);

- пакетного навчання ($T = 10 - 1000$); в цьому випадку навчальна множина розбивається на T підмножин, для кожного обчислюється свій градієнт, отримані градієнти сумуються і, таким чином, виходить сумарний напрямок налаштування ваг;

- конвеєрного навчання шарів нейромережі ($T = 3 - 30$);

- вузлів, тобто нейронів нейромережі ($T = 100 - 1000\ 000$ і більше);

- ваг нейронів ($T = 100 - 10\,000$ і більше);
 - біт (байт) орієнтованих обчислювальних, у тому числі стохастичних потоків, з відповідною організацією основних засобів обробки, тобто суматорів, помножувачів і блоків пам'яті (T на 1 – 2 порядки більше наведених вище значень);
- Останні три рівні забезпечують найбільший коефіцієнт паралелізму і особливо ефективні при використанні апаратних засобів для прискорення навчання штучних нейронних мереж, таких як GPU і FPGA.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Python як потужний аналітичний інструмент

Python (найчастіше вживане прочитання – «Пайтон», запозичено назву з британського шоу Монті Пайтон) – інтерпретована об'єктно-орієнтована мова програмування високого рівня з строгою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання існуючих компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується декілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Серед основних її переваг можна назвати такі [10]:

- чистий синтаксис (для виділення блоків слід використовувати відступи);
- переносність програм (що властиве більшості інтерпретованих мов);
- стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);
- можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач);
- стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке зветься IDLE і яке написано на мові Python;
- зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор);
- відкритий код (можливість редагувати його іншими користувачами).

Python має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типів, а також те, що це інтерпретована мова, роблять її ідеальною для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ.

Однак Python має і деякі недоліки. Одним із них являється порівняно невисока швидкість виконання програм, що зумовлено його інтерпретованістю. Однак, даний недолік нівелюється у порівнянні з перевагами мови при написанні програм, що критичні до швидкості виконання. До того ж, даний недолік можна виправити за допомогою сторонніх бібліотек, що перекомпілюють програму, написану на Python, у програму на C. Даний підхід може значно збільшити швидкість виконання Python-програм, що вони не будуть поступатися своїм альтернативам на більш швидкісних мовах, проте дозволяє зберегти всі переваги даної мови.

Інтерпретатор мови Python і багата Стандартна бібліотека (як вихідні тексти, так і бінарні дистрибутиви для всіх основних операційних систем) можуть бути отримані з сайту, і можуть вільно розповсюджуватися. Цей самий сайт має дистрибутиви та посилання на численні модулі, програми, утиліти та додаткову документацію.

Інтерпретатор мови Python може бути розширений функціями та типами даних, розробленими на C чи C++ (або на іншій мові, яку можна викликати із C). Python також зручна як мова розширення для прикладних програм, що потребують подальшого налагодження.

Python – стабільна та поширена мова. Вона використовується в багатьох проектах та в різних якостях: як основна мова програмування або для створення розширень та інтеграції додатків. Python використовується в багатьох великих компаніях. Особливою популярністю дана мова користується серед розробників, що часто вирішують задачі, пов'язані зі статистикою чи аналізом даних. Завдяки цьому, навколо даної тематики сформувалося доволі велике суспільство, що постійно доповнює та розширює стандартний функціонал Python своїми го-

товими розробками, бібліотеками та ін. Такою кількістю доступних готових до використання інструментів для проведення аналізу не може похизуватися жодна інша мова. На Python реалізована велика кількість проєктів, також вона активно використовується для створення прототипів майбутніх програм.

3.2 Нейронні мережі для класифікації зображень

В даний час, завдяки зростанню обчислювальної потужності та появі великих баз даних із зображеннями, технології комп'ютерного зору стали дуже поширеними. Це стало можливим застосовувати ці технології не тільки в різних сферах науково-технічної діяльності, але і в повсякденному житті. Найкращі результати в області розпізнавання зображень показують глибокі нейронні мережі, зокрема згортова нейронна мережа, яка на відміну від багат шарового персептрона враховує двовимірні топології зображення.

Свертові нейронні мережі отримують функції наскрізним багат шаровим способом. Дослідження показали, що зі збільшенням кількості шарів точність розпізнавання істотно не підвищується: спочатку точність може трохи зростати, але чим більше ми збільшуємо глибину мережі, тим гірше вона починає класифікуватися. Це пов'язано з тим, що мережі навчаються методом зворотного розповсюдження помилок, і в результаті диференціації за правилом ланцюга градієнт послаблюється для досить глибоких шарів.

Для вирішення цієї проблеми була розроблена структура залишкової нейронної мережі. Головною особливістю залишкової мережі є те, що результат класифікації розширеної мережі буде не гіршим за базовий. Основна ідея полягає в тому, щоб замінити звичайне з'єднання послідовного шару ярликівим з'єднанням.

Ярличні з'єднання пропускають один або декілька шарів та збігають ідентифікатори. Їх результати додаються до виходів згрупованих шарів. Основний вузол залишкової мережі складається з двох шарів з вагами, не обов'язково кон-

волюційних, і ярличного з'єднання, яке просто передає сигнал на вихід. Структурна схема залишкової мережі представлена на рис. 3.1.

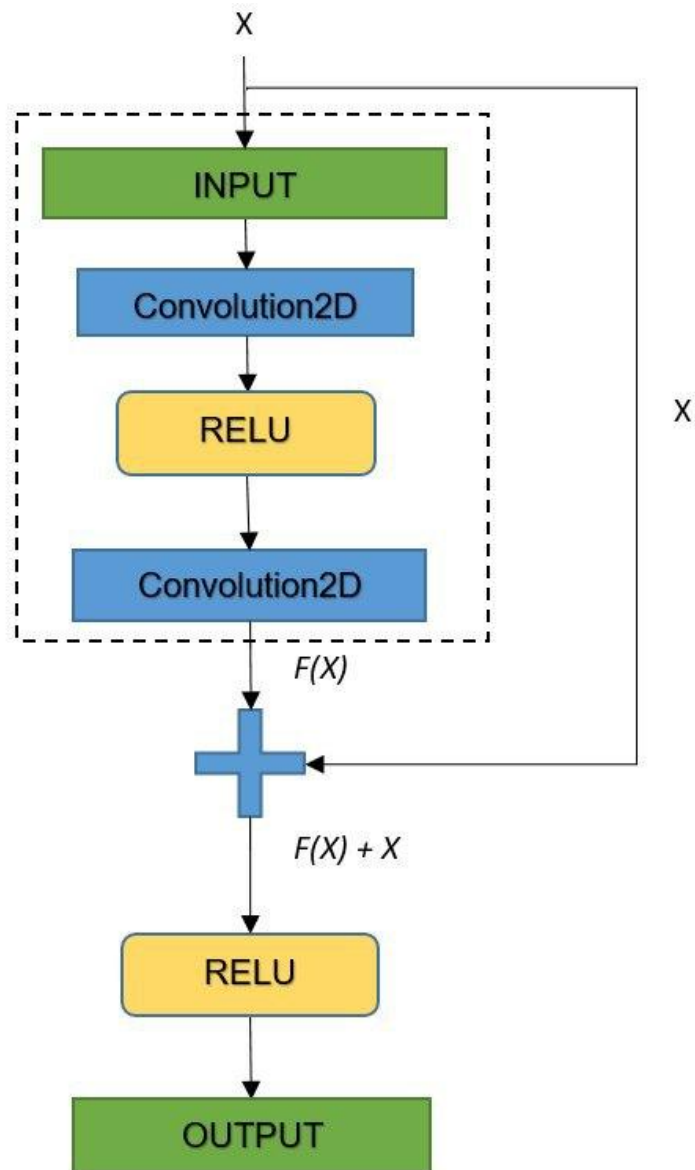


Рисунок. 3.1 Схема структури залишкової мережі

3.3 Опис програми

Програма для проведення аналізу часового ряду за допомогою графів видимості складається з трьох основних модулів main, SeriesUtils та GraphUtils. Для написання програми було використано деякі сторонні бібліотеки для роз-

ширення стандартного функціоналу, а саме: `numpy` та `matplotlib`. Код програми міститься у Додатку А.

Бібліотека `Numpy` – розширення мови `Python`, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами.

`Matplotlib` – бібліотека на мові програмування `Python` для візуалізації даних двовимірною 2D графікою (3D графіка також підтримується). Отримувані зображення можуть бути використані як ілюстрації в публікаціях.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ

Для проведення класифікації були відібрані популярні дані про записи мозкової діяльності. Початковий набір даних містить записи мозкової діяльності для різних станів людини: епілептичний напад, із зони пухлини, із здорової частини мозку, коли людина закриває очі і коли людина має відкриті очі. Таким чином, дані поділяються на 5 класів. Кожен клас містить 100 файлів, де кожен файл відповідає одному об'єкту (людині). Кожен файл містить записи мозкової активності за 23,5 секунди, що відповідає часовому ряду довжиною 4097 значень.

Хоча набір даних містить 5 класів часових рядів, більшість досліджень проводили двійкову класифікацію, де клас епілептичних нападів порівнювали з іншими. У роботі епілептичні напади діагностувались за допомогою методів машинного навчання, таких як штучні нейронні мережі, наївний байесівський, k-найближчий сусід, опорні векторні машини та k-засоби. Експеримент показав дуже добрі результати: для більшості алгоритмів точність двійкової класифікації записів тривалістю 23 секунди становила більше 99%.

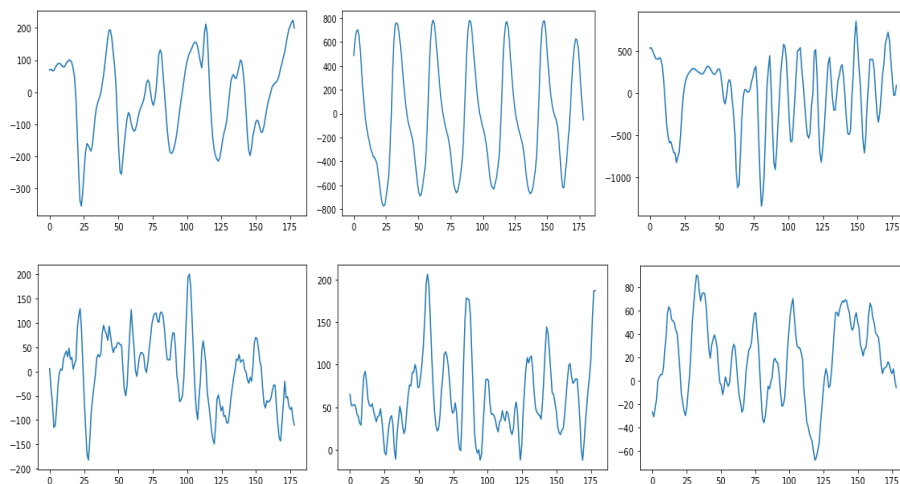


Рисунок 4.1 –ЕЕГ епілептичного нападу зверху та інших класів знизу

Більш складним завданням є визначення видалення у випадку значно коротших даних. У наборі даних кожен раз реалізація 4097 значень ділилася на 23

частини з 178 значень, що відповідає 1 секунді

У цій роботі ми також використовували згадані вище реалізації мозкової діяльності довжиною 178 значень. На рис. 4.1 показані часові реалізації ЕЕГ з 1-го класу (епілептичний напад) вгорі та реалізації з інших класів внизу.

На рис. 4.2 показані графіки рецидивів, що відповідають реалізаціям часу, показаним на рис. 4.1. Варто зазначити, що візуально знайти відмінності між класами ЕЕГ як у реалізаціях, так і в графіках рецидивів внизу майже неможливо.

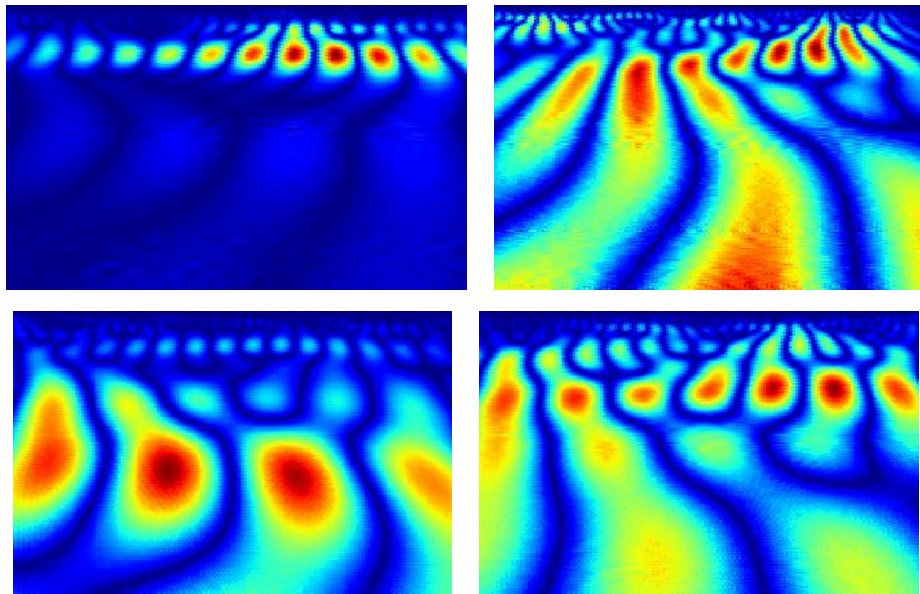


Рисунок 4.2 – Графіки рецидивів епілептичного нападу та інших класів

В експерименті використана залишкова нейронна мережа, що складається з 131 шару з вагами. Мережа складалася з 11 окремих та 120 підключених до блоків ярликового з'єднання. Перші 127 шарів призначені для виявлення характеристик (ознак), останні 4 використовуються для побудови класифікаційної функції. Глибока залишкова нейронна мережа складалася з десяти блоків, 3 згорткових шарів та одного повністю з'єданого шару. Вихід останнього повністю підключеного шару надходить на логістичну функцію, яка розподіляється на 2 класи. Нейрони в повністю зв'язаному шарі пов'язані з усіма нейронами попереднього шару. Шари суб-зразка слідує за другим і

третім згортковими шарами. Нелінійність ReLU застосовується до вихідних даних кожного згорткового і повністю пов'язаного шару. Для навчання в мережі був використаний адаптивний алгоритм оптимізації швидкості навчання Адама.

Набір даних складається з 5 класів, які містять записи мозкової активності різних станів людини. Експеримент був бінарною класифікацією реалізацій ЕЕГ, де клас епілептичних нападів порівнювали з іншими.

Кожен клас складається із 100 файлів, де кожен файл містить записи про одну людину. Кожен файл містить записи мозкової активності за 23,5 секунди. Кожен часовий ряд містить 4097 значень. Отже, набір даних містить записи мозкової активності близько 500 осіб протягом 23,5 секунд, які складаються з 4097 значень. В результаті було отримано 11 500 часових рядів із довжиною 178 значень за інтервал в 1 секунду.

Вперше експеримент був проведений на вибірці з 7500 реалізацій (6000 без нападу / 1500 з нападом), де 6000 (4800 без нападу / 1200 з нападом) були використані для навчання, і 1500 (1200 без нападу / 300 із судом) для тесту. Потім для визначення впливу обсягу вибірки на якість класифікації було використано цілий набір даних (11500 реалізацій), де для навчання було обрано 8500 реалізацій (6800 без вилучення / 1700 з вилученням) та 3000 для тесту (2400 без нападу / 600 з нападом).

Результати класифікації представлені в матриці плутанини (таблиця 4.1).

Таблиця 4.1. – Матриця похибок

	Напад (Правда)	Відсутність нападу (помилковий)
Позитивні	282(TP)	6(FP)
Негативні	18(FN)	1194(TN)

На основі матриці плутанини були розраховані показники оцінки класифікації, які представлені в таблиці 4.2.

Таблиця 4.2. – Показники оцінки класифікації

	Напад (Правда)
Точність	0.987
Precision	0.9792
Recall	0.94
F1 оцінка	0.9674

Варто зазначити, що результати класифікації, отримані в даній роботі на основі методу візуалізації сюжету рецидивів, не поступаються за точністю результатам, отриманим у інших роботах за цими самими даними для декількох різних класифікаторів.

ВИСНОВКИ

У роботі розглянуто та реалізовано метод класифікації реалізацій у часі за допомогою реалізацій ЕЕГ. Метод заснований на візуалізації рекурентних графіків та подальшій класифікації отриманих зображень. Залишкові нейронні мережі використовувались як класифікатор зображень. Була проведена бінарна класифікація реалізацій ЕЕГ, яка містила записи про епілептичний напад.

Даними для експерименту були записи мозкової активності 178 значень, що відповідає 1 секунді. Результатом бінарної класифікації було визначення наявності епілептичного нападу. Результати показали, що розглянутий метод класифікації має досить високу точність класифікації навіть при короткій тривалості реалізацій ЕЕГ.

У майбутніх дослідженнях ми маємо намір зосередитись на вивченні та тренуванні глибинних залишкових нейронних мереж, призначених для розпізнавання рецидивів графіків часових рядів з різних наборів даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Смоленцев Н. К. Основы теории вейвлетов. Вейвлеты в MATLAB. Москва : “ДМК Пресс”, 2005. 304 с.
2. Дьяконов В. П. Вейвлеты От теории к практике. Москва : “СОЛОН-Р”, 2002. 448 с.
3. Астафьева Н. М. Вейвлет-анализ: Основы теории и примеры применения // Успехи физических наук. 1996. Т. 166, № 11. С. 1145-1170.
4. Дьяконов В. П., Абраменкова И. В. Обработка сигналов и изображений: специальный справочник. Санкт-Петербург : “Питер”, 2002. 608 с.
5. Малла С. Вейвлеты в обработке сигналов. Москва : “МИР”, 2005. 671 с.
6. Переберин А. В. О систематизации вейвлет-преобразований // Вычислительные методы и программирование. 2002. Т. 2, с. 15-40.
7. Новиков Л.В. Основы вейвлет-анализа сигналов. Санкт-Петербург : “ИАНП РАН”, 1999. 152 с.
8. Теория и применение вейвлет-анализа. URL : <http://atm563.phus.msu.ru/Pyushin/index.htm>. (дата звернення 10.12.2020).
9. Хайкин. С. Нейронные сети: полный курс. 2-е изд. Москва : “Вильямс”, 2008. 1103 с.
10. Изучаем Python. Программирование игр, визуализация данных, веб-приложений. Мэттиз Эрик. URL : https://codernet.ru/books/python/izuchaem_python_programmirovanie_igr_vizualizaciya_dannyh_veb-prilozheniya/ (дата звернення 10.12.2020).