Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА МОДЕЛЮВАННЯ НЕЙРО-ФАЗЗІ МЕРЕЖІ КОХОНЕНА ДЛЯ КЛАСТЕРИЗАЦІЇ ВИКРИВЛЕНИХ ДАНИХ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-23-1
Личагіна С.М.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Шафроненко А.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Личагіній Світлані Миколаївні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та моделювання нейро-фаззі мережі Кохонена для кластеризації викривлених даних

затверджена наказом по університету від 25 листопада 2024 року № 1246Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 грудня 2024 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, перелік використаних програмних засобів: теоретичні відомості про методи кластеризації викривлених даних, мова програмування С++.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз особливостей кластеризації викривлених даних.
2. Дослідження існуючих методів кластеризації викривлених даних.
3. Моделювання нейро-фаззі мережі Кохонена для кластеризації викривлених даних.
4. Аналіз роботи методів кластеризації викривлених даних за умови однакових вхідних даних.
5. Надання оцінки отриманих результатів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми кластеризації викривлених даних, постановка задачі, схеми роботи алгоритмів.

6. Консультанти розділів роботи (п.6 включається до завдання 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

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	25.11.2024	
2	Аналіз завдання, підбір літератури	25.11.24-27.11.24	
3	Аналіз літератури з досліджуваної проблеми	27.11.24-28.11.24	
4	Аналіз технічних засобів	28.11.24-29.11.24	
5	Розробка методу	29.11.24-01.12.24	
6	Програмна реалізація	01.12.24-03.12.24	
7	Оформлення пояснювальної записки	03.12.24-05.12.24	
8	Перевірка на плагіат	14.12.2024	
9	Рецензування	16.12.2024	
10	Підготовка презентації та доповіді	19.12.2024	
11	Занесення роботи в електронний архів	02.01.2025	
12	Попередній захист кваліфікаційної роботи	02.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Шафроненко А.Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 73 с., 7 табл., 8 рис., 42 джерело.

КЛАСТЕРИЗАЦІЯ, ВИКРИВЛЕНІ ДАНІ, МЕРЕЖА КОХОНЕНА, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, АЛГОРИТМ ГУСТАФОНА-КЕССЕЛЯ, FUZZY C-MEANS.

Об’єктом дослідження є процес кластеризації даних із пропусками або викривленнями.

Метою дослідження є розробка і моделювання нейро-фаззі мережі Кохонена для кластеризації викривлених даних, оцінка її ефективності у порівнянні з традиційними методами кластеризації.

Використано методи нейронної мережі Кохонена в комбінації з фаззі логікою для підвищення точності кластеризації викривлених даних. Проведено дослідження та порівняння методів кластеризації на основі реальних датасетів з різною кількістю викривлених даних.

У результаті дослідження здійснена програмна реалізація нейро-фаззі мережі Кохонена для кластеризації викривлених даних та отримано якісні характеристики алгоритму у порівнянні з класичними алгоритмами Fuzzy C-Means та Густафсона-Кесселя.

CLUSTERING, DISTORED DATA, SELF-ORGANIZING MAP, DATA MINING, GUSTAFON-KASSEL ALGORITHM, FUZZY C-MEANS.

The object of the research is the process of clustering data with missing or distorted values.

The aim of the research is to develop and model a neuro-fuzzy Kohonen network for clustering distorted data and to evaluate its effectiveness in comparison with traditional clustering methods.

The research utilizes Kohonen neural network methods in combination with fuzzy logic to improve the accuracy of clustering distorted data. An investigation and comparison of clustering methods were conducted based on real datasets with varying levels of data distortion.

As a result of the research, a software implementation of the neuro-fuzzy Kohonen network for clustering distorted data was developed, and the qualitative characteristics of the algorithm were obtained in comparison with classical algorithms, such as Fuzzy C-Means and Gustafson-Kessel.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

SOM – Self-Organizing Maps (самоорганізовані карти Кохонена)

BMU – Best Matching Unit (блок найкращої відповідності)

MCAR – Missing Completely at Random (пропущені дані повністю випадкові)

MAR – Missing at Random (випадкові пропуски даних)

FCM – Fuzzy C-Means (с-середніх)

PC – Partition Coefficient (коефіцієнт розподілу)

CE – Classification Entropy (ентропія класифікації)

SC – Silhouette Coefficient (індекс компактності і розділення)

SI – Separation Index (індекс розділення)

XB – Xie-Beni Index (індекс Хіе-Бені)

DI – Dunn Index (індекс Данна)

GK – Gustafson-Kessel (алгоритм Густафсона-Кесселя)

ВСТУП

У сучасному світі досі існує проблема наборів даних, що описані кластеризацією векторних зображень пов’язаних з Data Mining а також у різних сферах, таких як біоінформатика, фінансовий аналіз, та обробка зображень.

Інколи виникають ситуації, коли набори даних містять відсутні значення, утворення яких втрачено. Це і є викривлені дані. У таких ситуаціях ефективним є використання математичного обчислювального інтелекту зокрема, штучних нейронних мереж, що вирішують завдання відновлення втрачених даних. Крім того, корисними є модифікації популярного методу нечітких с-середніх, що дозволяють вирішувати проблему кластеризації без повторного вилучення даних.

Існуючі методи обробки даних із викривленими значеннями ефективні, коли великі обсяги спостережень надходять у вигляді пакетів і залишаються незмінними протягом процесу обробки. Однак є чимало задач, у яких дані надходять у вигляді послідовностей в реальному часі, як це відбувається при використанні самоорганізовних карт Кохонена або їхніх модифікацій. Для вирішення проблеми кластеризації викривлених даних була створена адаптивна нейро-нечітка мережа Кохонена, яка використовує стратегію часткових відстаней (PDS FCM). Проте, якщо кількість відсутніх значень занадто велика, ця стратегія може втратити ефективність, тому, окрім нечіткої кластеризації, може бути необхідним одночасне відновлення пропущених спостережень.

Актуальність дослідження полягає у створенні ефективного методу присвяченому задачі онлайн кластеризації даних із використанням оптимальної стратегії розширення, адаптованої до умов, коли інформація обробляється в послідовному режимі, а її обсяг заздалегідь невідомий.

1 АНАЛІЗ ДАНИХ, ЗАСНОВАНИЙ НА НЕЙРОННІЙ МЕРЕЖІ

1.1 Підготовка даних

Процес підготовки даних є важливою складовою інтелектуального аналізу та грає в ньому вирішальну роль. Він повинен визначити і обробити дані, щоб зробити їх придатними для конкретних методів інтелектуального аналізу. Процес аналізу даних (data mining) може бути представлений трьома основними фазами: підготовка даних, аналіз даних, вираз і інтерпретація результатів (рис. 1.1). Власне інтелектуальний аналіз даних, заснований на нейронної мережі, складається з: підготовки даних, вилучення правил і оцінки правил (рис. 1.2).

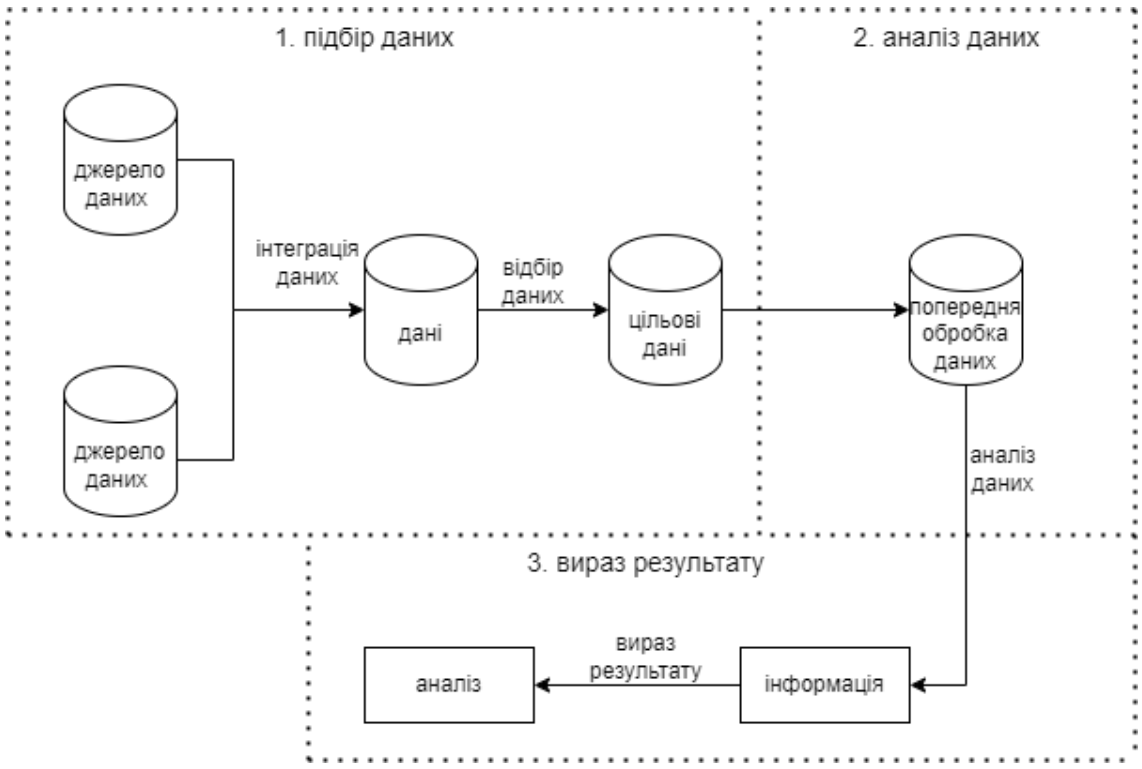

Рисунок 1.1 – Загальний процес аналізу даних

На рисунку 1.1 показано процес аналізу даних, що поділений на три основні етапи:

Етап 1. Підбір даних:

– джерело даних – початковий етап, де дані беруться з різних джерел;

– інтеграція даних – процес об’єднання даних з різних джерел для створення єдиного набору даних;

– відбір даних – вибір лише тих даних, які будуть використовуватися для аналізу, тобто цільових даних.

Етап 2. Аналіз даних:

– попередня обробка – підготовка даних, яка включає очищення, нормалізацію, трансформацію та інші необхідні кроки для покращення якості даних;

– аналіз даних – процес інтелектуального аналізу, що включає побудову моделей або використання алгоритмів для виявлення закономірностей у даних.

Етап 3. Вираз результату – перетворення отриманої інформації в зрозумілий формат, придатний для інтерпретації та використання.

Процес аналізу даних заснований на нейронній мережі показано на рисунку 1.2.

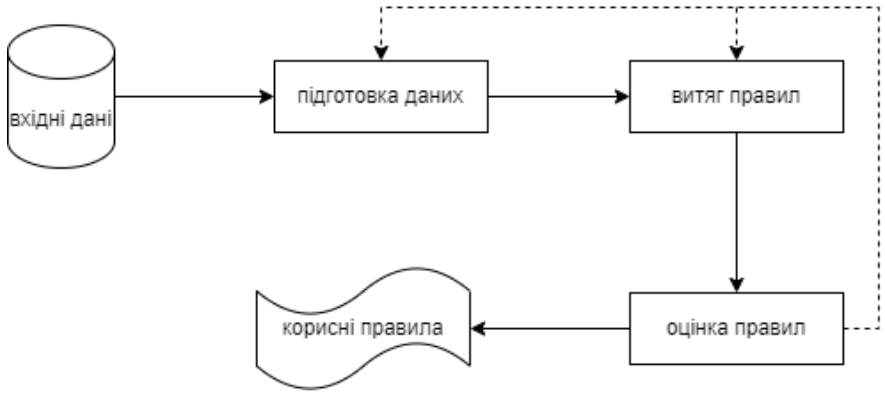

Рисунок 1.2 – Процес аналізу даних, заснований на нейронній мережі

На рисунку 1.2 зображено процес аналізу даних, а саме етапи видобування корисних правил з даних [1 – 3]. Основні етапи:

– вихідні дані – це початковий набір даних, які будуть використані для аналізу;

– підготовка даних – етап підготовки, де дані очищуються, нормалізуються та приводяться до формату, придатного для аналізу;

– витяг правил – на цьому етапі за допомогою алгоритмів інтелектуального аналізу даних (наприклад, асоціативні правила, кластеризація тощо) виділяються потенційно корисні правила або закономірності в даних;

– оцінка правил – здійснюється перевірка або оцінка корисних правил на предмет їх корисності та релевантності для конкретного завдання.

Корисні правила – після оцінки виділяються ті правила, які є дійсно корисними для застосування або подальшого використання в конкретних бізнес-задачах або процесах. Існує багато методів вилучення правил, серед яких найчастіше використовуються метод LRE (Limited Relative Error), метод «чорної скриньки», метод вилучення нечітких правил, метод вилучення правил з рекурсивної мережі, алгоритм вилучення правил з довічним входом і виходом (BIO-RE), алгоритм часткового вилучення правил (Partial-RE) та алгоритм повного вилучення правил (Full-RE).

Пунктирною лінією на рисунку 1.2 зображено можливе повернення до попередніх етапів, оскільки після аналізу або оцінки отриманих результатів може виникнути необхідність повторно переглянути початкові дані, провести додаткову обробку чи скоригувати методи витягу правил для досягнення кращого результату [4].

Аналіз даних, заснований на нейронній мережі, може працювати лише з числовими даними, з чого випливає необхідність перетворити символьні дані в числові. Найпростіший спосіб полягає в створенні таблиці відповідності між символьними даними та числовими, або інший більш складний спосіб – використання хеш-функцій для створення унікальних числових даних, що відповідають цьому рядку [5]. Незважаючи на те, що в реляційній базі даних існує багато типів даних, всі вони в основному можуть бути приведені до символьних, дискретних числових і безперервних числових даних, тобто трьох логічних типів даних.

Розглянемо приклад перетворення слова «Апельсин» в аналізі даних на основі нейронної мережі (рис. 1.3). Воно може бути перетворено в відповідні дискретні числові дані при використанні таблиці символів або хеш-функції. Вже потім дискретні числові дані можуть бути перетворені в безперервні числові дані, а також можуть бути зашифровані.

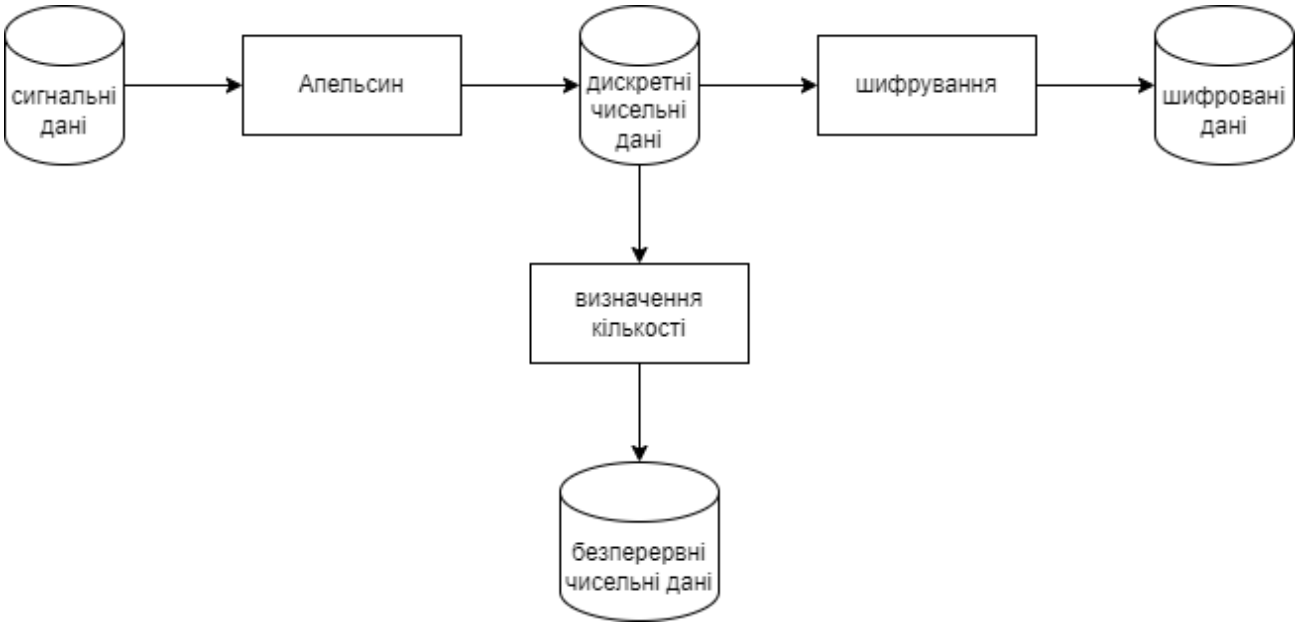

Рисунок 1.3 – Перетворення слова «Апельсин» в аналізі даних на основі нейронної мережі

1.2 Мережі Кохонена

Нейронні мережі Кохонена, або карти Кохонена (SOM) – це тип штучних нейронних мереж, які використовуються для кластеризації та візуалізації багатовимірних даних. Вони належать до класу неконтрольованого навчання і призначені для виявлення прихованих структур у даних без використання міток. Вони використовуються для вирішення завдань класифікації та похідних від них [6].

Карти Кохонена містять набір вхідних елементів, кількість яких відповідає розмірності вхідних векторів, а також набір вихідних елементів,

причому кожен з них відповідає певному кластеру (групі). Прикладом карт Кохонена може бути кластеризація клієнтів банку, наприклад, по 5 ознакам – вік, річний дохід, кількість здійснених транзакцій, середня сума транзакцій, кредитний рейтинг [6, 7]. У цьому випадку вхідним вектором буде клієнт, а координатами будуть його ознаки:

клієнт 1: [30 років, 60 000 грн, 200, 700 грн, 500];

клієнт 2: [37 років, 68 000 грн, 100, 400 грн, 360];

…

клієнт n: [50 років, 80 000 грн, 90, 1 000 грн, 900].

Навчання карт Кохонена:

– вхідні дані: багатовимірні вектори кожного клієнта подаються на вхід карти Кохонена;

– самоорганізація: під час навчання мережа самостійно формує двовимірну карту, на якій подібні клієнти групуються разом. Наприклад, клієнти з високим доходом і великим обсягом транзакцій можуть потрапити в один кластер, тоді як клієнти з низьким доходом і меншою кількістю транзакцій будуть в іншому кластері;

– вихід: кожен вихідний нейрон карти Кохонена відповідає певній групі клієнтів, наприклад:

кластер 1: молоді клієнти з низьким доходом і середньою кількістю транзакцій;

кластер 2: зрілий вік клієнтів, високий дохід, високий кредитний рейтинг;

кластер 3: клієнти з низьким кредитним рейтингом, незалежно від віку та доходу.

Приклад структури нейронної мережі Кохонена з набором двох вхідних елементів та *n* кількості вихідних елементів зображено на рисунку 1.4. Нейрони у вихідному шарі мають вагові вектори тієї ж розмірності, що й вхідні дані, у процесі навчання кожен вхідний вектор «притягує» найбільш схожий нейрон, коригуючи його ваги та ваги сусідніх нейронів.

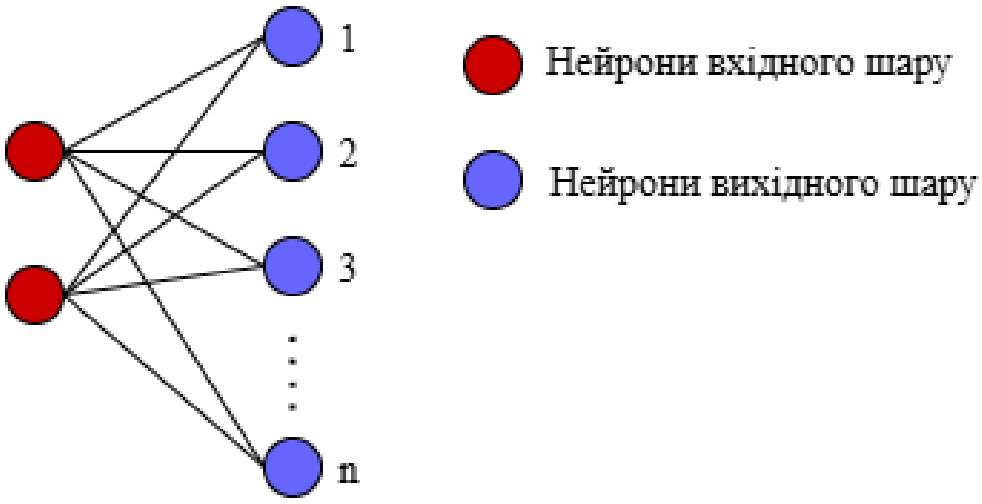

Рисунок 1.4 – Приклад мережі Кохонена

Для того, щоб використовувати нейронні мережі Кохонена в задачах класифікації, необхідно провести певну формалізацію. Кожен об’єкт, що підлягає класифікації, представлений у вигляді вектора, який подається на вхід нейронної мережі [8, 9]. Кількість вхідних нейронів відповідає числу елементів цього вектора, тобто, кількість вихідних нейронів дорівнює кількості класів: якщо є N класів, то вихідних нейронів також буде N. Кожен нейрон вихідного шару відповідає певному класу [10].

Значення на виході нейронів показує ступінь близькості вхідного вектора до певного класу, на думку мережі Кохонена. Чим більше значення нейрона, тим більше ймовірність, що об’єкт належить цьому класу. Іноді використовується активаційна функція, яка нормалізує суму всіх виходів до одиниці, що дозволяє трактувати кожен вихід як ймовірність належності об’єкта до певного класу [11].

Структура мережі Кохонена відображена на рисунку 1.5. Ця схема відображає роботу мережі в циклічному процесі. Першим кроком є отримання вхідних даних $X[1]$ та $X[2]$, далі необхідно визначити відстані між вхідним вектором і вагами нейронів, а також вибір нейрона, для якого відстань до вхідного вектора є найменшою. Після визначення такого нейрона його ваги та ваги сусідніх нейронів оновлюються, щоб краще відповідати вхідному

вектору. Даний процес повторюється для кожного вхідного вектора, до того, як мережа не навчиться правильно кластеризувати дані.

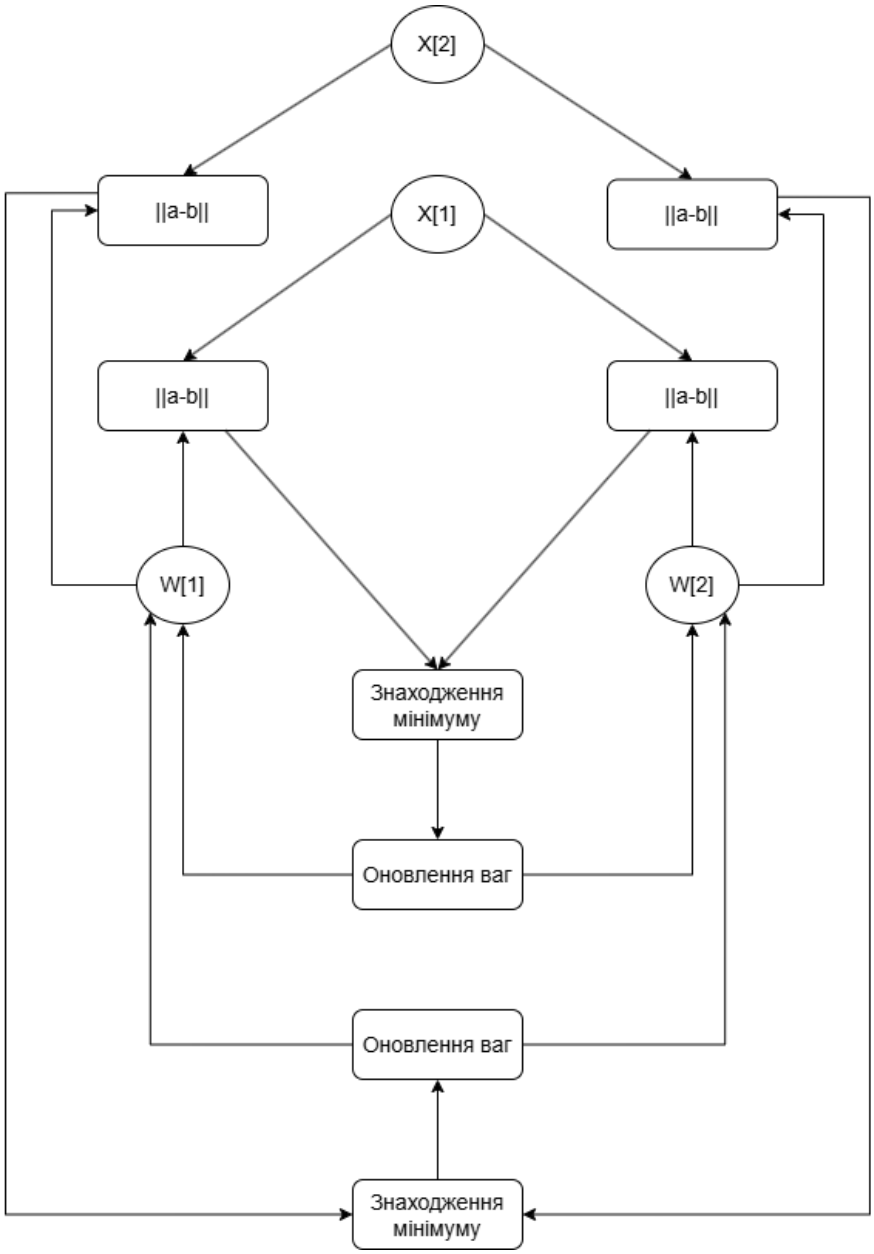

Рисунок 1.5 – Інформаційний граф алгоритму навчання або структура мережі Кохонена

1.2.1 Механізм роботи карт Кохонена

Мережа Кохонена виконує кластеризацію навчальних даних, групуючи їх за кластерами. Якщо мережа зіткнеться з набором даних, який не схожий на

жоден із відомих зразків, вона не зможе класифікувати ці дані, що дозволяє виявити їхню новизну [12].

Механізм роботи SOM можна розглянути як з концептуальної точки зору, так і з погляду реалізації алгоритму. Основні етапи алгоритму включають:

– ініціалізація. Аналітик задає кількість нейронів, після чого встановлюються початкові значення вагових векторів для кожного вузла;

– вибір вектора з вхідних даних (спостереження);

– знаходження BMU – вузла на карті з ваговим вектором, найближчим до обраного вектора вхідних даних;

– визначення кількості сусідніх вузлів BMU і навчання мережі. Під час цього процесу вагові вектори BMU та його сусідів оновлюються, наближаючись до обраного вектора даних;

– оцінка помилки карти – визначається різниця між ваговими векторами та вхідними векторами або використовується інший показник для оцінки точності карти.

Цикл повторюється доти, доки не буде досягнута або максимальна кількість ітерацій, задана аналітиком, або встановлений поріг помилки. Цей циклічний алгоритм є високоефективним, оскільки не тільки будує точну візуалізацію та проекцію багатовимірних даних на площину, а й слугує потужним інструментом для аналізу даних.

1.2.2 Завдання, які вирішуються за допомогою карт Кохонена

Самоорганізуючі карти Кохонена можуть використовуватись для рішення завдань широкого спектру, пов’язаних з кластеризацією, візуалізацією та аналізом багатовимірних даних. Однак найбільш поширене застосування SOM – рішення задач класифікації без вчителя, тобто

кластеризації [13, 14]. Основні завдання, які вирішуються за допомогою карт Кохонена, включають:

– кластеризація даних;

– візуалізація багатовимірних даних;

– аналіз аномалій;

– сегментація даних;

– компресія даних;

– розпізнавання образів та класифікація;

– аналіз часових рядків;

– аналіз текстових даних та класифікація документів.

Для завдань кластеризації даних SOM дозволяє групувати об’єкти, тобто автоматично розподілити об’єкти на кластери на основі їх характеристик, а також завданням кластеризації даних є виявлення прихованих структур, що можуть бути не виявлені при аналізі даних без кластеризації.

Для задач візуалізації багатовимірних даних SOM дозволяють перетворювати дані у двовимірні карти, що робить їх зручними для візуалізації. Це особливо корисно для даних із високою кількістю вимірів, коли важливо зрозуміти загальну структуру або кластеризацію. Також SOM можуть бути використані для зменшення розмірності багатовимірних даних, дозволяючи спростити подальший аналіз або класифікацію.

Карти Кохонена можуть допомогти виявляти аномальні дані або відхилення від нормального патерну. Це корисно в таких областях, як безпека, контроль якості або моніторинг систем, де важливо ідентифікувати нетипові поведінки або події. А також SOM можуть розпізнавати нові патерни або об’єкти, які не відповідають відомим класам чи групам, що дозволяє швидко реагувати на зміни в даних. Також, карти Кохонена широко застосовуються для сегментації клієнтів або ринків у маркетингу, розподілу медичних пацієнтів за групами на основі симптомів, а також класифікації продуктів чи інших об’єктів. Сегментація на основі SOM дозволяє отримати чітке розуміння різних груп, що допомагає в розробці цільових стратегій або

прийнятті рішень. У задачах сегментації даних SOM можуть використовуватися для стиснення даних, що зберігають ключові інформаційні ознаки, але мають зменшений обсяг. Це корисно для обробки великих наборів даних або для застосування у системах з обмеженими ресурсами.

Карти Кохонена ефективно застосовуються для розпізнавання образів, класифікації зображень або інших видів даних (наприклад, звуків або тексту) на основі подібностей. Мережа навчається розпізнавати шаблони і класифікувати об’єкти, спираючись на їхні ознаки. Використовується у системах штучного інтелекту для створення алгоритмів, здатних адаптуватися до нових даних. Також, карти Кохонена можуть бути адаптовані для роботи з часовими рядами, дозволяючи аналізувати послідовності подій або поведінку систем у часі. Це може бути корисно у фінансовому аналізі, прогнозуванні ринкових трендів, вивченні поведінки користувачів тощо.

Для аналізу текстових даних та класифікації документів SOM можуть застосовувати алгоритми обробки текстових даних на основі змісту, знаходячи схожі теми або категорії. Це корисно в задачах інформаційного пошуку, класифікації новин або групування наукових статей.

1.3 Методи відновлення втрачених даних

На даний момент розроблено багато методів відновлення пропусків у таблицях баз даних. Загалом, втрачені дані (або пропущені дані) – це відсутні значення в наборі даних, які можуть виникати з різних причин: технічні збої, людські помилки, некоректний збір даних або інші фактори. Наявність втрачених даних може ускладнити аналіз та моделювання, тому їх обробка є важливою частиною підготовки даних [15]. Можна виділити три типи втрачених даних:

– MCAR (Missing Completely at Random) – пропуски даних є повністю випадковими і не залежать ні від спостережуваних, ні від пропущених значень;

– MAR (Missing at Random) – пропуски даних можуть бути випадковими, але вони залежать від наявних, спостережуваних даних;

– MNAR (Missing Not at Random) – пропуски не є випадковими і залежать від самого пропущеного значення. Це найбільш складний для обробки тип втрачених даних.

Найбільш поширені методи обробки втрачених даних:

– виключення рядків із пропусками;

– заповнення пропусків середніми значеннями по колонках;

– метод найближчих сусідів;

– регресійний аналіз;

– метод сплайн-інтерполяції;

– метод максимальної правдоподібності та EM-алгоритм;

– алгоритми ZET і ZetBraid.

Метод виключення рядків із пропусками легко реалізувати, але його можна використовувати лише за умови виконання вимоги MCAR. Крім того, метод використовується лише при незначній кількості пропусків у таблиці, інакше отримані результати можуть бути нерепрезентативними. Основним недоліком такого підходу є втрата інформації [16].

Метод заповнення пропусків середніми значеннями по колонках має сенс лише за умови дотримання вимог MAR. Недоліком такого методу є внесення спотворень у розподіл даних та зменшення їх дисперсії.

Метод найближчих сусідів базується на пошуку рядків таблиці, які є найближчими до рядка з пропусками за певним критерієм. Пропуски заповнюються значеннями зі схожих рядків у тому ж стовпці. У цьому методі рядки з неповними даними усереднюються з використанням вагових коефіцієнтів, обернено пропорційних відстані до рядка з пропуском. Проте, за великої кількості пропусків метод малопридатний, оскільки він передбачає наявність зв’язків між рядками таблиці.

Метод регресійного аналізу ефективний за умов дотримання вимоги MAR, хоча в окремих випадках допускається застосування менш жорстких

умов. Важливо також виконувати передумови регресійного аналізу. Основним недоліком методу є те, що якість відновлення пропусків сильно залежить від правильного вибору регресійної моделі.

Успішне застосування методу сплайн-інтерполяції можливе лише за умови дотримання вимоги MAR. Основним недоліком методу є те, що при заповненні кількох підряд пропусків результат апроксимації сплайном не завжди може точно відображати реальні значення даних [17].

Метод максимальної правдоподібності та EM-алгоритм вимагає перевірки гіпотез щодо розподілу значень змінних. Застосування ускладнюється, якщо пропусків багато. Основна особливість методу полягає в побудові моделі для оцінки пропусків на основі функції правдоподібності, де параметри оцінюються за допомогою методу максимальної правдоподібності. Цей підхід дозволяє враховувати специфіку даних і застосовувати більш слабкі припущення щодо пропусків [18, 19];

Алгоритм ZET є вдосконаленою технологією верифікації експериментальних даних, що базується на гіпотезі їх надмірності. Основна ідея полягає у виборі «компетентної матриці», на основі якої визначаються параметри для прогнозування пропущених значень. Однак суб’єктивність у виборі розміру цієї матриці може призвести до врахування неінформативних і шумових факторів, що може змістити оцінки. Головна відмінність ZetBraid полягає у визначенні оптимального розміру «компетентної матриці». Хоча обидва алгоритми показали хороші результати, вони вимагають додаткової перевірки правдоподібності даних через кореляційно-регресійний аналіз і налаштування важливих параметрів.

1.4 Кластерний аналіз

Кластерний аналіз – це метод статистичної обробки даних, який використовується для групування (кластеризації) об’єктів або спостережень у

групи (кластери) на основі їхніх схожих характеристик. Основна мета кластерного аналізу полягає в тому, щоб об’єкти в межах одного кластера були максимально схожими один на одного, а об’єкти з різних кластерів – максимально відрізнялися [20].

Основні типи кластерного аналізу:

– ієрархічний кластерний аналіз;

– методи розбиття (partitional clustering);

– нечітка кластеризація (fuzzy clustering);

– щільнісні методи.

Ієрархічний кластерний аналіз поділяється на два методи, першим є агломеративний метод (знизу вгору): кожен об’єкт спочатку є окремим кластером, після чого кластери поступово об’єднуються, доки не залишиться один великий кластер. Та другий – дивізивний метод (згори вниз): навпаки, спочатку всі об’єкти належать одному кластеру, і поступово вони розділяються на менші кластери.

Серед методів розбиття найбільш поширеним прикладом є алгоритм *k*-середніх (*k*-means). У цьому методі кількість кластерів *k* задається заздалегідь, і дані розбиваються на *k* кластерів шляхом ітеративного процесу, при якому кожен об’єкт належить до найближчого центру кластера.

У нечіткій кластеризації кожен об’єкт може частково належати до кількох кластерів одночасно. Найвідоміший метод – нечіткий метод *c*-середніх (fuzzy *c*-means).

Найпоширенішим щільнісним методом є алгоритм DBSCAN, у якого кластеризація заснована на щільності точок у просторі. Він добре працює з нерегулярними структурами даних і може виявляти кластери будь-якої форми, а також ігнорувати шумові дані [21].

Кластерний аналіз має широкий спектр застосування. Його використовують у медицині, археології, хімії, біології, маркетингу, соціології, біоінформатиці, обробці зображень та інших дисциплінах. Проте універсальність застосування призвела до появи великої кількості несумісних

методів та підходів, що ускладнюють однозначне використання та інтерпретацію кластерного аналізу [22].

Основні завдання кластерного аналізу зосереджені на групуванні об’єктів таким чином, щоб максимізувати внутрішньокластерну схожість і мінімізувати міжкластерні відмінності [23, 24].

Основними завданнями кластерного аналізу є:

– групування об’єктів у класи (кластери). Метою кластерного аналізу є поділ набору даних на групи, що складаються з подібних об’єктів. Об’єкти в одному кластері мають бути максимально схожими між собою;

– визначення кількості кластерів. Вибір правильного числа кластерів важливий, щоб уникнути надмірного або недостатнього кластерування;

– візуалізація багатовимірних даних. кластерний аналіз допомагає звести багатовимірні дані до кількох груп, що дозволяє візуалізувати їх структуру в спрощеній формі;

– виявлення аномалій. Кластерний аналіз може бути використаний для ідентифікації аномалій або об’єктів, що не вписуються в жоден кластер;

– сегментація. Сегментація допомагає групувати об’єкти на основі спільних характеристик і виділяти цільові групи;

– зниження розмірності даних. Виділення кластерів допомагає визначити основні фактори, які визначають структуру даних, і скоротити набір даних до його найважливіших компонентів;

– виявлення прихованих партнерів. Завдання кластерного аналізу включає пошук закономірностей або груп, які можуть бути неочевидними при простому перегляді даних. Це дозволяє виявляти структури, що допомагають приймати рішення на основі отриманих знань;

– оцінка якості кластеризації. Це включає в себе визначення внутрішньокластерної схожості та міжкластерних відмінностей, використання показників, таких як індекс Сілуета, коефіцієнт Дейвіса-Болдіна та інші метрики;

– побудова ієрархій. У випадку ієрархічного кластерного аналізу завданням є створення дерева кластерів, яке показує, як об’єкти об’єднуються у великі групи або розбиваються на підгрупи.

Загалом завдання кластерного аналізу дозволяють розкрити структуру даних, знайти приховані зв’язки та закономірності, а також застосовувати отримані результати для прийняття обґрунтованих рішень [25].

1.5 Постановка задачі дослідження

Таким чином, кластеризація даних є актуальним завданням, тому окремі її випадки досі потребують окремих досліджень. Тому ставиться завдання розробки нейро-фаззі мережі Кохонена для кластеризації викривлених даних.

Об’єктом дослідження є процес кластеризації даних із пропусками або викривленнями.

Метою дослідження є розробка і моделювання нейро-фаззі мережі Кохонена для кластеризації викривлених даних, оцінка її ефективності у порівнянні з традиційними методами кластеризації. Використання нейронної мережі Кохонена в комбінації з фаззі логікою для підвищення точності кластеризації викривлених даних. Аналіз параметрів мережі, таких як навчальна швидкість, радіус сусідства та функції належності для покращення адаптації до викривлених даних.

Для досягнення мети необхідно вирішити такі завдання:

– провести аналіз існуючих методів кластеризації з викривленими даними;

– підготовити дані для навчання нейронної мережі;

– створити архітектуру нейронної мережі Кохонена;

– провести порівняння ефективності розробленої мережі з іншими існуючими методами кластеризації.

2 НЕЧІТКА КЛАСТЕРИЗАЦІЯ З ВІДСУТНІМИ ДАНИМИ НА ОСНОВІ ОПТИМАЛЬНОЇ СТРАТЕГІЇ ЗАВЕРШЕННЯ

2.1 Архітектура адаптивної нечіткої кластеризуючої мережі Кохонена

Архітектура нейро-фаззі мережі буде містити два шари – шар Кохонена і шар обчислення приладдя. Мережа поєднує елементи нейронних мереж та нечіткої логіки для виконання задач кластеризації або класифікації даних. Дана архітектура надає можливість ефективно обробляти складні, неоднозначні або нечіткі дані, надаючи гнучкість у кластеризації та розподілі даних між різними класами [26, 27].

Шар Кохонена – це перший шар, де відбувається процес кластеризації даних та визначення центроїдів кластерів. На вхід мережі подаються багатовимірні вектори $x_k = (x_1^k,\ x_2^k, \dots, x_n^k)^T$, де k означає номер вхідного вектора або момент часу в процесі обробки даних. Вхідний (рецепторний) шар містить набір нейронів, кожен з яких відповідає за один компонент вхідного вектора. Вектори даних подаються до нейронів рецепторного шару, і далі подаються на шар Кохонена. Нейрони шару Кохонена організовані у двовимірну решітку (сітку). Кожен нейрон має синаптичні ваги c_{ji}^k, які відповідають за вагові коефіцієнти зв’язку між вхідними нейронами та нейронами Кохонена. Ваги визначають центроїди кластерів. Набір ваг кожного нейрона можна уявити як вектор, що визначає середнє положення або центр певного кластера даних.

Конкуренція і кооперація. У шарі Кохонена реалізуються процеси конкуренції та кооперації між нейронами. Ці процеси визначають, який нейрон є найбільш «близьким» до поточного вхідного вектора (за допомогою обчислення відстаней, наприклад, за метрикою Евкліда). Нейрон, який виграє конкуренцію, коригує свої ваги, а також може частково коригувати ваги сусідніх нейронів. Це допомагає організувати нейрони в кластеризуючий шар, де кожен нейрон представляє певний кластер даних. У результаті процесу навчання нейрони Кохонена адаптують свої ваги так, щоб вони представляли

центри кластерів. Кожен нейрон Кохонена відповідає одному кластеру, і його вагові коефіцієнти представляють прототип цього кластера.

Обчислення належності – це другий шар, який використовується для визначення ступеня належності вхідного вектора до кожного з кластерів. На основі відстаней між вхідними векторами та центроїдами кластерів, які були визначені у шарі Кохонена, обчислюється ступінь належності вектора до кожного кластера. Це робиться за допомогою нечітких правил, які враховують можливість часткової належності вектора до кількох кластерів одночасно. Для кожного вектора обчислюється значення функції належності u_j^k, де j – номер кластера, а k – індекс вхідного вектора. Це значення показує, наскільки вектор x_k належить до кластера j. Функція належності зазвичай нормується, щоб сума належності до всіх кластерів дорівнювала 1. Це дозволяє інтерпретувати ці значення як ймовірність належності об’єкта до того чи іншого кластера [27].

На відміну від класичних алгоритмів кластеризації, де об’єкт може належати лише одному кластеру, у нечітких мережах об’єкт може частково належати кільком кластерам одночасно. Саме це і забезпечує гнучкість у класифікації даних, особливо коли межі між класами є нечіткими.

Використання мережі Кохонена для кластеризації викривлених даних є одним з ефективних методів для аналізу та групування складних і неоднозначних наборів даних. Класичні алгоритми кластеризації, як-от метод k-середніх, часто стикаються з труднощами при роботі з викривленими або нерівномірно розподіленими даними. Мережа Кохонена, завдяки своїй здатності до адаптації, забезпечує більш гнучкий підхід до таких даних.

У разі кластеризації викривлених даних мережа Кохонена адаптується до форми розподілу даних завдяки локальним адаптаціям ваг. Це означає, що кожен нейрон може переміщатися в напрямку локальної щільності даних, що дозволяє уникнути спотворень, характерних для класичних методів.

Формалізація навчання мережі Кохонена для кластеризації викривлених даних проходить у декілька етапів:

Етап 1. Адаптація ваг, що виконується на основі функції належності $U_q(\tau+1)$ та швидкості навчання $\eta(k+1)$:

$$w_q^{(\tau+1)}(k+1) = w_q^{(\tau)}(k) + \eta(k+1)U_q^{(\tau+1)}(k+1)^{\beta}(x_{\tau+1} - w_q^{(\tau)}(k+1). \quad (2.1)$$

Етап 2. Обчислення функції належності. Після адаптації ваг кожного нейрона, можна обчислити ступінь належності кожного вхідного вектора *x* до кожного нейрона q за допомогою нечіткої належності:

$$U_q(\tau+1) = \frac{1}{1+\left(\frac{\left\|x_{\tau+1} - w_q(\tau+1)\right\|^2}{\mu_q^{\tau}}\right)^{\frac{2}{\beta-1}}}, \quad (2.2)$$

де μ_q^{τ} – параметр масштабу для кожного кластера;

β – коефіцієнт, що визначає ступінь нечіткості.

Етап 3. Оновлення параметра масштабу. Параметр масштабу $\mu_q^{\tau+1}$для кожного нейрона розраховується з урахуванням оновлених ваг:

$$\mu_q^{(\tau+1)}(k+1) = \sum_{p=1}^{k} U_q^{(\tau+1)}(p)^{\beta} \left\| x_{\tau+1} - w_q^{(\tau)}(p) \right\|^2. \quad (2.3)$$

Таким чином, кожен вектор *x* може частково належати до кількох кластерів, що дозволяє SOM враховувати багатозначність та нечіткість кластерів у викривлених даних.

Переваги мережі Кохонена для кластеризації викривлених даних:

– адаптивність до складної структури даних;

– збереження топології та структури даних;

– гнучкість у роботі з нерівномірною щільністю кластерів;

– стійкість до шуму та аномалій;

– підтримка нечіткої кластеризації;

– візуалізація та зменшення розмірності даних;

– адаптивне навчання та точне налаштування кластерів.

Мережа Кохонена здатна самонавчатися і адаптувати свою структуру під час кластеризації, що дозволяє обробляти дані з довільною формою кластерів. На відміну від методів, які припускають сферичну форму кластерів (наприклад метод *k*-середніх), мережа Кохонена здатна адаптуватися до довільної геометрії та форми кластерів, зберігаючи природну топологію даних. Це є важливою перевагою при кластеризації викривлених або нерівномірно розподілених даних. SOM зберігає топологічні зв’язки між вхідними даними. Це означає, що схожі вхідні дані будуть відображені на сусідніх нейронах решітки. Завдяки цьому SOM може ідентифікувати та відображати природні зв’язки між кластерами, зберігаючи їхню взаємну близькість. Така здатність робить SOM надзвичайно корисною для виявлення та візуалізації структури даних, зокрема для багатовимірних наборів даних.

При кластеризації викривлених даних важливо мати можливість враховувати неоднорідну щільність кластерів. Мережа Кохонена дозволяє адаптивно розподіляти нейрони в залежності від щільності даних: нейрони в щільних областях розміщуються ближче один до одного, тоді як у розріджених областях вони можуть бути розташовані більш віддалено. Це дозволяє краще відображати структуру даних і точно ідентифікувати кластери з різною щільністю. Викривлені дані часто містять шум або аномальні значення, які можуть впливати на якість кластеризації. Завдяки процесу самоорганізації та використанню сусідніх нейронів при адаптації ваг, мережа Кохонена має високу стійкість до шуму. Шумові точки незначно впливають на ваги нейронів, оскільки SOM враховує лише сусідні нейрони при корекції. Це дозволяє SOM ефективно працювати навіть з великими шумовими наборами даних.

Мережа Кохонена дозволяє кожному вхідному вектору частково належати до кількох кластерів, що реалізується через нечітку кластеризацію. Це особливо корисно для викривлених даних, де межі між кластерами можуть

бути розмитими. Нечітка кластеризація в SOM допомагає відобразити різні рівні належності об’єктів до кількох кластерів одночасно, що забезпечує гнучкість у кластеризації складних даних. SOM ефективно зменшує розмірність даних і створює двовимірну карту, на якій кластери можна візуально ідентифікувати. Це корисно для багатовимірних викривлених даних, де важко інтерпретувати результати кластеризації без візуалізації. Завдяки SOM можна виявляти приховані патерни та візуально аналізувати структуру даних, що є важливим при роботі з великими або багатовимірними наборами даних. Також, SOM має здатність поступово уточнювати позиції нейронів, що представляють кластери, через процеси конкуренції та кооперації. Радіус сусідства та швидкість навчання зменшуються з часом, дозволяючи мережі Кохонена спочатку створювати загальні кластери, а згодом налаштовувати їх точніше для більш детальної структури.

2.2 Алгоритми на основі оптимальної стратегії завершення

Стратегія оптимального завершення – це підхід, що використовується для вирішення завдань, де метою є завершити процес з максимальною ефективністю. Оптимальне завершення може означати досягнення найкоротшого часу виконання, мінімальних витрат ресурсів, або досягнення найкращого результату в умовах обмежених ресурсів. Такі стратегії широко застосовуються в алгоритмах оптимізації, задачах планування, управлінні проектами та логістиці. В інформатиці вони знаходять застосування в задачах обробки великих даних, штучного інтелекту, логістики, управління проєктами, тощо [28].

Основні принципи стратегії оптимального завершення:

– мінімізація витрат ресурсів, що полягає в оптимізації використання часу, енергії, пам’яті чи інших обмежених ресурсів для виконання завдання;

– максимізація ефективності процесу означає досягнення найкращих результатів у межах заданих обмежень. У програмуванні це може означати оптимізацію алгоритму, щоб мінімізувати кількість виконуваних операцій або забезпечити найкращий час відгуку;

– прийняття оптимальних рішень у реальному часі, так як у багатьох сценаріях рішення повинні прийматися негайно (або за обмежений час), наприклад, у системах реального часу, де важливе значення має оперативне завершення завдання для запобігання затримкам [29].

Використання стратегії оптимального завершення:

– алгоритми планування задач у системах реального часу;

– динамічне програмування;

– жадібні алгоритми;

– метод гілок і меж;

– підкріплене навчання (Reinforcement Learning).

Алгоритми планування задач у системах реального часу націлені на те, щоб завдання завершувались точно вчасно або раніше, щоб уникнути критичних затримок. Тут застосовується стратегія, яка визначає порядок виконання завдань, щоб мінімізувати затримки або забезпечити необхідну швидкодію. У методах динамічного програмування, таких як алгоритм Беллмана-Форда, кожен етап розраховується з урахуванням оптимальних рішень для попередніх етапів, що дозволяє мінімізувати сумарні витрати або максимізувати вигоду.

Жадібні алгоритми приймають оптимальне локальне рішення на кожному кроці, що має на меті привести до глобального оптимального завершення. Наприклад, жадібні алгоритми використовуються в задачах вибору інтервалів, де потрібно вибрати максимум несуперечливих завдань із набору, або в алгоритмі побудови мінімального кістякового дерева.

Метод гілок і меж використовується для вирішення задач комбінаторної оптимізації, де необхідно знайти найкраще рішення серед безлічі можливих варіантів, як у задачі комівояжера. Цей метод дозволяє відсіювати ті варіанти,

які не можуть привести до оптимального завершення, скорочуючи кількість перевірених варіантів.

У підкріпленому навчанні агент навчається виконувати дії, які приводять до максимальної «винагороди», і застосовується стратегія оптимального завершення, яка допомагає агенту прийняти найкращі рішення для досягнення заданої мети в найкоротші терміни.

Переваги стратегії оптимального завершення:

– підвищена ефективність, тобто оптимальні стратегії дозволяють досягати результатів швидше, використовуючи менше ресурсів, що є важливим для ресурсомістких і критичних систем;

– гнучкість та адаптивність, тому стратегії оптимального завершення можуть адаптуватися до змінних умов середовища і дозволяють приймати ефективні рішення в реальному часі;

– можливість зменшення витрат, що мінімізує потребу в ресурсах і зменшує час простою або виконання, тому дозволяє скоротити фінансові витрати.

Отже, алгоритми на основі оптимальної стратегії завершення є важливими інструментами в інформатиці для вирішення задач, пов’язаних із оптимізацією та управлінням ресурсами. Вони використовуються в різних галузях – від обробки даних до штучного інтелекту, від систем реального часу до хмарних обчислень [31].

2.3 Імовірнісно-адаптивна нечітка кластеризація з відсутніми даними на основі оптимальної стратегії завершення

Базова інформація для вирішення завдань кластеризації в пакетному режимі представлена зразком спостережень, сформованих з N n-мірних векторів ознак $X = \{x_1, x_2, ..., x_N\} \subset R^n$, $x_k \in X$, $k = 1, 2, ..., N$. Результатом кластеризації є розбиття вихідного набору даних на m класів ($1 < m < N$) з

деяким рівнем $U_q(k)$ для k-го вектора ознак до q-го кластеру ($1 \le q \le m$). Вхідні дані раніше центровані та стандартизовані за всіма ознаками, отже усі спостереження належать до гіперкуба $[-1,1]^n$. Отже, дані для кластеризації форми масиву $\tilde{X} = \{\tilde{x}_1,...,\tilde{x}_k,...,\tilde{x}_N\} \subset R^n$, $\tilde{x}_k = (\tilde{x}_{k1},...,\tilde{x}_{ki},...,\tilde{x}_{kn})^T$, тобто всі спостереження $\tilde{x}_{ki}$ доступні для обробки.

Впроваджуючи об'єктивну функцію кластеризації (2.3) з обмеженнями (2.4) та розв'язуючи стандартні задачі нелінійного програмування, отримуємо ймовірнісний нечіткий алгоритм кластеризації

$$E(U_q(k), w_q) = \sum_{k=1}^{N}\sum_{q=1}^{m} U_q^{\beta}(k) D^2(\tilde{x}_k, w_q), \tag{2.4}$$

$$\sum_{q=1}^{m} U_q(k) = 1,\ 0 < \sum_{k=1}^{N} U_q(k) < N, \tag{2.5}$$

Отже, в результаті отримано ймовірнісний нечіткий алгоритм кластеризації:

$$\begin{cases} U_q^{(\tau+1)}(k) = \dfrac{(D^2(\tilde{x}_k, w_q^{(\tau)}))^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m}(D^2(\tilde{x}_k, w_l^{(\tau)}))^{\frac{1}{1-\beta}}}, \\ w_q^{(\tau+1)} = \dfrac{\sum_{k=1}^{N}(U_q^{(\tau+1)}(k))^{\beta}\tilde{x}_k}{\sum_{k=1}^{N}(U_q^{(\tau+1)}(k))^{\beta}}, \end{cases} \tag{2.6}$$

де w_q – прототип (центроїд) q-го кластера;

$\beta > 1$ – нечіткий параметр, який визначає невизначеність меж між класами;

$D^2(\tilde{x}_k, w_q)$ – відстань між $\tilde{x}_k$ і w_q в прийнятій матриці;

τ – індекс епохи обробки інформації, яка організована як послідовність $w_q^{(0)} \to U_q^{(1)} \to w_q^{(1)} \to U_q^{(2)} \to \cdots$.

Процес обчислення триває доти, доки не буде задоволено умову (2.7) або до зазначеної максимальної кількості епох Q ($\tau = 0,1,2,\ldots,Q$).

$$\left\| w_q^{(\tau+1)} - w_q^{(\tau)} \right\| \le \varepsilon \ \forall \ 1 \le q \le m, \tag{2.7}$$

де ε – визначає поріг точності.

При $\beta = 2$ і $D^2(\tilde{x}_k, w_q) = \left\| \tilde{x}_k - w_q \right\|^2$ буде отримано відомий алгоритм нечіткого *c*-засобу Бездека. Власне процес нечіткої кластеризації може бути організований в режимі онлайн як послідовна обробка. У цьому випадку алгоритм пакетної обробки може бути переписаний у рекурентній формі:

$$\begin{cases} U_q(k+1) = \dfrac{(D^2(\tilde{x}_{k+1}, w_q(k)))^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m} (D^2(\tilde{x}_{k+1}, w_l(k)))^{\frac{1}{1-\beta}}}, \\ w_q(k+1) = w_q(k) + \eta(k+1) U_q^{\beta}(k+1)(\tilde{x}_{k+1} - w_q(k)), \end{cases} \tag{2.8}$$

де $\eta(k+1)$ – параметр швидкості навчання;

$U_q^{\beta}(k+1)$ – дзвоноподібна функція сусідства нейро-нечіткої мережі Кохонена, розроблена для вирішення задач нечіткої кластеризації заснована на принципі переможець бере більше (Winner Takes More) [26].

За наявності невідомої кількості відсутніх значень у векторних зображеннях $\tilde{x}_k$, що утворюють масив $\tilde{X}$, введемо підмасиви для подальшої обробки:

$$\begin{aligned} X_F &= \{\tilde{x}_k \in \tilde{X} \mid \tilde{x}_k \text{- } vector\ containing\ all\ components\} \\ X_P &= \{\tilde{x}_{ki}, 1 \le i \le n, 1 \le k \le N \mid \ values\ \tilde{x}_k,\ available\ in\ \tilde{X}\} \\ X_G &= \{\tilde{x}_{ki} = ?, 1 \le i \le n, 1 \le k \le N \mid \ values\ \tilde{x}_k, absent\ in\ \tilde{X}\} \end{aligned} \tag{2.9}$$

Елементи підмасиву X_G позначені як «*?*» є відсутніми значеннями. Оптимальна стратегія завершення полягає в тому, що елементи підмасиву X_G розглядаються як додаткові змінні, які оцінюються шляхом мінімізації цільової функції E. Таким чином, паралельно з кластеризацією (оптимізацією E шляхом $U_q(k)$ і w_q) проводиться оцінка відсутніх спостережень (оптимізація

E за допомогою $\tilde{x}_{ki} \in X_G$). У цьому випадку алгоритм нечітких *c*-засобів, що базується на оптимальній стратегії завершення, може бути представлений у вигляді такої послідовності етапів:

Етап 1. Встановлення початкових умов для алгоритму: $\beta > 0$; $1 < m < N$; $\varepsilon > 0$; $w_q^{(0)}$; $1 \le q \le m$; $\tau = 0,1,2,...,Q$; $X_G^{(0)} = \{-1 \le \hat{x}_{ki}^{(0)} \le 1\}$,

де $X_G^{(0)} - N_G (1 \le N_G \le (n-1)N)$ – довільні початкові оцінки;

$\hat{x}_{ki}^{(0)}$ – відсутніх значень $\tilde{x}_{ki} \in X_G$.

Етап 2. Розрахунок рівнів членства шляхом вирішення задачі оптимізації:

$$U_q^{(\tau+1)}(k) = \underset{U_q(k)}{argmin} E\left(U_q(k), w_q^{(\tau)}, X_G^{(\tau)}\right) = \frac{(D^2(\hat{x}_k^{(\tau)}, w_q^{(\tau)}))^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m}(D^2(\hat{x}_k^{(\tau)}, w_l^{(\tau)}))^{\frac{1}{1-\beta}}} =$$

$$= \frac{(\left\|\hat{x}_k^{(\tau)} - w_q^{(\tau)}\right\|^2)^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m}(\left\|\hat{x}_k^{(\tau)} - w_l^{(\tau)}\right\|^2)^{\frac{1}{1-\beta}}}. \qquad (2.10)$$

Етап 3. Розрахунок центроїдів кластерів:

$$w_q^{(\tau+1)} = \arg\min_{w_q} E(U_q^{(\tau+1)}(k), w_q, X_G^{(\tau)}) = \frac{\sum_{k=1}^{N}(U_q^{(\tau+1)}(k))^{\beta} \hat{x}_k^{(\tau)}}{\sum_{k=1}^{N}(U_q^{(\tau+1)}(k))^{\beta}}. \qquad (2.11)$$

Етап 4. Перевірка умов зупинки: якщо $\left\|w_q^{(\tau+1)} - w_q^{(\tau)}\right\| < \varepsilon \ \forall \ 1 \le q \le m$ або $\tau = Q$, тоді алгоритм закінчується, інакше перехід на Етап 5.

Етап 5. Оцінка відсутніх спостережень шляхом вирішення задачі оптимізації:

$$X_G^{(\tau+1)} = \arg\min_{X_G} E(U_q^{(\tau+1)}(k), w_q^{(\tau+1)}, X_G), \qquad (2.12)$$

що призводить до $\hat{x}_{ki}^{(\tau+1)} = \sum_{q=1}^{m} (U_q^{(\tau+1)}(k))^{\beta} w_{qi}^{(\tau+1)} \Big/ \sum_{q=1}^{m} (U_q^{(\tau+1)}(k))^{\beta}$.

Отже, обробка інформації за допомогою цього алгоритму організована як послідовність:

$$w_q^{(0)} \to U_q^{(1)} \to \hat{x}_{ki}^{(1)} \to w_q^{(1)} \to U_q^{(2)} \to \ldots \to w_q^{(\tau)} \to U_q^{(\tau+1)} \to \\ \to \hat{x}_{ki}^{(\tau+1)} \to w_q^{(\tau+1)} \to \ldots \to w_q^{(Q)}. \tag{2.13}$$

2.4 Детальний розбір алгоритму кластеризації з відсутніми даними

В імовірнісних алгоритмах кластеризації цільова функція має вигляд:

$$E(U_q(k), w_q, \mu_q) = \sum_{k=1}^{N} \sum_{q=1}^{m} U_q^{\beta}(k) D^2(\tilde{x}_k, w_q) + \sum_{q=1}^{m} \mu_q \sum_{k=1}^{N} (1 - U_q(k))^{\beta}, \tag{2.14}$$

де скалярний параметр $\mu \geq 0$ визначає відстань, на якому рівень членства дорівнює 0,5, тобто якщо $D^2(\tilde{x}_k, w_q) = \mu_q$, то $w_q(k) = 0.5$.

Його мінімізація призводить до результату:

$$\begin{cases} U_q^{(\tau+1)}(k) = \dfrac{1}{1 + \left(\dfrac{D^2(\tilde{x}_k, w_q^{(\tau)})}{\mu_q^{(\tau)}}\right)^{\frac{1}{\beta-1}}}, \\ w_q^{(\tau+1)} = \dfrac{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta} \tilde{x}_k}{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta}}, \\ \mu_q^{(\tau+1)} = \dfrac{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta} D^2(\tilde{x}_k, w_q^{(\tau+1)})}{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta}}. \end{cases} \tag{2.15}$$

Мінімізація здійснена на $U_q(k)$, w_q, μ_q. В онлайн обробці така форма алгоритму (2.15) може бути записана як:

$$\begin{cases} U_q(k+1) = \frac{1}{1+(\frac{D^2(\tilde{x}_{k+1}, w_q(k))}{\mu_q(k)})^{\frac{1}{\beta-1}}}, \\ w_q(k+1) = w_q(k) + \eta(k+1)U_q^{\beta}(k+1)(\tilde{x}_{k+1} - w_q(k)), \\ \mu_q(k+1) = \frac{\sum_{p=1}^{k+1} U_q^{\beta}(p) D^2\left(\tilde{x}_p, w_q(k+1)\right)}{\sum_{p=1}^{k+1} U_q^{\beta}(p)}. \end{cases} \quad (2.16)$$

Така форма алгоритму відрізняється від форми (2.15) тільки у вигляді функції $U_q^{\beta}(k+1)$. Таким чином, нечітка кластеризація базується на конкурентному самонавчанні Кохонена.

Розглядаючи ситуацію з відсутніми спостереженнями використовуючи стратегію формули (2.14) в якості цільової функції імовірнісної нечіткої кластеризації використовуємо формулу:

$$E(U_q(k), w_q, \mu_q, X_G) = \sum_{k=1}^{N} \sum_{q=1}^{m} U_q^{\beta}(k) D^2(\hat{x}_k, w_q) + \sum_{q=1}^{m} \mu_q \sum_{k=1}^{N} (1 - U_q(k))^{\beta}. \quad (2.17)$$

Мінімізація цього виразу призводить до системи очевидних відносин:

$$\begin{cases} U_q^{(\tau+1)}(k) = \frac{1}{1+(\frac{D^2(\hat{x}_k^{(\tau)}, w_q^{(\tau)})}{\mu_q^{(\tau)}})^{\frac{1}{\beta-1}}}, \\ w_q^{(\tau+1)} = \frac{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta} \hat{x}_k^{(\tau)}}{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta}}, \\ \hat{x}_{ki}^{(\tau+1)} = \frac{\sum_{q=1}^{m} (U_q^{(\tau+1)}(k))^{\beta} w_{qi}^{(\tau+1)}}{\sum_{q=1}^{m} (U_q^{(\tau+1)}(k))^{\beta}}, \\ \mu_q^{(\tau+1)} = \frac{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta} D^2(\hat{x}_k^{(\tau+1)}, w_q^{(\tau+1)})}{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta}}. \end{cases} \quad (2.18)$$

Аналогічно моделюємо процес кластеризації з відсутніми спостереженнями, що засновано на оптимальній стратегії завершення:

$$\begin{cases} U_q^{(\tau+1)}(k+1) = \dfrac{1}{1+\left(\dfrac{\left\|\hat{x}_{k+1}^{(\tau)} - w_q(k)\right\|^2}{\mu_q^{(\tau)}}\right)^{\frac{1}{\beta-1}}}, \\ \hat{x}_{k+1,i}^{(\tau+1)} = \dfrac{\sum_{q=1}^{m}(U_q^{(\tau+1)}(k+1))^{\beta} w_{qi}(k)}{\sum_{q=1}^{m}(U_q^{(\tau+1)}(k+1))^{\beta}}, \\ \mu_q^{(\tau+1)}(k+1) = \dfrac{\sum_{p=1}^{k+1}(U_q^{(\tau+1)}(p))^{\beta}\left\|\hat{x}_p^{(\tau+1)} - w_q(k)\right\|^2}{\sum_{p=1}^{k+1}(U_q^{(\tau+1)}(p))^{\beta}}, \\ w_q(k+1) = w_q(k) + \eta(k+1)(U_q^{(Q)}(k+1))^{\beta}(\hat{x}_{k+1}^{(Q)} - w_q(k)), \end{cases} \quad (2.19)$$

або для прискорення часу обробки:

$$\begin{cases} U_q^{(\tau+1)}(k+1) = \dfrac{1}{1+\left(\dfrac{\left\|\hat{x}_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)\right\|^2}{\mu_q^{(\tau)}}\right)^{\frac{1}{\beta-1}}}, \\ w_q^{(0)}(k+1) = w_q^{(Q)}(k), \\ w_q^{(\tau+1)}(k+1) = w_q^{(\tau)}(k+1) + \eta(k+1)(U_q^{(\tau+1)}(k+1))^{\beta} * (\hat{x}_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)), \\ \hat{x}_{k+1,i}^{(\tau+1)} = \dfrac{\sum_{q=1}^{m}(U_q^{(\tau+1)}(k+1))^{\beta} w_{qi}^{(\tau+1)}(k+1)}{\sum_{q=1}^{m}(U_q^{(\tau+1)}(k+1))^{\beta}}, \\ \mu_q^{(\tau+1)}(k+1) = \dfrac{\sum_{p=1}^{k+1}(U_q^{(\tau+1)}(p))^{\beta}\left\|\hat{x}_p^{(\tau+1)} - w_q^{(\tau+1)}(k+1)\right\|^2}{\sum_{p=1}^{k+1}(U_q^{(\tau+1)}(p))^{\beta}}. \end{cases} \quad (2.20)$$

Перевага обох алгоритмів полягає в тому, що вони можуть використовуватись в режимі онлайн для виявлення появи нових кластерів, але алгоритм (2.20), хоч він і більш громіздкий, має перевагу оскільки використовує ваги $w_q^{(0)}(k+1)$ та $w_q^{(\tau+1)}(k+1)$ для швидшої збіжності, особливо в умовах де важливе швидке наближення до оптимального значення.

Також особливістю та перевагою є те, що в алгоритмах (2.19) та (2.20) нема необхідності накопичувати оброблювальний зразок, що важливо в задачах, таких як наприклад Web Mining, де обробляється велика кількість інформації.

Враховуючи усі особливості алгоритмів (2.19) та (2.20) проведено детальний покроковий розбір алгоритму, що наведено на рисунку 2.1.

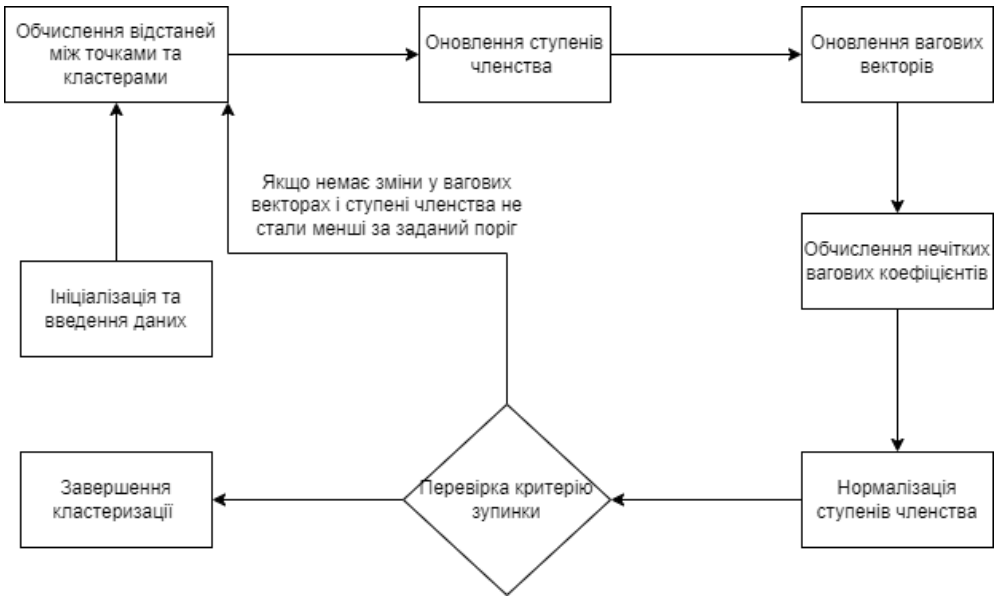

Рисунок 2.1 – Схема кластеризації з викривленими або відсутніми даними

Покроковий розбір алгоритму:

Крок 1. Ініціалізація та введення даних. На початку необхідно ініціалізувати вагові вектори $w_q^{(0)}(k)$ для кожного кластера, випадковим чином, або на основі початкових даних. Стартові значення ступенів членства $U_q^{(0)}(k)$ також повинні бути визначені.

Крок 2. Обчислення відстаней між точками та кластерами. Для кожної точки q необхідно визначити відстань до кожного центроїда кластерів w_q.

Крок 3. Оновлення ступенів членства. На цьому кроці оновлюються ступені членства кожної точки до кожного кластера. Ступінь членства точки q до кластера $k + 1$ залежить від відстані між ними та вагових коефіцієнтів.

Крок 4. Оновлення вагових векторів. Ваги центроїдів кластерів оновлюються на основі поточних значень ступенів членства. Для кожної точки оновлюються відповідні вагові вектори.

Крок 5. Обчислення нечітких вагових коефіцієнтів. Вагові коефіцієнти використовуються для модифікації ступенів членства. Ці коефіцієнти зважують ступінь членства кожної точки на основі обчислених відстаней та обчислюються на основі нечіткої логіки, що дозволяє зменшити похибки в кластеризації.

Крок 6. Нормалізація ступенів членства. Нормалізація ступенів членства до того, щоб їх сума по всім кластерам дорівнювала 1. Це забезпечує правильну інтерпретацію кластерів та їх вплив на кожну точку.

Крок 7. Перевірка критерію зупинки. На цьому кроці перевіряється чи зміни у вагових векторах і ступенях членства стали меншими за заданий поріг. Якщо так, то алгоритм закінчується, а якщо ні, то перехід на Крок 2 (обчислення відстаней між точками та кластерами).

Крок 8. Завершення кластеризації. Алгоритм завершується після досягнення умов збіжності. Виводяться результати кластеризації, зокрема вагові вектори кластерів і ступені членства точок.

Розглянутий алгоритм кластеризації демонструє структурований підхід до аналізу даних, базуючись на ітеративному процесі. Починаючи з ініціалізації вагових векторів і ступенів членства, алгоритм поступово вдосконалює результати, використовуючи інформацію про відстані між точками і центроїдами кластерів. Важливу роль відіграє оновлення вагових векторів і ступенів членства, що дозволяє адаптивно уточнювати структуру кластерів залежно від характеристик даних.

Обчислення нечітких вагових коефіцієнтів додає алгоритму гнучкості, враховуючи різний ступінь невизначеності в даних. Процес нормалізації ступенів членства гарантує, що результати кластеризації є математично коректними і піддаються інтерпретації. Завдяки критерію зупинки забезпечується стабільність роботи алгоритму, оскільки ітерації припиняються, коли зміни стають незначними.

Завершення алгоритму дозволяє отримати чіткі результати кластеризації у вигляді вагових векторів і ступенів членства, які надають повну інформацію про структуру даних і взаємозв'язки між точками. Загалом, мережа Кохонена є базовою структурою цих формул, а нечітка логіка додає гнучкості алгоритмам , дозволяючи об’єктам частково належати до кількох кластерів одночасно.

3 ДОСЛІДЖЕННЯ НЕЙРО-ФАЗЗІ МЕРЕЖІ КОХОНЕНА ДЛЯ КЛАСТЕРИЗАЦІЇ ВИКРИВЛЕНИХ ДАНИХ

3.1 Методологія проведення експериментального дослідження

Методологія проведення порівняльного аналізу алгоритму імовірнісно-адаптивної нечіткої кластеризації з відсутніми даними на основі оптимальної стратегії завершення полягає у порівнянні з класичними алгоритмами кластеризації викривлених даних по деяким критеріям. Порівняльний аналіз включає в себе декілька етапів, що дозволяє отримати надійний результат.

У ході дослідження було використано два датасети Iris UCI та Wine UCI для більш детального порівняння [32, 33]. Для вдалої реалізації необхідно провести попередню обробку даних. У даному дослідженні обробка даних включає наступні кроки:

Крок 1. Завантаження та ознайомлення з даними.

Крок 2. Попередній аналіз даних. На цьому кроці було проведено попередній огляд структури даних, визначено кількість зразків, характеристик, перевірено наявність пропущених значень та статистичні характеристики.

Крок 3. Аналіз розподілу та візуалізація даних.

Крок 4. Нормалізація або стандартизація ознак.

Для порівняльного аналізу було обрано два алгоритму кластеризації:

– метод нечітких *c*-середніх (FCM);

– алгоритм Густафсона-Кесселя.

Для оцінки якості кластеризації важливо застосовувати критерії ефективності, які дозволяють виміряти, наскільки успішно алгоритм кластеризації розділяє дані на окремі групи. Для порівняння ефективності кластеризації було обрано шість критеріїв ефективності, що розбиваються на кластери:

– коефіцієнт розподілу (PC);

– ентропія класифікації (CE);

– індекс компактності і розділення (SC);

– індекс розділення (SI);

– індекс Хіе-Бені (XB);

– індекс Данна (DI).

3.1.1 Критерії ефективності кластеризації

При оцінюванні якості кластеризації важливо використовувати критерії ефективності, які допомагають виміряти, наскільки добре алгоритм кластеризації поділяє дані на групи [34]. Детальний розбір використаних у досліджені критеріїв ефективності, що використовуються для оцінки результатів кластеризації наведено нижче.

Коефіцієнт розподілу (Partition Coefficient, PC) визначає, наскільки чітко дані поділені на кластери. Він базується на матриці належності U, де кожен елемент u_{ij} показує ступінь належності об'єкта i до кластера j.

Формула для коефіцієнта розподілу:

$$PC = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^2, \tag{3.1}$$

де n – кількість об'єктів;

c – кількість кластерів.

Значення PC знаходиться в діапазоні від 1/c до 1. Чим ближче значення до 1, тим чіткіше кластеризовані дані. Проте високий коефіцієнт не обов'язково означає гарну кластеризацію, оскільки він не враховує можливість перекриття між кластерами.

Ентропія класифікації (Classification Entropy, CE) вимірює нечіткість кластеризації, оцінюючи, наскільки розподілені об'єкти між кластерами. Вона

також базується на матриці належності U і характеризує рівень невизначеності у кластеризації.

Формула ентропії класифікації:

$$CE = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}\ln\left(u_{ij}\right). \quad (3.2)$$

Значення *CE* варіюється від 0 до $ln(c)$. Чим менше значення *CE*, тим менш розмитою є кластеризація. Низьке значення ентропії вказує на те, що об'єкти чітко належать до одного кластера.

Індекс силуету (Silhouette Coefficient, SC) є одним з найпоширеніших критеріїв для оцінки якості кластеризації. Він поєднує дві властивості – компактність (наскільки об'єкти близькі до центру свого кластера) та розділення (наскільки об'єкти віддалені від інших кластерів).

Форма для коефіцієнтів силуету:

$$SC = \frac{1}{n}\sum_{i=1}^{n}\frac{b(i)-a(i)}{\max\left(a(i),b(i)\right)}, \quad (3.3)$$

де $a(i)$ – середня відстань від об'єкта *i* до інших об'єктів у його кластері;

$b(i)$ – середня відстань від об'єкта i до об'єктів найближчого сусіднього кластера.

Значення *SC* варіюється від -1 до 1. Значення ближче до 1 означають хороше розділення між кластерами та компактність у межах кожного кластера, значення близьке до 0 вказує на перекриття кластерів, а від'ємні значення означають, що об'єкти частіше відносяться до сусіднього кластера.

Індекс розділення (Separation Index, SI) оцінює, наскільки добре кластери відокремлені один від одного. Він базується на відстанях між центроїдами кластерів, щоб виміряти віддаленість кожного кластера від інших кластерів.

Функція для індексу розділення:

$$SI = \frac{\sum_{i=1}^{c}\sum_{j=1,j\neq i}^{c}\frac{1}{d(v_i,v_j)}}{c\times(c-1)}, \quad (3.4)$$

де $d(v_i, v_j)$ – відстань між центроїдами кластерів *i* та *j*.

Чим менше значення *SI*, тим краще кластери розділені. Низьке значення S означає, що центри кластерів розташовані далеко один від одного.

Індекс Хіе-Бені (Xie-Beni Index, XB) поєднує компактність і розділення кластерів, вимірюючи відношення між середньою варіацією всередині кластерів і відстанню між найближчими кластерами. Це дозволяє оцінити, наскільки добре дані розділені на кластери і чи не перекриваються вони.

Формула для індексу Хіе-Бені:

$$XB = \frac{\sum_{i=1}^{n}\sum_{j=1}^{c}u_{ij}^{m}\|x_i - v_j\|^2}{n\times min_{i\neq j}\|v_i - v_j\|^2}, \quad (3.5)$$

де $\|x_i - v_j\|$ – відстань від точки x_i до центра кластера v_i;

$min_{i\neq j}\|v_i - v_j\|$ – мінімальна відстань між центрами двох різних кластерів.

Чим менше значення *XB*, тим краща якість кластеризації. Низьке значення індексу Хіе-Бені свідчить про компактні кластери, які добре відокремлені один від одного.

Індекс Данна (Dunn Index, DI) розроблений для оцінки якості кластеризації, щоб знаходити кластеризації з компактними і добре розділеними кластерами. Він використовує мінімальну відстань між центрами різних кластерів і максимальний діаметр кластера для оцінки розділення і компактності.

Його перевагою є те, що він враховує як внутрішню компактність кластерів, так і відокремленість між ними, та являється безрозмірною величиною, що робить його універсальним для різних задач кластеризаці.

Формула для індексу Данна:

$$DI = \frac{min_{i \neq j} d(v_i, v_j)}{max_k \delta(C_k)}, \quad (3.6)$$

де $d(v_i, v_j)$ – відстань між центрами кластерів *i* та *j*;

$\delta(C_k)$ – діаметр кластера C_k, який визначається як максимальна відстань між будь-якими двома точками в цьому кластері.

Чим більше значення *DI*, тим краща якість кластеризації. Високе значення *DI* свідчить про те, що кластери є компактними і добре відокремленими один від одного.

Отже, кожен з наведених критеріїв має свою специфіку і підходить для різних цілей кластеризації. Застосування кількох критеріїв одночасно дозволяє краще оцінити якість кластеризації і вибрати оптимальну кількість кластерів для даних.

3.1.2 Алгоритм нечіткої кластеризації *c*-середніх

Алгоритм *c*-середніх (FCM) є одним з популярних методів нечіткої кластеризації, який дозволяє кожному об’єкту належати одночасно до кількох кластерів з певним ступенем належності. Це корисно для роботи з даними, де межі між кластерами розмиті, і об’єкти можуть частково належати до кількох груп. Алгоритм FCM спрямований на мінімізацію функції нечіткої вартості, яка визначає ступінь схожості об’єктів із центрами кластерів.

Основні етапи роботи FCM:

Етап 1. Ініціалізація:

– вибирається кількість кластерів *c* – кількість центрів кластерів;

– ініціалізується матриця належності U випадковими значеннями, де кожен елемент u_{ij} визначає ступінь належності *i*-го об’єкта до *j*-го кластера.

$u_{ij} \in [0,1]$, і сума належностей об’єкта до всіх кластерів повинна дорівнювати 1.

Етап 2. Визначення центрів кластерів, центр кожного кластера v_{ij} обчислюється як зважене середнє всіх точок на основі їхньої належності до цього кластера:

$$v_{ij} = \frac{\sum_{i=1}^{n} u_{ij}^{m} \times x_i}{\sum_{i=1}^{n} u_{ij}^{m}}, \tag{3.7}$$

де x_i – координати *i*-го об’єкта;

u_{ij} – ступінь належності об’єкта до кластера;

m – параметр нечіткості (зазвичай $m = 2$), який визначає рівень розмитості кластерів.

Етап 3. Оновлення матриці належності, тобто для кожного об’єкта обчислюється новий ступінь належності до кожного кластера на основі відстані до центру кластера. Це дозволяє об’єкту одночасно належати до кількох кластерів. Формула оновлення матриці належності:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}}, \tag{3.8}$$

де $\|x_i - v_j\|$ – відстань від об’єкта x_i до центру кластера v_j.

Найчастіше для визначення відстані використовується Евклідова відстань.

Етап 4. Перевірка умови зупинки, що обчислюється за формулою значення функції вартості:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \|x_i - v_j\|^2. \tag{3.9}$$

Алгоритм завершує роботу, коли різниця значень функції вартості для двох ітерацій є меншою за деякий малий поріг зупинки ϵ, або коли максимальна кількість ітерацій досягнута.

Блок-схема алгоритму FCM зображена на рисунку 3.1.

Рисунок 3.1 – Блок-схема алгоритму *c*-середніх

Основні параметри алгоритму FCM:

– кількість кластерів c визначає, скільки центрів кластерів буде знайдено. Цей параметр встановлюється вручну, залежно від задачі;

– параметр нечіткості m зазвичай обирається в межах [1,5, 3], стандартно – $m = 2$. Чим вище значення m, тим більше розмиття між кластерами;

– поріг зупинки ϵ визначає чутливість алгоритму до змін між ітераціями, наприклад, $\epsilon = 0{,}001$.

Переваги алгоритму FCM:

– розмиті межі, що дозволяє об’єктам належати до кількох кластерів одночасно;

– гнучкість, яка підходить для складних наборів даних, де традиційна кластеризація з жорсткими межами може бути неефективною.

Недоліки алгоритму FCM:

– чутливість до початкових значень, а саме початкова ініціалізація центроїдів може впливати на результат, а також на стабільність кластеризації;

– потреба у виборі кількості кластерів, необхідно заздалегідь визначити кількість кластерів, що може бути непросто для нових даних;

– чутливість до шуму та аномалій, алгоритм FCM є менш стійким, оскільки шум та аномалії можуть значно зміщувати центри кластерів.

Підсумовуючи переваги та недоліки алгоритму, можна сказати, що алгоритм Fuzzy *C*-Means є потужним інструментом для кластеризації, особливо коли дані мають розмиті межі між кластерами. Завдяки гнучкій структурі, яка дозволяє кожному об’єкту частково належати до кількох кластерів, FCM є ефективним у задачах, де традиційні методи кластеризації можуть бути недостатніми. Однак цей алгоритм є чутливим до шуму та початкових значень, що слід враховувати під час його використання.

3.1.3 Алгоритм кластеризації Густафона-Кесселя

Алгоритм Густафсона-Кесселя (GK) – це один з алгоритмів нечіткої кластеризації, який є розширенням алгоритму FCM. Його особливість полягає в тому, що він дозволяє кластерам мати довільну форму, не обмежуючись сферичною, як у випадку FCM. Для цього алгоритм використовує адаптивні матриці, що дозволяє йому краще кластеризувати дані з викривленими формами та різними розмірами кластерів.

У класичному FCM для обчислення ступеня належності об’єктів до центрів кластерів використовується Євклідова відстань, що передбачає сферичну форму кластерів. У GK алгоритмі використовується адаптивна метрика, яка дозволяє кожному кластеру мати власну форму і орієнтацію. Це

досягається через використання матриць форми (матриць коваріації) для кожного кластера, які адаптуються в процесі навчання.

Основні етапи роботи алгоритму Густафсона-Кесселя:

Етап 1. Ініціалізація:

– визначається кількість кластерів c;

– ініціалізується матриця належності U випадковими значеннямидля кожного об’єкта щодо кожного кластера, причому сума ступенів належності кожного об’єкта до всіх кластерів повинна дорівнювати 1;

– ініціалізуються початкові центри кластерів u_{ij} для $j = 1, \dots, c$.

Етап 2. Оновлення центрів кластерів такий самий як і Етап 2 алгоритму FCM.

Етап 3. Оновлення матриці форми (матриці коваріації): матриця форми A_j для кожного кластера j обчислюється так, щоб вона враховувала розташування точок щодо центру кластера:

$$A_j = \frac{det(F_j)^{\frac{1}{d}}}{trace(F_j)} \times F_j, \tag{3.10}$$

$$F_j = \frac{\sum_{i+1}^{n} u_{ij}^m \times (x_i - v_j)(x_i - v_j)^T}{\sum_{i=1}^{n} u_{ij}^m}, \tag{3.11}$$

де F_j – матриця коваріації;

d – розмірність даних;

$\frac{det(F_j)^{\frac{1}{d}}}{trace(F_j)}$ – нормалізує форму кластера, зберігаючи об’єм.

Етап 4. Оновлення ступенів належності: ступінь належності u_{ij} обчислюється з використанням відстані Махаланобіса, що враховує матрицю форми A_j:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_i - v_j\|_{A_j}}{\|x_i - v_k\|_{A_k}}\right)^{\frac{2}{m-1}}}, \quad (3.12)$$

де $\|x_i - v_j\|_{A_j} = \sqrt{(x_i - v_j)^T A_j (x_i - v_j)}$.

Етап 5. Перевірка умови зупинки, під час якої обчислюється значення функції вартості:

$$J_m = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m \|x_i - v_j\|_{A_j}^2. \quad (3.13)$$

Алгоритм завершує роботу, коли різниця значень функції вартості для двох ітерацій є меншою за деякий малий поріг, або коли досягнута максимальна кількість ітерацій.

Основні параметри алгоритму Густафсона-Кесселя такі самі як в алгоритму FCM, оскільки він є розширенням цього алгоритму.

Переваги алгоритму Густафсона-Кесселя:

– адаптивна форма кластерів, тобто алгоритм дозволяє кластеризувати дані з різною формою та орієнтацією, оскільки кожен кластер має власну матрицю форми;

– гнучкість, завдяки якій алгоритм ефективно працює з викривленими та неоднорідними даними, де кластери мають різні розміри та напрямки;

– збереження топології, враховуючи локальну структуру даних, GK-кластери часто відображають реальні зв’язки між точками.

Недоліки алгоритму Густафсона-Кесселя:

– чутливість до початкових умов;

– чутливість до шуму;

– вибір кількості кластерів.

Через чутливість до початкових умов, початкові значення центрів та ступенів належності можуть вплинути на кінцевий результат. Чутливість до шуму означає що алгоритм може бути чутливим до шумових точок, оскільки ті можуть змінювати матрицю форми і, відповідно, форму кластерів. А також недоліком є вибір кількості кластерів, що потрібно задавати заздалегідь, що може бути складним для нових даних.

Блок-схема алгоритму Густафсона-Кесселя показана на рисунку 3.2.

Рисунок 3.2 – Блок-схема алгоритму Густафсона-Кесселя

Враховуючи переваги та недоліки алгоритму Густафсона-Кесселя, можна сказати, що даний алгоритм є ефективним методом кластеризації для даних зі складною структурою, де кластери можуть мати різну форму, орієнтацію та розмір. На відміну від класичного FCM, GK використовує адаптивну метрику на основі матриць коваріації, що дозволяє алгоритму враховувати локальні особливості даних. Однак, при роботі з GK-алгоритмом важливо враховувати чутливість до початкових значень та можливий вплив шуму на результати кластеризації.

3.2 Викривлені дані

Викривлені дані – це дані, які мають нерівномірний або специфічний розподіл, часто зі складною структурою, що відрізняється від звичайних або симетричних розподілів, таких як нормальний розподіл. Такі дані можуть містити аномалії, викиди, різку асиметрію, перекоси або сильно витягнуті форми кластерів. У роботі з викривленими даними виникають специфічні труднощі, адже стандартні алгоритми обробки та аналізу, наприклад, кластеризації або регресії, можуть давати некоректні результати [18].

Характеристики викривлених даних:

– асиметрія розподілу. Часто викривлені дані мають асиметричний розподіл, де велика кількість даних сконцентрована в одній частині діапазону значень. Це призводить до появи довгого «хвоста» у напрямку великих або малих значень;

– наявність аномалій або викидів. Дані можуть містити точки, що суттєво відрізняються від основного масиву. Такі точки можуть сильно впливати на результати обробки, наприклад, зсувати середнє значення або центр кластера;

– складні форми кластерів. Кластери у викривлених даних часто мають нестандартну форму, наприклад, витягнуті або криволінійні структури, що ускладнює їх розпізнавання стандартними алгоритмами;

– неоднорідна щільність даних. Викривлені дані можуть бути розподілені з різною щільністю, де одна частина містить густо сконцентровані точки, а інша є значно розрідженішою. Це ускладнює кластеризацію та виділення груп.

Причини викривлених даних:

– помилки вимірювань або збору даних, тобто технічні збої, людські помилки або особливості обладнання можуть призводити до викривлень у даних;

– неповна вибірка, тому дані можуть бути зібрані нерівномірно або містити тільки певну частину популяції, що викликає перекоси;

– природні фактори, у реальних умовах розподіли часто не є ідеальними, так наприклад, доходи людей або вік населення не мають нормального розподілу, а схильні до асиметричних форм.

Викривлені дані ускладнюють аналіз і обробку, тому важливим є знаходження способу роботи з ними. Методи роботи з викривленими даними:

– нормалізація або стандартизація;

– використання стійких алгоритмів;

– виявлення та обробка аномалій;

– застосування методів зниження впливу викидів.

Зміна масштабу та нормалізація даних може допомогти зменшити вплив викривлень. Це включає логарифмічну трансформацію, Box-Cox трансформацію та інші методи, які можуть виправити асиметрію [35]. Також деякі алгоритми кластеризації, такі як алгоритм Густафсона-Кесселя або алгоритм DBSCAN, краще працюють з викривленими даними, оскільки вони здатні адаптуватися до нестандартної форми та неоднорідної щільності [36].

Використання методів для виявлення аномалій, таких як ізольований ліс або методи, засновані на відстанях, дозволяє ідентифікувати та вилучати аномальні точки перед аналізом даних. Та не менш важливим є застосування методів зниження впливу викидів. Наприклад, медіанна регресія є стійкішою до викривлень і може бути корисною в задачах регресії з аномаліями.

Викривлені дані є поширеним явищем у реальних задачах, і їх обробка вимагає спеціальних методів і підходів. Використання адаптивних алгоритмів, методів виявлення аномалій і корекції розподілу дозволяє підвищити якість аналізу та отримати більш точні результати навіть при значних викривленнях [37].

Використання імовірнісно-адаптивної нечіткої кластеризації для роботи з викривленими даними. Імовірнісно-адаптивна нечітка кластеризація – це підхід, який об’єднує методи нечіткої кластеризації та адаптивного

налаштування параметрів для роботи з викривленими даними. Цей метод є ефективним у випадках, коли дані містять значні шуми, викривлення або аномалії, які ускладнюють їх поділ на чіткі кластери.

Досліджуваний алгоритм кластеризації (2.20) може працювати з викривленими даними, оскільки він враховує адаптивне оновлення центрів кластерів і ступеня належності даних до кластерів. Основні компоненти алгоритму та їх роль у кластеризації викривлених даних:

– адаптивне оновлення ступеня належності. У випадку викривлених даних, де кластери можуть мати нерівномірну форму або щільність, використання показника міри належності дозволяє точкам частково належати до декількох кластерів. Це особливо важливо для розмитих або неоднорідних розподілів, де дані не завжди чітко відносяться до одного кластера;

– оновлення центрів кластерів. Центри кластерів оновлюються з урахуванням поточної міри належності та навчальної швидкості. Це дозволяє центрам кластерів адаптуватися до форми і розподілу даних, що особливо важливо для викривлених даних. Таким чином, кластери можуть підлаштовуватися під дані, формуючи центри, які не обов’язково є симетричними або мають однакову щільність;

– зважені середні значення і адаптація. Формули для обчислення нових центрів кластерів і міри належності базуються на зважених середніх значеннях. Це дозволяє алгоритму враховувати різну щільність та розподіл даних, що є важливим при кластеризації викривлених даних. Наприклад, якщо один кластер має більшу щільність або іншу форму, ніж інші, центр цього кластера буде оновлюватись відповідно до цих особливостей, що дозволяє адаптуватись до нетипового розподілу;

– метрика відстані. Використання метрики відстані у алгоритму, наприклад, Евклідової або іншої специфічної для задачі, дозволяє гнучко вимірювати схожість між точками і центрами кластерів. У випадку викривлених даних, обчислення відстані може бути налаштоване таким чином, щоб краще враховувати нерівномірний розподіл;

– застосування параметра β. Параметр β, що визначає ступінь розмитості належності, дозволяє налаштовувати міру належності так, щоб дані могли частково належати до кількох кластерів одночасно. Це особливо корисно для кластеризації викривлених даних, оскільки вони можуть містити точки, які природно розташовані на межі кількох кластерів або мають нечітку належність [38].

Виходячи з ролі компонентів алгоритму для кластеризації викривлених даних, можна сказати, що ця формула дозволяє адаптивно змінювати центри кластерів і міру належності з урахуванням викривлених даних, забезпечуючи більшу гнучкість при роботі з нестандартними, асиметричними або неоднорідними розподілами. Використання зважених середніх, параметра β, а також можливість оновлення центрів кластерів допомагає алгоритму підлаштовуватися під структуру викривлених даних і забезпечувати більш точну кластеризацію.

3.3 Програмна реалізація та результати порівняння методів кластеризації

Для програмної реалізації моделі було вирішено використовувати мову програмування C++.

Реалізація алгоритмів кластеризації мовою програмування C++ має свої особливості та специфічні вимоги, оскільки ця мова є потужним інструментом для роботи з даними завдяки своїй високій продуктивності, можливостям низькорівневого контролю і гнучкості в управлінні пам’яттю.

Ключові аспекти, які впливають на ефективність та складність реалізації алгоритмів кластеризації на мові програмування C++:

– ефективне управління пам’яттю;

– використання стандартної бібліотеки шаблонів (STL);

– оптимізація швидкодії;

– висока точність у роботі з числами з плаваючою комою;

– розділення алгоритму на функції та класи;

– реалізація специфічних алгоритмів кластеризації;

– робота з бібліотеками для лінійної алгебри.

Ефективне управління пам’яттю поділяється на роботу з динамічними структурами даних та контроль за витоком пам’яті. Кластеризація часто вимагає обробки великих обсягів даних, які можуть не поміщатися в статичні масиви. Тому в C++ використовуються динамічні структури даних, такі як вектори (std::vector), які дозволяють автоматично змінювати розмір в процесі виконання. Щодо контролю за витоком пам’яті C++ вимагає уваги до управління пам’яттю, особливо якщо застосовуються сирі вказівники або масиви. Наприклад, важливо звільняти динамічно виділену пам’ять після використання, щоб уникнути витоків пам’яті. Використання розумних вказівників (std::unique_ptr, std::shared_ptr) може спростити цей процес.

Використання стандартної бібліотеки шаблонів для кластеризації викривлених даних має дві важливі переваги, структуру даних та алгоритми STL. C++ надає багато корисних контейнерів (таких як std::vector, std::map, std::set), які спрощують зберігання та обробку даних. Наприклад, для реалізації алгоритму *k*-середніх або алгоритму Кохонена зручно використовувати std::vector для зберігання координат центрів кластерів і набору точок даних. Також стандартні алгоритми (наприклад, std::sort, std::accumulate) спрощують реалізацію кластеризації, дозволяючи швидше виконувати операції над контейнерами.

До оптимізації швидкодії відносяться математичні обчислення та багатопотоковість даних. Алгоритми кластеризації часто включають в себе велику кількість математичних обчислень, таких як обчислення відстаней, середніх значень і метрик подібності. У C++ для цього можна використовувати спеціалізовані бібліотеки, такі як Eigen або Armadillo, які оптимізовані для роботи з матрицями та векторами [39]. Також C++ підтримує багатопотоковість (наприклад, через бібліотеку <thread>), що дозволяє

паралелізувати обчислення в алгоритмах, де це можливо, щоб прискорити процес кластеризації великих наборів даних;

Кластеризація часто передбачає роботу з числами з плаваючою точкою (floating-point), що може призвести до накопичення помилок у випадку неточної обробки. Важливо використовувати відповідні типи даних (float, double, або long double) для забезпечення потрібної точності. Також при обчисленні відстаней, особливо Евклідової, потрібно враховувати можливість накопичення числових помилок і мінімізувати їх, наприклад, шляхом використання стабільних алгоритмів обчислення.

Перевагою розділення алгоритму на функції та класи є модульність та інкапсуляція даних. Для більшої зручності підтримки та розширення коду алгоритм кластеризації можна реалізувати у вигляді класів і функцій. Наприклад, можна створити клас Cluster, який зберігатиме інформацію про один кластер (центр, кількість точок), і клас KMeans, який реалізує алгоритм *k*-середніх. Також важливо інкапсулювати дані та операції з ними, щоб забезпечити безпеку доступу та спростити модифікацію коду в майбутньому.

До реалізації специфічних алгоритмів кластеризації належить нечіткий алгоритм *c*-середніх (FCM), а також самоорганізовані карти Кохонена (SOM). Алгоритм FCM вимагає додаткового рівня складності для обчислення ступеня належності кожної точки до кластерів. Це можна реалізувати за допомогою матриці членства, яку зручно представляти у вигляді вектора векторів (std::vector<std::vector<double>>). Для реалізації SOM C++ використовує сітку нейронів, де кожен нейрон має свої ваги, і на кожній ітерації знаходиться ближчий нейрон до вхідного сигналу, після чого коригуються його ваги та ваги сусідніх нейронів. Це потребує реалізації топології (наприклад, сітки) і функції сусідства.

Щодо роботи з бібліотеками лінійної алгебри, реалізація складних алгоритмів кластеризації може потребувати роботи з великими матрицями та векторами. Для цього в C++ існують бібліотеки, такі як Eigen або Armadillo,

які дозволяють ефективно виконувати операції лінійної алгебри і забезпечують високий рівень продуктивності.

Отже, для реалізації алгоритму кластеризації з викривленими даними оптимальним є використання мови програмування C++.

Було проведено порівняння класичних алгоритмів кластеризації FCM, Густафсона-Кесселя та алгоритму імовірнісно-адаптивної нечіткої кластеризації на основі оптимальної стратегії завершення. Для порівняння важливо розглянути ключові особливості та підходи до роботи з викривленими даними [40, 41].

Алгоритм нейро-фаззі мережі Кохонена на основі оптимальної стратегії завершення. Даний алгоритм використовує адаптивний підхід, який змінює центр кластеру та параметри відповідно до викривлення вхідних даних. Формула враховує:

– ваги кластерів $w_q^{(\tau+1)}$ і центри кластерів $U_q^{(\tau+1)}$, що оновлюються на кожному етапі, базуючись на відстані до кожної точки;

– використання матриці належності та множинних коефіцієнтів ваги, що дозволяє адаптуватися до викривлених даних та забезпечує кращу кластеризацію у випадку асиметричних кластерів;

– використання ступеня нечіткості β що регулює рівень нечіткості або невизначеності між кластерами.

Основна особливість цього алгоритму полягає у використанні вагових коефіцієнтів і ступеня адаптивності, що дозволяє краще обробляти викривлені та неоднорідні дані. Він також більш гнучкий у визначенні форм кластерів, порівняно з іншими стандартними методами.

Класичний алгоритм кластеризації Fuzzy C-Means – це класичний алгоритм нечіткої кластеризації, який дозволяє точкам належати до кількох кластерів одночасно з різним ступенем належності.

Основні характеристики FCM:

– простота реалізації та зручність у використанні для базових випадків, коли кластери мають приблизно сферичну форму;

– формула членства u_{ij} визначається виключно на основі евклідової відстані між точками і центрами кластерів;

– FCM не працює з коваріаційними матрицями та має обмеження щодо форми кластерів, оскільки він краще підходить для симетричних розподілів даних.

Алгоритм Густафсона-Кесселя (GK) є покращенням FCM і додає можливість обробляти кластери з різною формою за допомогою коваріаційних матриць.

Основні характеристики GK:

– врахування коваріаційної матриці F_j для кожного кластера, що дозволяє кластеру мати еліптичну форму, тобто цей метод краще підходить для кластерів з різними формами та розмірами;

– оновлення значень членства залежить не тільки від відстані до центру кластера, але й від форми самого кластера (коваріаційної матриці), що дозволяє моделювати кластери з асиметричними формами;

– використовується для аналізу даних з несиметричним розподілом, проте вимагає більш складних обчислень.

Порівняльна таблиця характеристик алгоритмів наведено у таблиці 3.1.

Таблиця 3.1 – Результати порівнянь

Характеристика	**Алгоритм нейро-фаззі мережі Кохонена**	**Fuzzy C-Means (FCM)**	**Густафсон-Кессель (GK)**
1	**2**	**3**	**4**
Форма кластерів	Може адаптуватися до викривлених даних	Сферичні кластери	Еліптичні кластери
Використання ваг	Використовує ваги для адаптації	Відсутні	Відсутні

Продовження таблиці 3.1

1	2	3	4
Коваріаційна матриця	Ні	Ні	Так
Гнучкість для викривлених даних	Висока	Низька	Середня
Ступінь нечіткості (множинність)	Враховується за допомогою β	Враховується	Враховується
Обчислювальна складність	Висока через адаптивні параметри	Низька	Середня
Придатність для неоднорідних розподілів	Висока	Низька	Висока

Підсумовуючи порівняльний аналіз по основним критеріям можна сказати, що алгоритм нейро-фаззі мережі Кохонена на основі оптимальної стратегії завершення підходить для кластеризації сильно викривлених або неоднорідних даних. Він забезпечує адаптивність до складних структур даних за рахунок використання вагових коефіцієнтів, що підвищує точність, проте збільшує обчислювальну складність. Алгоритм Fuzzy *C*-Means є більш базовим підходом, підходить для симетричних або простих структур даних, але обмежений у випадках з викривленими або еліптичними кластерами. Алгоритм Густафсона-Кесселя ефективно працює з даними, що мають еліптичну форму кластерів, проте менш пристосований до сильно викривлених даних, як наведений алгоритм.

Таким чином, вибір алгоритму кластеризації залежить від форми та викривлення даних: для сферичних більше підходить алгоритм FCM, для еліптичних – алгоритм Густафсона-Кесселя, а для складних та

викривлених – досліджуваний алгоритм нейро-фаззі мережі Кохонена на основі оптимальної стратегії завершення.

3.4 Результати проведення дослідження

Для оцінки якості запропонованого алгоритму було проведено дослідження на двох стандартних зразках даних, а саме Wine і Iris UCI репозиторіїв.

Датасет Wine UCI містить чотири числові ознаки: sepal length, sepal width, petal length, petal width, а також мітку target, яка має три класи – типи ірисів Setosa, Versicolor, Virginica.

Датасет Iris UCI містить тринадцять числових ознак, таких як alcohol, malic acid, ash, alcalinity of ash, а також мітку target, яка представляє три класи типів вина.

Було обрано декілька критеріїв якості, що розбиваються на кластери, такі як: коефіцієнт розподілу (PC), ентропія класифікації (CE), індекс компактності і розділення (SC), індекс розділення (SI), індекс Хіе-Бені (XB), індекс Данна (DI). Для дослідження кластеризації з відсутніми даними оптимальним для експериментів вважається діапазон від 5% до 20% відсутніх даних. Цього зазвичай достатньо для оцінки стабільності алгоритму, а також для виявлення межі, коли відсутні дані починають суттєво впливати на якість кластеризації.

Для порівняння результатів дослідження було обрано відомі алгоритми кластеризації: кластеризація нечітких *C*-means (FCM) та алгоритм кластеризації Густафсона-Кесселя. Було проведено тестування на 10, 50 та 100 відсутніх значеннях, мовою програмування C++. У таблицях 3.2, 3.3 наведено результати дослідження з 10 відсутніми значеннями на обох датасетах Wine UCI та Iris UCI. У таблицях виділено кращі результати для кожного з критеріїв якості.

Таблиця 3.2 – Результати дослідження з 10 відсутніми значеннями на датасеті Wine UCI

Алгоритми	**Пропуск**	**Датасет Wine UCI**					
		PC	**CE**	**SC**	**SI**	**XB**	**DI**
Алгоритм нейро-фаззі мережі Кохонена	10	1,1640e-13	**-4,5675e-04**	**7,3872e+05**	2,7115e+08	2,7180e+08	0,0218
Fuzzy C-Means (FCM)		**0,7908**	0,3806	7,3348e-04	**6,8417e-06**	5,7110	0,0117
Густафсон-Кессель (GK)		0,5507	0,6393	8,5933	0,0483	**1,0750**	**0,1015**

Таблиця 3.3 – Результати дослідження з 10 відсутніми значеннями на датасеті Iris UCI

Алгоритми	**Пропуск**	**Датасет Iris UCI**					
		PC	**CE**	**SC**	**SI**	**XB**	**DI**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Алгоритм нейро-фаззі мережі Кохонена	10	9,1249e-07	**-4,6617e-04**	0,3733	48,7067	48,8056	**0,4010**
Fuzzy C-Means (FCM)		0,7617	0,4283	0,0143	**1,4946e-04**	3,8569	0,0275

Продовження таблиці 3.3

1	2	3	4	5	6	7	8
Густафсон-Кессель (GK)	10	**0,9462**	0,1145	**0,4789**	0,0032	**3,4618**	0,3398

Наступним було проведено таке саме дослідження, але з більшою кількістю відсутніх даних – 50 на обох датасетах Wine UCI та Iris UCI, що показано у таблицях 3.4, 3.5.

Таблиця 3.4 – Результати дослідження з 50 відсутніми значеннями на датасеті Wine UCI

Алгоритми	**Пропуск**	**Датасет Wine UCI**					
		PC	**CE**	**SC**	**SI**	**XB**	**DI**
Алгоритм нейро-фаззі мережі Кохонена	50	9,1249e-07	**-5,4992e-04**	**1,1834e+04**	4,1981e+06	4,2024e+06	0,0240
Fuzzy C-Means (FCM)		0,7399	0,3838	7,6110e-04	**7,1760e-06**	8,8618	0,0237
Густафсон-Кессель (GK)		**0,9422**	0,6010	4,7678	0,0268	**1,1703**	**0,1030**

Таблиця 3.5 – Результати дослідження з 50 відсутніми значеннями на датасеті Iris UCI

Алгоритми	**Пропуск**	**Датасет Iris UCI**					
		PC	**CE**	**SC**	**SI**	**XB**	**DI**
Алгоритм нейро-фаззі мережі Кохонена	50	9,1249e-07	**-4,6617e-04**	0,3775	48,7067	48,8301	**0,3365**
Fuzzy C-Means (FCM)		0,7399	0,4632	0,0174	**1,8345e-04**	4,4887	0,0355
Густафсон-Кессель (GK)		**0,9422**	0,1177	**0,5219**	0,0035	**3,4413**	0,3341

У процесі дослідження, аналогічно до попередніх етапів, було збільшено кількість відсутніх даних до 100 значень на обох наборах даних, що дозволило створити більш складні умови для перевірки алгоритмів. Це дослідження стало необхідним кроком для оцінки того, як алгоритми поводяться за умов значного дефіциту інформації. Результати, представлені у таблицях 3.6 і 3.7, наочно демонструють зміну їхньої ефективності в залежності від рівня пропусків. Зокрема, можна побачити, як збільшення обсягу відсутніх даних поступово знижує якість роботи алгоритмів, особливо коли пропуски сягають критичних рівнів.

Отримані висновки можуть бути корисними не тільки для оцінки поточних алгоритмів, але й для розробки нових методів, здатних краще адаптуватися до неповних даних. Подальші дослідження в цій сфері можуть бути спрямовані на розробку стратегій роботи з пропусками, що включають удосконалені моделі імпутації та інтеграцію інформації із зовнішніх джерел. Це дозволить підвищити надійність алгоритмів навіть за умов значного

дефіциту даних, забезпечуючи високу точність та ефективність їхнього застосування.

Таблиця 3.6 – Результати дослідження зі 100 відсутніми значеннями на датасеті Wine UCI

Алгоритми	**Пропуск**	**Датасет Wine UCI**					
		PC	**CE**	**SC**	**SI**	**XB**	**DI**
Алгоритм нейро-фаззі мережі Кохонена	100	2,1059e-11	**-5,7912e-04**	**4,3587e+03**	1,4987e+06	1,5006e+06	**0,0336**
Fuzzy C-Means (FCM)		**0,7798**	0,3973	7,6412e-04	**7,3406e-06**	9,2453	0,0330
Густафсон-Кессель (GK)		0,5959	0,5854	7,1943	0,0404	**1,2627**	0,0904

Таблиця 3.7 – Результати дослідження зі 100 відсутніми значеннями на датасеті Iris UCI

Алгоритми	**Пропуск**	**Датасет Iris UCI**					
		PC	**CE**	**SC**	**SI**	**XB**	**DI**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Алгоритм нейро-фаззі мережі Кохонена	100	1,7429e-06	**-4,6617e-04**	0,1089	25,5	25,7608	**0,2277**

Продовження таблиці 3.7

1	2	3	4	5	6	7	8
Fuzzy C-Means (FCM)	100	0,8161	0,3536	0,0091	**9,2846e-05**	3,6533	0,0575
Густафсон-Кессель (GK)		**0,8454**	0,2750	**0,8994**	0,0060	**2,9395**	0,1392

Проведемо підсумки проведених досліджень для кожного з порівнюваних алгоритмів та по кожному з критеріїв якості.

Підсумки для алгоритму кластеризації з відсутніми значеннями. За коефіцієнтом розподілу алгоритм показує найнижчі значення серед інших методів, що свідчить про високий ступінь невизначеності в кластеризації. Ентропія класифікації також має найнижчі значення, що вказує на чітке розділення кластерів і дозволяє легко ідентифікувати об'єкти, які належать до конкретних груп.

Щодо індексу компактності і розділення, алгоритм демонструє найвищі значення у датасеті Wine UCI, тоді як у датасеті Iris UCI показники залишаються середніми порівняно з іншими алгоритмами. Це є позитивним результатом, адже високі значення цього індексу вказують на компактність кластерів, тобто об’єкти всередині кластерів розташовані близько один до одного, а також добре відокремлені від інших кластерів. Проте, за індексом розділення алгоритм не демонструє найкращих результатів у жодному з датасетів. Це свідчить про те, що кластери недостатньо добре відокремлені, а об'єкти з різних кластерів часто мають схожі характеристики.

Схожа ситуація спостерігається з індексом Хіе-Бені: його значення вказують на невеликі міжкластерні відстані, що свідчить про меншу якість розділення кластерів. Водночас, індекс Данна показує хороші результати, підтверджуючи високу якість кластеризації. Високі значення цього індексу

вказують на те, що кластери чітко розділені, а об’єкти всередині кластерів близькі за своїми характеристиками.

Таким чином, хоча алгоритм демонструє певні недоліки у розділенні кластерів за деякими критеріями, він все ж забезпечує хорошу якість кластеризації завдяки своїм сильним сторонам. Якісна кластеризація спостерігається по 3 критеріям оцінки. Це непоганий результат, найкраще алгоритм показав себе зі 100 відсутніми значеннями, тобто чим більше відсутніх значень, тим краще кластеризує алгоритм.

Підсумки для алгоритму Fuzzy C-Means (FCM). Коефіцієнт розподілу для цього алгоритму є вищим порівняно з алгоритмом, що працює з відсутніми даними, що свідчить про кращу чіткість у кластеризації, хоча цей показник не є найкращим серед усіх порівнюваних алгоритмів. Ентропія класифікації показує стабільні значення, незалежно від рівня відсутніх даних, що свідчить про стійкість алгоритму до викривлень. Однак, цей показник не досягає максимального рівня порівняно з іншими підходами.

Індекс компактності і розділення має низькі значення, що вказує на високу компактність кластерів, тобто об'єкти всередині кластерів розташовані близько один до одного. Водночас індекс розділення демонструє досить малі значення, що є ознакою доброго розділення між різними кластерами. Значення індексу Хіе-Бені також є доволі хорошими, що додатково підтверджує компактність кластерів і їхню чітку структурованість.

Щодо індексу Данна, алгоритм FCM показує непогані результати, які свідчать про задовільне розділення між кластерами. Проте, за цим показником інші алгоритми виявляються більш ефективними. Загалом, алгоритм Fuzzy C-Means залишається класичним підходом до кластеризації, що забезпечує стабільність навіть у випадках з викривленнями даних. Однак якість кластеризації знижується зі збільшенням кількості пропусків у даних, що є його слабкою стороною.

Незважаючи на недоліки, FCM залишається одним із ефективних методів для роботи із розмитими межами кластерів, забезпечуючи високу

адаптивність у багатьох прикладних задачах, та показує стабільну кластеризацію навіть за наявності викривлених даних, але якість кластеризації знижується зі збільшенням пропусків даних.

Підсумки для алгоритму Густафсона-Кесселя. Коефіцієнт розподілу для цього алгоритму зазвичай є вищим, ніж у Fuzzy C-Means, проте спостерігається зниження цього показника зі збільшенням кількості пропусків у даних. Ентропія класифікації залишається стабільною, що свідчить про високу адаптивність алгоритму до роботи з відсутніми значеннями.

Щодо індексу компактності і розділення, алгоритм демонструє високі показники для датасету Iris UCI, хоча для датасету Wine UCI ці значення не є найкращими. Це свідчить про те, що алгоритм забезпечує добру компактність кластерів, але його ефективність залежить від специфіки даних. Індекс розділення залишається на середньому рівні, що вказує на задовільну якість розділення між кластерами.

Значення індексу Хіе-Бені для цього алгоритму є найкращими серед усіх порівнюваних методів кластеризації, що підтверджує високий рівень компактності кластерів. Індекс Данна також демонструє хороші результати, особливо для випадків із меншою кількістю викривлених даних. Це свідчить про те, що алгоритм забезпечує високу якість кластеризації, зокрема з точки зору чіткого розділення між кластерами.

У висновку алгоритм Густафсона-Кесселя є значно ефективнішим ніж класичний алгоритм FCM, він зберігає високу якість кластерів за умов відсутніх значень. Це свідчить про його адаптивність і здатність формувати чіткі і компактні кластери в умовах невизначеності. У порівнянні з алгоритмом адаптивної кластеризації вони обидва показують хороші результати по різним показникам, тобто обидва алгоритма необхідно використовувати до різних задач кластеризації викривлених даних, звертаючи уваги на їх переваги та недоліки.

ВИСНОВКИ

У рамках кваліфікаційної роботи було проведено дослідження та моделювання нейро-фаззі мережі Кохонена для кластеризації викривлених даних, а також програмний порівняльний аналіз з двома класичними алгоритмами кластеризації Fuzzy C-Means та Густафсона-Кесселя.

Науковою новизною дослідження є розроблення ефективного методу кластеризації з викривленими даними який побудовано на основі нейро-фаззі мережі Кохонена.

За результатами дослідження алгоритмів кластеризації викривлених даних можна зробити наступні висновки. Змодельований алгоритм нейро-фаззі мережі Кохонена показав хороші результати щодо якості кластеризації при високій кількості відсутніх даних, що робить його ефективним у задачах такого типу. Перевагами використання глибоких згорткових мереж є точність результатів. Недоліком є тривалий час навчання моделі. Алгоритм Густафсона-Кесселя, який є розширенням алгоритму Fuzzy C-Means, також показав непогані результати, однак він є менш ефективним при значній кількості відсутніх даних.

Алгоритм Fuzzy C-Means виявився найменш ефективним в умах відсутніх даних, проте він найбільш стабільний. Даний метод виявився ефективним для кластеризації з маленькою кількістю відсутніх даних, зі збільшенням їх кількості ефективність методу знижується.

Подальші перспективи дослідження у сфері кластеризації викривлених даних можуть бути спрямовані на вдосконалення вже існуючих алгоритмів та розробку нових методів покращення кластеризації викривлених даних.

По-перше, варто провести дослідження в умовах ще більшої кількості викривлених даних, та з більшою кількістю алгоритмів кластеризації для виявлення найкращих алгоритмів для різних умов та кількості відсутніх даних.

По-друге, варто провести навчання інших нейронних мереж до великої точності результатів.

Результати дослідження апробовано у вигляді тез доповіді під час VI міжнародної студентської наукової конференції «Концепт науки XXI: стратегії, методи та наукові інструменти» [42].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Лобода, Ю. Г., Лобода, Ю. Г., & Лобода, Ю. Г. (2022). *Особливості підготовки даних з метою подальшого аналізу* (Doctoral dissertation, Одеса) С 32-34.

2. Yakovleva, O., Slyusar, V., Kushnir, O., & Sabovchyk, A. (2021). New trends in scientific and technological revolution (STR) and transformation of science and education systems in the paradigm of sustainable development. In E3S Web of Conferences (Vol. 277, p. 06006). EDP Sciences.

3. Яровенко, Г., & Кобзенко, В. (2022). Попередній аналіз і підготовка даних для прогнозування трендів кібератак. *Економіка та суспільство*, (45).

4. Lyashenko V., Kobylin O., Selevko O. (2020) Wavelet Analysis and Contrast Modification in the Study of Cell Structures Images. International Journal of Advanced Trends in Computer Science and Engineering. 9(4), pp. 4701-4706.

5. Гороховатський, В. О., & Творошенко, І. С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.

6. Ткаченко, О. М., Руденко, Н. В., Куфтеріна, С. Р., Лемешко, А. В., & Захаржевський, А. Г. (2020). Алгоритм визначення оптимальної кількості кластерів на базі нейронної мережі Кохонена. *Зв'язок*, (3), С. 8-11.

7. Tvoroshenko, I. S., & Bielinskyi, Y. (2021). On the features of methods of processing and recognition of handwritten text, (September 21 – 24, 2021, Boston, USA), pp. 410-415.

8. Руденко, Д., Лотвінова, В., & Безверха, Є. (2023). Навчання нейронної мережі в задачах обробки даних. *Collection of scientific papers «SCIENTIA»*, (September 22, 2023; Singapore, Singapore), pp. 94-95.

9. Лещинський, О. Л., & Іщенко, А. О. (2017). Використання нейромереж у процесі інтелектуального (кластерного) аналізу даних. *Економіка і суспільство*, (11), С. 578-581.

10. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the

structural image description, Advances in Electrical and Electronic Engineering, 21(1), pp. 19-27.

11. Одегов, М., & Бабіч, Ю. (2024). Принцип далекодії-близькодії у задачах струкруризації та навчання штучних нейронних мереж. *Herald of Khmelnytskyi National University. Technical sciences*, *337*(3 (2)), С. 357-365.

12. Bodyanskiy, Y., Shafronenko, A., & Pliss, I. (2021). Правдоподібна нечітка кластеризація даних на основі еволюційного методу божевільних котів. *System research and information technologies*, (3), С. 110-119.

13. Субботін, С. О., & Субботин, С. А. (2020). Нейронні мережі: теорія та практика: навч. посібник 184 с.

14. Москаленко, Ю. В. (2020). Середовище моделювання нейронних мереж для розв'язання задачі кластеризації. *Математичне та комп'ютерне моделювання. Серія: Технічні науки*, С. 79-87.

15. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень: навч. посібник. *Харків: ХНУРЕ,* 124 *с*.

16. Коляда, К. В., Марковський, О. П., Саверченко, В. Г., & Торошанко, А. І. (2020). Метод резервування та відновлення втрачених даних в глобальних мережах. *Телекомунікаційні та інформаційні технології*, (1), С. 4-14.

17. Коляда, К. В., Марковський, О. П., Саверченко, В. Г., & Торошанко, А. І. (2020). Метод резервування та відновлення втрачених даних в глобальних мережах. *Телекомунікаційні та інформаційні технології*, (1), С. 4-14.

18. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. *Системи управління, навігації та зв'язку. Збірник наукових праць*, *5*(57).

19. Koliada, K., Romankevich, V., Orlova, M., & Markovskyi, O. (2020). Метод відновлення даних при їх розподіленому зберіганні на віддалених сховищах. *Computer-integrated technologies: education, science, production*, (40), pp. 44-50.

20. Шафроненко, А. Ю., & Бодянський, Є. В. (2023). Нечітка достовірна кластеризація великих масивів даних з гіпереліпсоїдальними класами з

довільною орієнтацією осей. *Наука і техніка Повітряних Сил Збройних Сил України*, (1 (50)), 93-99.Захаров, Є. (2023, November).

21. Дичка, І. А., Сулема, Є. С., Дичка, И. А., & Сулема, Е. С. (2020). Модель подання мультимодальних даних для комплексного опису об'єктів спостереження. *Вісник Вінницького політехнічного інституту.№ 1:* С. *53-60.*

22. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.

23. Тихонов, Є. С., & Тихонова, К. В. (2020). Аналіз існуючих алгоритмів кластеризації даних. Переваги та недоліки. *Зв'язок*, (1), С. 17-20.

24. Шафроненко, А. Ю., Касаткіна, Н. В., & Бодянський, Є. В. Достовірна робастна онлайн нечітка кластеризація в задачах інтелектуального аналізу потоків даних.

25. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

26. Li, D., Gu, H., & Zhang, L. (2010). A fuzzy c-means clustering algorithm based on nearest-neighbor intervals for incomplete data. *Expert Systems with Applications*, *37*(10), 6942-6947.

27. Мартинюк, Т. Б., Круківський, Б. І., & М'якішев, О. А. (2022). Особливості моделей нейромережевого класифікатора для розпізнавання об'єктів. *Вісник Вінницького політехнічного інституту.№ 4: 56-63.*

28. Gorokhovatskyi, V.O., Tvoroshenko, I.S., and Peredrii O.O. (2020) Image classification method modification based on model of logic processing of bit description weights vector, *Telecommunications and Radio Engineering*, 79(1), pp. 59-69.

29. Ткаченко, В. А., Холод, Б. І., Ковальчук, К. Ф., & Вітлінський, В. В. (2023). Перспективи економіко-математичного моделювання в цифровій економіці. *Сімдесят треті економіко-правові дискусії. Серія: Соціальні та*

гуманітарні науки: матеріали Міжнародної мультидисциплінарної наукової інтернет-конференції,(м. Львів, Україна–м. Переворськ, Польща, 22-23 березня 2023 р.)[редкол.: О. Патряк та ін.]; ГО "Наукова спільнота"; WSSG w Przeworsku.–Львів: ФО-П Шпак ВБ–183 с.–ISSN 2522-963X., 39.

30. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

31. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, International Journal of Academic and Applied Research, 7(9), pp. 57-70.

32. Iris dataset uci. URL:https://www.kaggle.com/datasets/jillanisofttech/iris-dataset-uci (дата звернення 15.10.2024).

33. Uci wine datacet. URL: https://www.kaggle.com/datasets/aarontanjaya/uci-wine-dataset/data (дата звернення: 16.10.2024).

34. Bodyanskiy, Y. V., Pliss, I. P., Shafronenko, A. Y., & Kalynychenko, O. V. (2022). Нечітка довірча кластеризація даних на основі аналізу щільності розподілу даних та їх піків. *Radio Electronics, Computer Science, Control*, (3), 58-58.

35. Atkinson, A. C., Riani, M., & Corbellini, A. (2021). The box–cox transformation: Review and extensions, С. 239 - 255.

36. Chen, Y., Zhou, L., Bouguila, N., Wang, C., Chen, Y., & Du, J. (2021). BLOCK-DBSCAN: Fast clustering for large scale data. *Pattern Recognition*, *109*, 107624.

37. Bas, E., & Egrioglu, E. (2022). A fuzzy regression functions approach based on Gustafson-Kessel clustering algorithm. *Information Sciences*, *592*, 206-214.

38. Шафроненко, А. Ю., Свистунов, І. О., & Танянський, О. С. (2021, November). Адаптивна нечітка кластеризація даних на основі еволюційних

процедур. *The 5 th International scientific and practical conference – Topical issues of modern science, society and education (November 28-30, 2021) SPC – Sci-conf. com. ua, Kharkiv, Ukraine. 2021. 2101 p.* (p. 644).

39. A look at the performance of expression templates in C++: Eigen vs Blaze vs Fastor vs Armadillo vs XTensor. URL: https://romanpoya.medium.com/a-look-at-the-performance-of-expression-templates-in-c-eigen-vs-blaze-vs-fastor-vs-armadillo-vs-2474ed38d982 (дата звернення: 20.10.2024).

40. Gorokhovatskyi, V., & Tvoroshenko, I. (2020). Image Classification Based on the Kohonen Network and the Data Space Modification. In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020). 2608 p. (pp. 1013-1026).

41. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, International Journal of Academic and Applied Research, 7(11), pp. 134-145.

42. Личагіна С.М. (2024) Дослідження та моделювання нейро-фаззі мережі Кохонена для кластеризації викривлених даних. Молодіжна наукова ліга. VI Міжнародна студентська наукова конференція «Концепт науки XXI: стратегії, методи та наукові інструменти», Вересень 13, 2024, Харків, Україна, с. 71-72.