Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи інтелектуального аналізу великих даних
за допомогою машинного навчання

(тема)

Виконав:
здобувач 2 року навчання,
групи СПзм-23-1
Лідія ФАТІЙ
(власне ім'я, прізвище)

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

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

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: ас. Павло КРАВЧЕНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системне програмування
(повна назва)

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

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Фатій Лідії Сергіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Методи інтелектуального аналізу великих даних за допомогою машинного навчання

затверджена наказом по університету від "07" квітня 2025 р. № 53 Стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи
голосові команди
навколишні умови
Python
Google Colab

4. Перелік питань, що потрібно опрацювати у роботі
Інтелектуальний аналіз даних та методи машинного навчання
Апарат кластеризації даних з використанням карт Кохонена
Паралельні системи керування базами даних

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 18 слайдів

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	07.04.2025–29.04.2025	
2	Огляд існуючих моделей та методів	30.04.2025–10.05.2025	
3	Розробка методу	11.05.2025–20.05.2025	
4	Вибір програмних засобів	21.05.2025–29.05.2025	
5	Програмна реалізація	30.05.2025–02.06.2025	
6	Аналіз отриманих результатів	03.06.2025–05.06.2025	
7	Оформлення записки	06.06.2025–12.06.2025	

Дата видачі завдання " 07 " квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Павло КРАВЧЕНКО
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 59 с., 21 рис., 1 дод., 15 джерел.

ВЕЛИКІ ДАНІ, МАШИННЕ НАВЧАННЯ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, ПРОГНОЗНА АНАЛІТИКА, ШТУЧНИЙ ІНТЕЛЕКТ, КЛАСИФІКАЦІЯ, КЛАСТЕРИЗАЦІЯ, РЕГРЕСІЯ, НЕЙРОННІ МЕРЕЖІ, ГЛИБИННЕ НАВЧАННЯ, НАВЧАННЯ З УЧИТЕЛЕМ, НАВЧАННЯ БЕЗ УЧИТЕЛЯ, ВИБІР ОЗНАК, ЗМЕНШЕННЯ РОЗМІРНОСТІ, МАСШТАБОВАНІСТЬ, ПАРАЛЕЛЬНА ОБРОБКА, ПЕРЕДОБРОБКА ДАНИХ, ОЦІНЮВАННЯ МОДЕЛЕЙ.

Метою кваліфікаційної роботи є аналіз методів інтелектуального аналізу великих даних в паралельних системах керування базами даних.

У ході виконання кваліфікаційної роботи здійснено глибокий аналіз інтеграції методів інтелектуального аналізу даних у середовище реляційних систем управління базами даних. Розглянуто концептуальні засади та практичні підходи до створення паралельних алгоритмів, орієнтованих на виконання у кластерних обчислювальних середовищах, що базуються на сучасних багатоядерних прискорювачах. Особливу увагу приділено архітектурним рішенням і принципам побудови інтегрованої платформи, реалізованої на основі відкритої СКБД PostgreSQL з використанням апаратного забезпечення типу Intel Many Integrated Core.

ABSTRACT

Master’s thesis: 59 pages, 21 figures, 1 appendices, 15 sources.

BIG DATA, MACHINE LEARNING, DATA MINING, PREDICTIVE ANALYTICS, ARTIFICIAL INTELLIGENCE, CLASSIFICATION, CLUSTERING, REGRESSION, NEURAL NETWORKS, DEEP LEARNING, SUPERVISED LEARNING, UNSUPERVISED LEARNING, FEATURE SELECTION, DIMENSIONALITY REDUCTION, SCALABILITY, PARALLEL PROCESSING, DATA PREPROCESSING, MODEL EVALUATION.

The major goal of this thesis is to analyze methods of intelligent big data analysis within parallel database management systems.

In the course of the study, an in-depth study was conducted on the integration of data mining techniques into the environment of relational database management systems. The work examines both conceptual foundations and practical approaches to the development of parallel algorithms designed for execution in clustered computing environments that leverage modern many-core accelerators. Particular emphasis is placed on architectural solutions and implementation principles of an integrated platform based on the open-source DBMS PostgreSQL, utilizing Intel Many Integrated Core (MIC) hardware architecture.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

БЗ – база знань

БП – база прецедентів

ІАД – інтелектуальний аналіз даних

ІС – інтелектуальна система

ЛПР – людина, що приймає рішення

СКБД – система керування базами даних

ШНМ – штучна нейронна мережа

BI – Business Intelligence

CBR – Case-Based Reasoning

DM – Data Mining

KDD – Knowledge Discovery in Databases

ВСТУП

Сучасні інтелектуальні системи широко інтегрують методи інтелектуального аналізу даних, які стають невід'ємною складовою при розробці високорозвинених технологічних рішень. Вони знаходять застосування в різноманітних програмних середовищах, включаючи системи підтримки прийняття рішень, засоби керування базами даних і знань, корпоративні інформаційні системи, електронний документообіг, а також у технологіях машинного навчання. Функціональна цінність цих методів полягає у здатності автоматизовано витягувати релевантні закономірності з великих масивів інформації, що надає змогу істотно підвищити ефективність аналітичних процесів у прикладних сферах.

У межах інтелектуального аналізу даних застосовуються різні підходи, які базуються на статистичному і логічному моделюванні, а також на методах штучного інтелекту. До таких підходів належать, зокрема, алгоритмічні стратегії, що реалізують ідеї машинного навчання, еволюційного програмування, нейроподібного узагальнення, а також методи, орієнтовані на аналіз структурних схожостей. Особливу увагу в межах цього дослідження зосереджено на використанні методу правдоподібних міркувань на основі прецедентів, який демонструє високу ефективність при роботі з неоднорідними даними в умовах неповної інформації.

Інтелектуальний аналіз даних, як науково-практична дисципліна, формує підґрунтя для автоматизованого виявлення латентних взаємозв'язків, які можуть бути недоступними при традиційному аналізі. Його концептуальна база охоплює теоретичні і прикладні аспекти дослідження інформаційного простору з метою формування нових знань. Зокрема, така аналітика охоплює процеси пошуку, ідентифікації, класифікації та узагальнення даних у складних інформаційних середовищах, а також підтримує створення та адаптацію баз знань, що є фундаментом для

побудови навчальних, експертних та перекладних систем.

Дослідження, представлене в цій кваліфікаційній роботі, зосереджене на аналізі можливостей інтеграції методів інтелектуального аналізу даних у паралельні системи управління базами даних. Основною метою є вивчення ефективних технологічних підходів до обробки даних у паралельному середовищі за допомогою методів ІАД, з урахуванням специфіки архітектури реляційних баз. Об’єктом аналізу виступають саме методи інтелектуального аналізу в контексті функціонування реляційних СКБД, що дозволяє поєднати засоби аналітик даних із інструментами високопродуктивної обробки.

У рамках дослідження передбачено обґрунтування вибору алгоритмічних стратегій для реалізації ІАД у паралельному середовищі, проведення порівняльного аналізу відповідних рішень з позицій продуктивності та масштабованості, а також розроблення архітектурної моделі системи, що дозволяє впровадити модифікований підхід до аналізу даних у рамках паралельної реляційної СКБД.

Метою кваліфікційної роботи є дослідження методів інтелектуального аналізу великих даних в паралельних системах керування базами даних.

Об’єктом дослідження є методи ІАД в реляційних СКБД.

Завдання:

- розглянути методи та алгоритми для впровадження інтелектуального аналізу великих даних в паралельну СКБД;
- проаналізувати паралельні алгоритми рішення завдань кластеризації великих даних засобами паралельної реляційної СКБД;
- розробити архітектуру паралельної СКБД з використанням модифікованого методу.

1 ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ ТА МЕТОДИ МАШИННОГО НАВЧАННЯ

У цьому розділі висвітлюються ключові теоретичні засади машинного навчання, які становлять фундамент для реалізації практичної частини дослідження. Розглядається загальна характеристика цієї галузі знань, а також методологічні підходи, що мають безпосереднє відношення до поставленої наукової проблематики. Зокрема, акцент зроблено на тих алгоритмах, які забезпечують ефективне виявлення закономірностей у даних і є релевантними для виконання завдань інтелектуального аналізу.

1.1 Машинне навчання

Становлення машинного навчання як самостійного напряму відбулося внаслідок інтенсивного розвитку технологій обробки великих обсягів інформації. Цей напрям сьогодні трактується як частина штучного інтелекту, основним принципом якого є здатність системи адаптивно змінювати свою поведінку на основі накопиченого досвіду. Концепція навчання без явного програмування передбачає побудову алгоритмів, які з часом вдосконалюються за рахунок аналізу власної діяльності. У цьому контексті варто навести одне з класичних визначень машинного навчання, запропоноване Т. Мітчеллом, згідно з яким програма набуває навичок у межах певного класу завдань, якщо її результативність щодо цих завдань зростає пропорційно кількості накопиченого досвіду, що оцінюється визначеним критерієм ефективності.

Механізм машинного навчання передбачає створення моделі, яка формалізує знання, отримані з навчальних даних, і застосовується для прогнозування результатів у нових випадках. Кінцевий результат такого прогнозування визначається як вихід моделі, сформованої відповідно до

обраного алгоритму та особливостей початкових вхідних даних. Прикладом типового практичного застосування є оцінка вартості нерухомості за допомогою таких параметрів, як площа чи кількість кімнат. У сфері медичної діагностики модель може бути використана для класифікації зображень, наприклад, диференціації між здоровими тканинами та зображеннями, що свідчать про наявність патології. Інший варіант застосування передбачає виконання кластеризації – процесу автоматичного групування об'єктів на основі спільних характеристик, наприклад, розподілу зображень тварин за подібністю.

Для ґрунтовного розуміння принципів функціонування таких моделей доцільно звернутися до узагальненої схеми робочого процесу машинного навчання, яка представлена на рисунку 1.1. Ця схема демонструє основні етапи побудови моделі – від обробки вхідних даних до формування прогнозів – і слугує методологічною основою для розробки прикладних систем інтелектуального аналізу даних.

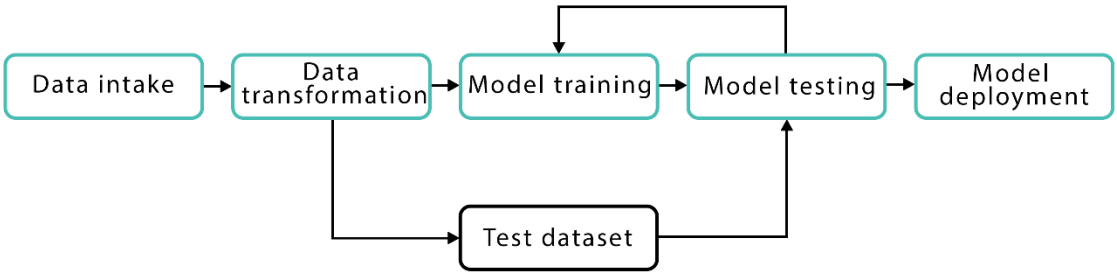

Рисунок 1.1 – Загальний процес роботи

Типовий процес розробки моделей машинного навчання включає послідовну реалізацію низки етапів, кожен із яких виконує критичну функцію у формуванні ефективної системи аналізу даних. Початковою фазою є завантаження та зберігання даних у пам'яті комп'ютера, що створює основу для подальших перетворень. На наступному етапі здійснюється трансформація інформації: дані очищуються від шумів, нормалізуються відповідно до обраного масштабу і структуруються таким чином, щоб бути

сумісними з вимогами алгоритмів навчання. У цьому ж контексті здійснюється виокремлення релевантних ознак, що мають значення для формування прогнозної моделі, та виконується поділ на навчальний і тестовий підмножини.

Фаза навчання моделі полягає у побудові алгоритмічної структури, що оперує на основі вхідних прикладів для формалізації закономірностей у даних. Після цього модель проходить перевірку її ефективності через тестування на окремому наборі даних, не використаному в процесі навчання. Результати цього етапу дозволяють здійснити оцінку якості побудованої моделі, а також за потреби внести зміни або сформувати нову модель із урахуванням отриманого досвіду. Завершальним етапом є інтеграція моделі в цільове середовище, тобто її розгортання, яке відбувається після досягнення задовільного рівня точності або по завершенні заздалегідь визначеної кількості ітерацій. У процесі розгортання обирається найбільш оптимальне рішення з-поміж побудованих варіантів, що забезпечує найкращу продуктивність для реального використання.

1.1.1 Виділення ознак

Для забезпечення ефективного функціонування алгоритмів машинного навчання критично важливою є здатність правильно інтерпретувати та структурувати вхідні дані, що передбачає виділення інформативних характеристик, які містять релевантну інформацію для вирішення задачі. У будь-якому практичному випадку, незалежно від специфіки задачі, необхідно здійснити попередню обробку даних таким чином, щоб вони набули уніфікованої форми, придатної для подальшого аналізу. Наприклад, у задачі прогнозування вартості нерухомості початкові дані можуть бути представлені у вигляді матриці, де кожен стовпець відповідає певному атрибуту об’єкта, а кожен рядок – конкретному екземпляру, тобто числовим значенням цих атрибутів. У випадку обробки візуальної інформації вхідними

параметрами можуть бути значення інтенсивностей пікселів, зазвичай подані у форматі RGB.

Такі характеристики, що виражають суттєві властивості об'єктів, прийнято називати ознаками, а сукупність цих параметрів, структуровану у формі числового подання, розглядають як вектор ознак. Формування такого вектора здійснюється шляхом процедури, відомої як виділення ознак, яка має на меті створити набір значущих і несуперечливих змінних. Якість цього етапу значною мірою визначає точність побудованої моделі, адже відображення неінформативних або надлишкових характеристик призводить до зниження її ефективності. Враховуючи специфіку предметної області, процес виділення ознак часто є складним і потребує глибокого емпіричного опрацювання.

Важливим аспектом побудови адекватної моделі є також унеможливлення надмірної кореляції між ознаками. Присутність дублікатів або змінних, що відображають подібну інформацію, може викликати ефект переобучення, у результаті чого модель втрачає здатність до генералізації й демонструє упереджене прогнозування. Коли розмірність початкового векторного простору є надмірною, застосовуються методи скорочення кількості ознак, що відомо як процедура вибору ознак. Її мета полягає в тому, щоб зберегти найбільш репрезентативні змінні, зменшивши обсяг вхідних даних без суттєвої втрати точності, що сприяє стабільності моделі та скороченню часу її навчання.

Крім виділення і вибору ознак, ефективність моделі залежить також від приведення даних до єдиного масштабу. З огляду на те, що в одному наборі можуть одночасно бути присутніми ознаки з абсолютно різними одиницями виміру, наприклад, кількість кімнат і площа у квадратних метрах, такі показники потребують узгодження. У протилежному разі арифметичні операції над ними можуть давати некоректні результати. З цією метою застосовуються процедури нормалізації та стандартизації, що дозволяють вирівняти масштаб усіх вхідних значень і забезпечити коректну взаємодію

між компонентами векторного простору.

У процесі підготовки вхідних даних для навчання моделей машинного навчання важливо враховувати не лише приведення ознак до уніфікованого масштабу за допомогою середнього значення та стандартного відхилення, а й можливість розширення простору ознак у разі, коли вихідна інформація є недостатньо репрезентативною. Хоча в багатьох випадках завдання зниження розмірності є актуальним для оптимізації обчислювальних ресурсів і підвищення узагальнювальної здатності моделі, іноді, навпаки, доцільно здійснити розширення ознакового простору. Такий підхід набуває особливої значущості в задачах підвищеної складності, коли лінійні взаємозв'язки між змінними не дозволяють досягти належного рівня точності.

Збільшення розмірності в цьому контексті може бути виправданим, оскільки воно дає змогу моделі охопити більш складні залежності другого і вищих порядків, які не були враховані в початковому представленні даних. Подібне розширення, зазвичай, реалізується шляхом побудови нових ознак, що є функціональними перетвореннями існуючих, і лежить в основі багатьох сучасних алгоритмів, зокрема методів, що базуються на гіпотезі перетворення простору. Яскравим прикладом є метод опорних векторів, де застосування ядрових функцій дозволяє ефективно здійснювати поділ у вищому, нелінійно трансформованому просторі, недосяжному для звичайного лінійного розв'язання.

1.1.2 Навчання з вчителем та без вчителя

До цього моменту розгляд машинного навчання відбувався в контексті наявності маркованих вхідних даних, які безпосередньо використовуються для побудови моделей. Однак така ситуація не завжди має місце у практичних умовах, що зумовлює необхідність звернення до фундаментального поділу підходів у машинному навчанні – на навчання з учителем і навчання без учителя. Ці два підходи різняться як за вихідними

передумовами, так і за методами досягнення результатів.

У випадку навчання з учителем система має доступ до набору даних, кожен елемент якого супроводжується відповідним очікуваним результатом. Завдяки такому маркуванню модель отримує змогу узагальнювати наявні залежності між вхідними параметрами та відомими відповідями, що дозволяє згодом ефективно передбачати результати для нових прикладів. Типовим прикладом керованого навчання є задача передбачення вартості нерухомості, де для кожного об’єкта відомі як його характеристики, так і фактична ціна. Залежно від типу вихідної змінної, задачі в цьому підході можуть набувати форми регресії або класифікації. Якщо результатом є числове значення, що змінюється в континуальному просторі, то модель виконує регресійне прогнозування. У випадках, коли необхідно визначити, до якого з обмеженого набору класів належить конкретний приклад, маємо справу з класифікацією, яка забезпечує віднесення нових даних до певної категорії.

На противагу цьому, у навчанні без учителя система не має доступу до попередньо визначених відповідей або класів. Метою такого підходу є виявлення прихованої структури у невпорядкованих і немаркованих даних. Тут модель формує уявлення про природні групи або закономірності, притаманні досліджуваній вибірці, без наявності зовнішніх вказівок. Одним із найбільш поширених типів такого підходу є кластеризація, яка полягає в автоматичному поділі даних на групи за критерієм подібності між об'єктами. Ці групи, або кластери, формуються на основі спільних ознак без попереднього визначення їх кількості чи складу, що дозволяє застосовувати метод у ситуаціях, коли відсутня інформація про бажані результати або їх неможливо отримати.

1.2 Методи класифікації

У контексті машинного навчання завдання виявлення шкідливого програмного забезпечення може бути інтерпретоване як задача класифікації

або кластеризації залежно від характеру даних та поставленої цілі. У випадку роботи з невідомими типами зловмисного коду доцільно застосовувати кластерний підхід, що дозволяє згрупувати об'єкти за схожими характеристиками, які автоматично виявляються алгоритмом. Водночас, якщо доступний обширний набір прикладів, що містить як шкідливі, так і нешкідливі файли з відповідними мітками, проблема зводиться до класифікації. У ситуації, коли розглядається вже відоме сімейство шкідливих програм, використання класифікаційної моделі є особливо ефективним, оскільки дозволяє ідентифікувати конкретний клас з високою точністю за наявності обмеженого набору можливих варіантів. Таким чином, у разі добре структурованих вхідних даних, задача визначення типу шкідливого об’єкта може бути вирішена більш успішно за допомогою класифікаційних алгоритмів, а не методів кластеризації.

У цьому підрозділі розглядаються теоретичні основи алгоритмів машинного навчання, які застосовано в рамках дослідження. Зокрема, першочергово аналізується один із базових, але водночас ефективних методів – алгоритм k-найближчих сусідів.

Метод k-найближчих сусідів (KNN) належить до класу простих, проте надійних підходів, що активно використовуються для розв’язання задач як класифікації, так і регресії. Його принциповою особливістю є непараметричний характер, тобто відсутність необхідності уявлень про форму розподілу вхідних даних. Такий підхід особливо корисний у практичних умовах, де структуру даних не завжди можна описати за допомогою узагальнених теоретичних моделей. Модель у класичному розумінні при цьому не формується: увесь обчислювальний процес базується на використанні збереженого навчального набору, а прогнозування відбувається під час обробки нового запиту.

Механізм роботи методу полягає у виявленні найближчих прикладів до нового екземпляра вхідних даних і прийнятті рішення на основі значень, що відповідають цим найближчим сусідам. У випадку класифікації результат

визначається більшістю голосів серед k найближчих об’єктів, тобто належністю нового об’єкта до того класу, який найчастіше зустрічається серед вибраних сусідів. Якщо йдеться про регресійне прогнозування, то значення вихідної змінної обчислюється як середнє значення цієї змінної серед сусідніх прикладів. Такий підхід дозволяє забезпечити адаптивну гнучкість моделі в умовах складних і нерегулярних розподілів даних, зберігаючи при цьому достатню точність прогнозування.

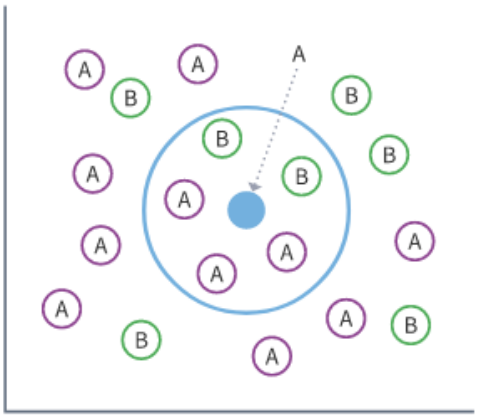

Рисунок 1.2 – Приклад KNN

У процесі реалізації методу найближчих сусідів ключову роль відіграє обчислення ступеня схожості між об'єктами, яке визначається через обчислення відстані між векторними поданнями прикладів. Саме від вибору відповідної метрики залежить ефективність алгоритму у контексті конкретної задачі, адже різні відстані по-різному реагують на розподіл і масштаб ознак. Серед поширених варіантів метрик, які знаходять застосування у цьому контексті, варто відзначити відстань Геммінга, що особливо доречна при роботі з бінарними векторами, манхеттенську відстань, яка характеризується підсумовуванням абсолютних різниць між координатами, а також більш загальну формалізацію – відстань Мінковського, яка охоплює обидва попередні випадки як часткові випадки для певних значень параметра степеня. Вибір тієї чи іншої метрики повинен здійснюватися з урахуванням природи вхідних даних і цільових характеристик моделі, з огляду на бажану

чутливість до відхилень або впливу окремих змінних.

$$\text{відстань Геммінга: } d_{ij} = \sum_{k=1}^{p} |x_{ik} - x_{jk}| \quad (1.1)$$

$$\text{Манхеттенська метріка: } d_1(p,q) = ||p-q||_1 = \sum_{i=1}^{n} |p_i - q_i| \quad (1.2)$$

$$\text{відстань Мінковського } = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} \quad (1.3)$$

$$\text{Евклідова відстань} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad (1.4)$$

У застосуванні методу k-найближчих сусідів вибір метрики для обчислення відстані між об’єктами суттєво впливає на точність класифікації, особливо коли ознаки мають різну природу. Евклідова відстань демонструє високу ефективність у тих випадках, коли всі ознаки мають однорідний тип та вимірювані в однакових одиницях. У ситуаціях, коли дані складаються з різнорідних атрибутів, доцільно застосовувати альтернативні підходи, зокрема манхеттенську метрику, яка є менш чутливою до різниці у масштабах і типах ознак.

Оцінка належності до певного класу в задачах класифікації може реалізовуватися не лише у вигляді остаточного рішення, а й через ймовірнісний підхід. У бінарних задачах ймовірність приналежності об'єкта до певного класу визначається співвідношенням кількості сусідів, що належать до цього класу, до загального числа обраних найближчих елементів. Зокрема, ймовірність віднесення до класу з нульовим індексом визначається як частка відповідних представників серед k найближчих сусідів, що дозволяє сформувати прогноз з урахуванням ступеня впевненості моделі.

Одним із критично важливих параметрів у цьому методі є значення k, від якого безпосередньо залежить якість класифікації. Обрання занадто малого значення може спричинити високу варіативність моделі, зокрема в умовах значного рівня шуму у вхідних даних, коли окремі приклади можуть непропорційно впливати на результат. З іншого боку, надто велике значення цього параметра призводить до згладжування меж між класами, зменшуючи чутливість моделі до локальних особливостей даних і, відповідно, погіршуючи точність прогнозування. Переобучення також можливе в разі надмірного узагальнення, коли враховуються елементи, що насправді не мають суттєвого зв’язку з об’єктом, який класифікується.

У зв’язку з цим пошук оптимального значення k є нетривіальним завданням, що потребує емпіричної перевірки або використання узагальнених емпіричних формул, які враховують обсяг навчальної вибірки. Такий підхід дозволяє балансувати між локальною точністю й глобальною стабільністю класифікаційної моделі.

$$k = \sqrt{n} \tag{1.5}$$

У задачах класифікації з рівномірним розподілом кількості класів доцільним є використання непарного значення параметра k, що дозволяє уникнути ситуацій із рівною кількістю голосів при визначенні найбільш ймовірного класу. Це рішення забезпечує однозначність у процесі прийняття рішень і сприяє підвищенню стійкості алгоритму до неоднозначних результатів.

Попри простоту реалізації та універсальність застосування, алгоритм k-найближчих сусідів має низку обмежень, зокрема чутливість до нерівномірного розподілу об’єктів у вибірці. У випадках, коли один із класів суттєво переважає за кількістю представників, модель демонструє схильність до упередженості на користь цього класу. Це відбувається тому, що домінуючий клас формує більшу частину локального простору навколо

нових об’єктів, і, як наслідок, навіть екземпляри, які природно належали б до меншого класу, з великою ймовірністю будуть класифіковані помилково. Така особливість алгоритму істотно впливає на точність у ситуаціях із нерівноваженими наборами даних, де потрібне попереднє балансування або застосування модифікованих підходів до зважування сусідів.

1.3 Метод опорних векторів

Метод опорних векторів (SVM) є одним із ключових алгоритмів машинного навчання, який знаходить широке застосування в задачах класифікації. Його принципова ідея полягає у побудові розділяючої гіперплощини, яка дозволяє найбільш ефективно розмежовувати об'єкти, що належать до різних класів. На відміну від деяких інших методів, SVM спрямований не просто на поділ даних, а на пошук такого просторового розмежування, яке забезпечить максимальну відстань між найближчими точками кожного класу й гіперплощиною. Ці критичні точки, що визначають положення гіперплощини, називаються опорними векторами. Вони мають особливе значення для моделі, оскільки їх вилучення призведе до зміщення межі класифікації.

Відстань від опорних векторів до гіперплощини визначається як проміжок (margin), і саме його максимізація є основним критерієм оптимальності в методі SVM. Ідея полягає в тому, що чим ширший цей проміжок, тим більш стабільною є модель до варіацій у нових даних, а отже – тим вищою є її узагальнювальна здатність. Хоча в межах простору ознак може існувати безліч можливих гіперплощин, які здатні розділити дані, алгоритм SVM формулює задачу так, щоб обрана гіперплощина забезпечувала найбільше можливе віддалення від обох класів, що і є запорукою високої точності прогнозування. Цей геометричний підхід дозволяє SVM демонструвати ефективність навіть у задачах з високою розмірністю ознак та складними розподілами.

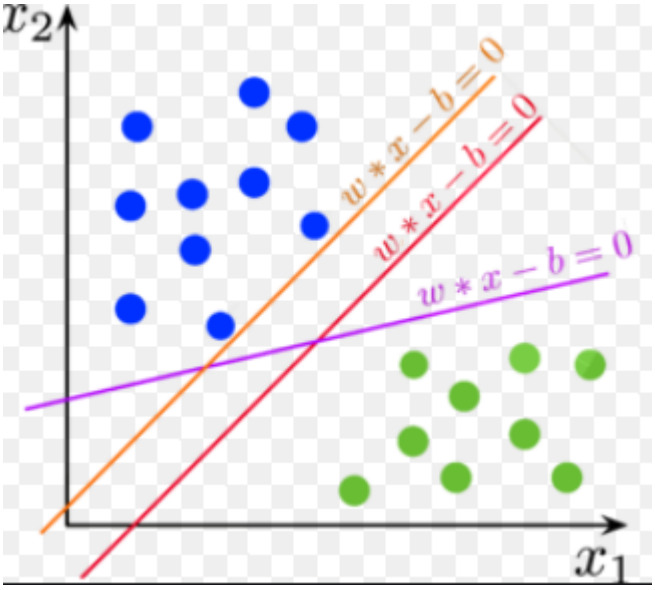

Рисунок 1.3 – Приклад методу опорних векторів

Для такого варіанту з набором даних час навчання може бути високим. [1].

2 АПАРАТ КЛАСТЕРИЗАЦІЇ ДАНИХ З ВИКОРИСТАННЯМ КАРТ КОХОНЕНА

Нейронні мережі Кохонена становлять специфічний клас штучних нейронних мереж, що орієнтовані на виконання завдань класифікаційного характеру, а також на вирішення пов’язаних із ними задач, таких як моделювання й прогнозування. Концепція цих мереж була розроблена фінським дослідником Тейво Кохоненом і заснована на принципі відображення вхідного багатовимірного простору у простір з нижчою розмірністю, зазвичай – двовимірний. Такий підхід забезпечує збереження топологічних властивостей даних, дозволяючи виявити їхню внутрішню структуру та закономірності.

Архітектурною основою мережі є так званий шар Кохонена, який складається з набору адаптивних обчислювальних елементів, що функціонують за принципом лінійних формальних нейронів. Особливість навчального процесу в цій мережі полягає у відсутності порівняння отриманих вихідних значень із заздалегідь відомими еталонами, що відрізняє її від класичних моделей із навчанням з учителем. Вхідні дані подаються до мережі без інформації про їхню належність до певних класів, і завдяки цьому відбувається самоорганізація структури – система самостійно формує уявлення про внутрішні взаємозв’язки між прикладами у вибірці. Таким чином, мережа Кохонена виконує функцію проєкційного узагальнення, виявляючи подібності між зразками та групуючи їх відповідно до виявлених закономірностей у вхідному просторі.

2.1 Структура мережі Кохонена

Мережа Кохонена є особливим різновидом штучної нейронної мережі, спеціалізованої на виконанні задач кластеризації. Структурно вона

складається лише з двох шарів – вхідного та вихідного, які функціонують відповідно до принципу повнозв'язності між шарами, при цьому внутрішньошарові з'єднання відсутні. Вихідний шар, який відіграє ключову роль у самоорганізації даних, зазвичай називають шаром Кохонена. Його функція полягає в інтерпретації та аналізі вхідних сигналів з метою виявлення топологічної близькості між об'єктами, що подаються до мережі.

На вхідний шар надходять вектори ознак, які репрезентують характеристики об'єктів, що підлягають кластеризації. Незважаючи на присутність вхідного шару, його нейрони не беруть участі в обробці чи адаптації моделі: вони виконують роль механізму передавання інформації до вихідного рівня. Кількість нейронів у вхідному шарі відповідає кількості ознак, що характеризують об'єкт, тобто розмірності вектора ознак. Саме у вихідному шарі відбувається процес самоорганізації, де мережа за допомогою навчального алгоритму формує топологічно впорядковану карту, яка відображає просторову структуру вхідних даних. Такий підхід дозволяє ефективно знижувати розмірність і водночас зберігати найсуттєвіші відношення між об'єктами у вихідному просторі.

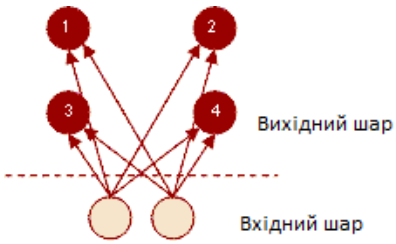

Рисунок 2.1 – Вхідний і вихідний шари нейронної мережі Кохонена

У мережі Кохонена кількість вихідних нейронів безпосередньо відповідає кількості кластерів, які необхідно сформувати в результаті кластеризації. Кожен із цих нейронів має асоціацію з певною групою об'єктів і виконує функцію ідентифікації належності вхідного зразка до відповідного кластера. Принцип функціонування вихідного шару ґрунтується на механізмі «переможець отримує все», згідно з яким лише один нейрон – той, який

виявляє найвищий рівень активації у відповідь на вхідні дані – залишається активним. Відповідно, його вихід встановлюється на одиницю, тоді як усі інші нейрони деактивуються, формуючи вихідні значення, рівні нулю.

Такий механізм дозволяє однозначно визначити кластер, до якого слід віднести новий об’єкт, адже кожен активний нейрон представляє конкретну зону у просторі ознак. Унаслідок обробки вхідного сигналу мережа здійснює вибір нейрона-переможця, що відповідає найбільш подібному кластеру, після чого здійснюється прив'язка об'єкта до цієї групи. Навчання такої мережі відбувається через модифікацію вагових коефіцієнтів між нейронами вхідного та вихідного шарів, однак на відміну від класичних нейронних мереж, де використовуються алгоритми з учителем, у цьому випадку застосовується конкурентне навчання. Його суть полягає в тому, що лише переможець і, за потреби, його сусіди змінюють свої ваги для кращого наближення до вхідного вектора, що забезпечує поступову адаптацію карти до внутрішньої структури даних.

Загальна архітектура мережі Кохонена може бути представлена у вигляді двовимірної прямокутної решітки з ММ нейронами, кожен з яких займає певне положення у топологічному просторі вихідного шару, що дозволяє візуалізувати результати кластеризації. Така структура детально ілюструється на рисунку 2.2.

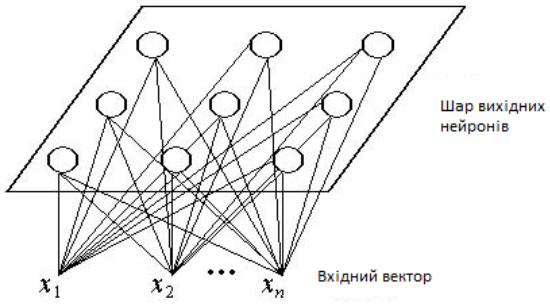

Рисунок 2.2 – Модель мережі Кохонена

У структурі мережі Кохонена кожен нейрон розміщується в межах двовимірної площини, яка утворює топологічну карту вихідного шару. До

кожного такого елемента надходить багатовимірний вхідний сигнал через синаптичні з’єднання, що забезпечують передачу збудження. Кожен нейрон має власне фіксоване положення у структурі шару та характеризується відповідним вектором ваг, який визначає його реакцію на поданий вхід. Просторова організація нейронів описується певною метрикою, яка визначає ступінь взаємного впливу між ними. Під час навчання найближчі до збудженого нейрона елементи реагують на сигнал активніше, тоді як з віддаленням рівень збудження зменшується.

Фізіологічна модель передбачає, що кожен нейрон генерує зважену відповідь на основі сумування вхідних сигналів, яка може бути як збудливою, так і гальмівною – залежно від характеру синаптичних зв’язків. Взаємодія між нейронами зумовлює ефект просторового поширення активності, при якому імпульс, спричинений активацією одного нейрона, породжує дифузний вплив на сусідні елементи, що поступово згасає із зростанням відстані. Така поведінка дозволяє мережі формувати центр збудження, який просторово співвідноситься з найбільш активованим нейроном. У свою чергу, зміна вхідного вектора призводить до збудження іншого нейрона, що змінює просторову відповідь усієї карти та дозволяє відстежити реакцію системи на нові стимули. Цей механізм становить основу самоорганізованого групування даних, яке реалізується мережею Кохонена.

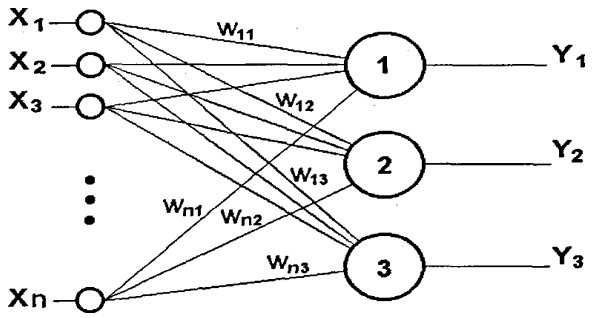

Рисунок 2.3 – Приклад нейронної мережі Кохонена

На рисунку 2.3 подано приклад базової архітектури нейронної мережі Кохонена, яка включає nn вхідних нейронів та три вихідні елементи, що

відповідають трьом можливим класам. Така структура використовується для класифікаційних задач, де кожен вихідний нейрон асоційований із певною категорією. Кожен вхідний нейрон передає значення відповідної ознаки об’єкта, що аналізується, до всіх нейронів вихідного шару. У результаті мережа здійснює віднесення об’єкта до одного з трьох кластерів шляхом виявлення найбільш активного нейрона, що демонструє найкращу відповідність вхідному вектору.

2.2 Навчання мережі Кохонена

Процес навчання нейронної мережі Кохонена базується на принципі адаптації вагових коефіцієнтів таким чином, щоб вхідні вектори зі схожими характеристиками активували один і той самий нейрон вихідного шару. Цей підхід реалізується без використання контрольованого навчання, що передбачає відсутність еталонних вихідних значень. Основним завданням такого самонавчання є забезпечення здатності мережі відокремлювати несхожі приклади, формуючи при цьому просторову структуру, в якій подібні об’єкти групуються разом.

На відміну від багатошарових нейронних мереж, модель Кохонена має спрощену архітектуру, яка складається лише з вхідного та вихідного шарів. Вихідні елементи розміщені в регулярній топології, найчастіше у вигляді двовимірної решітки, що дозволяє легко візуалізувати результат кластеризації. Навчання відбувається ітераційно, методом послідовного наближення, де мережа адаптується до внутрішніх структурних закономірностей у вхідних даних.

Початкове значення вагових коефіцієнтів задається випадковим чином. Далі, під час подачі кожного нового вхідного прикладу, обчислюється скалярний добуток між цим вектором та векторами ваг усіх нейронів вихідного шару. Нейрон, для якого цей добуток є найбільшим, визнається переможцем. Саме він і визначає напрям адаптації: у нього, а також у його

найближчих топологічних сусідів, коригуються вагові коефіцієнти в напрямі поданого вхідного вектора. Завдяки цьому відбувається поступове зближення вагових векторів нейронів до груп подібних об’єктів.

Навчальний алгоритм послідовно проходить через серію епох, кожна з яких охоплює обробку певного набору навчальних прикладів. Поступово мережа виявляє внутрішню структуру даних, і після достатньої кількості ітерацій формується топологічно впорядкована карта. Нейрони, що розташовані близько один до одного, стають чутливими до схожих вхідних векторів. З плином часу радіус впливу нейрона-переможця, тобто його топологічна околиця, звужується, забезпечуючи деталізацію кластеризації.

У результаті навчання мережа формує карту, на якій об’єкти, що мають подібні характеристики, відображаються просторово згрупованими, утворюючи області, що відповідають окремим кластерам. Таким чином, нейронна мережа Кохонена забезпечує ефективне узагальнення та візуалізацію складної багатовимірної інформації в зручному для інтерпретації вигляді.

2.3 Побудова карти Кохонена

Самоорганізовані карти Кохонена належать до класу конкурентних нейронних мереж із навчанням без учителя й використовуються для розв’язання задач кластеризації, візуалізації та попередньої інтерпретації даних. Цей тип мережі формує топологічну карту, яка забезпечує збереження просторових відношень між вхідними векторами у зниженому вимірі, що особливо цінне для аналізу складних багатовимірних структур.

Побудова карти Кохонена передбачає низку послідовних кроків, зокрема вибір топології нейронної решітки (прямокутна чи гексагональна), визначення розмірності карти, спосіб ініціалізації вагових коефіцієнтів (випадковий, проєкційний на площину головних компонент тощо), а також вибір метрики відстані між вхідним вектором і нейронами. Значну роль

відіграє алгоритм навчання та механізм зупинки адаптації, який визначає, коли вважається досягнутим стабільний стан мережі. Аналіз залишкової дисперсії та топологічної помилки допомагає оцінити якість кластеризації й підтвердити адекватність обраної архітектури.

Додатково необхідно оцінити розмальовку й поведінку карти при незначних змінах у вихідних даних, що дозволяє виявити чутливі ділянки з підвищеною ймовірністю спотворення. Проєкція даних на карту може виконуватись різними способами: у вузли, через апроксимації або за допомогою геометричних побудов на основі локальної топології карти.

Реалізація мережі Кохонена пов’язана з низкою потенційних труднощів, які впливають на ефективність навчання. Зокрема, правильний вибір коефіцієнта навчання є вирішальним – надто велике його значення забезпечує швидке зближення, проте знижує стійкість результату, тоді як надто мале уповільнює процес. Оптимальним підходом є поступове зменшення коефіцієнта з часом. У разі вимоги безперервної адаптації коефіцієнт слід зберігати сталим.

Ще одним критичним аспектом є початкова рандомізація ваг. Через нерівномірність розподілу вхідних даних, випадково ініціалізовані вагові вектори можуть розташовуватись занадто далеко від корисних ділянок простору, що призводить до їхньої неактивації та зниження ефективності класифікації. Неправильний вибір початкових ваг також унеможливлює активацію окремих нейронів, що обмежує участь у навчанні. Надто вузьке або передчасне зменшення параметра відстані призводить до втрати взаємодії між віддаленими нейронами, що порушує топологічну цілісність карти.

Кількість нейронів у шарі має відповідати складності класифікаційного завдання: надто мале число призводить до злиття кластерів, тоді як надлишкове – до надмірної фрагментації. Водночас топологічні обмеження моделі Кохонена передбачають, що лише кластери з опуклою структурою можуть бути адекватно представлені мережею.

Сфера застосування карт Кохонена охоплює широке коло аналітичних задач. Вони застосовуються для візуального моделювання, попереднього виявлення прихованих залежностей у великих обсягах даних, редукції розмірності, прогнозування та побудови класифікаторів. Карта дозволяє як виявити природні кластери в даних, так і оцінити їх близькість один до одного, що особливо корисно при проведенні розвідувального аналізу. У випадку, коли після навчання дані було класифіковано, можлива побудова повноцінної системи класифікації, здатної призначати нові об’єкти до визначених класів.

Мережа також здатна виявляти нові, раніше не спостережувані явища: якщо новий вхідний сигнал не знаходить відповідності серед існуючих кластерів, це вказує на наявність унікального патерну, що не був присутній у навчальній вибірці. Такий функціонал дозволяє використовувати карти Кохонена для виявлення аномалій, нових тенденцій або структур у даних, підвищуючи аналітичну глибину дослідження.

2.4 Висновки до розділу

Мережа Кохонена є прикладом системи з самонавчанням, що реалізує принципи самоорганізації в умовах відсутності еталонних результатів. Така архітектура призначена для автоматичної класифікації даних, де немає необхідності у заздалегідь позначеній навчальній вибірці. Завдяки цьому мережа здатна самостійно виявляти структури у вхідному просторі та здійснювати поділ об’єктів на класи на основі внутрішніх взаємозв’язків між векторами ознак.

Структурно мережа складається з двох основних шарів – вхідного та шару Кохонена. Вхідний шар виконує роль передавача сигналів, що відповідають окремим компонентам вхідного вектора, тоді як шар Кохонена, який також називають шаром активних нейронів, відповідає за аналіз і кластеризацію. Цей шар може мати різну просторову конфігурацію: у

найпростішому випадку нейрони формують одновимірний ланцюжок, у більш загальному – двовимірну решітку прямокутної або квадратної форми, а в окремих реалізаціях – тривимірну структуру, що дає змогу відобразити складні багатовимірні закономірності.

Аналіз функціонування та навчального алгоритму мережі Кохонена дозволяє визначити низку переваг, особливо в умовах зашумлених або неповних даних. За рахунок заздалегідь фіксованої кількості нейронів, що відповідають класифікаційним кластерам, навчання відбувається стабільно, з поступовим коригуванням вагових коефіцієнтів, і не потребує значного часу для досягнення збіжності. Крім того, сам факт реалізації навчання без учителя вказує на високу адаптивність цієї моделі, яка здатна працювати з необробленими даними та виявляти приховані закономірності без втручання оператора. Такий підхід до навчання відображає глибинні біологічні принципи, що лежать в основі концепції штучних нейронних мереж, і підкреслює природну здатність системи до узагальнення та адаптації.

3 ПАРАЛЕЛЬНІ СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ

У межах функціональної організації системи керування базами даних можна виокремити низку логічно взаємопов'язаних компонентів, кожен з яких виконує специфічні завдання в межах обробки, зберігання та взаємодії з даними. До складу такої системи входять підсистеми, що забезпечують опис і підтримку структур бази даних, обробку користувацьких запитів, управління транзакціями, а також реалізацію інтерфейсів для введення, запиту й виведення інформації. Крім того, до складу СКБД входить блок формування звітів, який відповідає за агрегування та представлення результатів обробки у зручному для користувача форматі.

На рисунку 3.1 представлено узагальнену функціональну архітектуру системи керування базами даних, що відображає взаємодію її основних модулів і порядок проходження інформаційних потоків у процесі експлуатації системи.

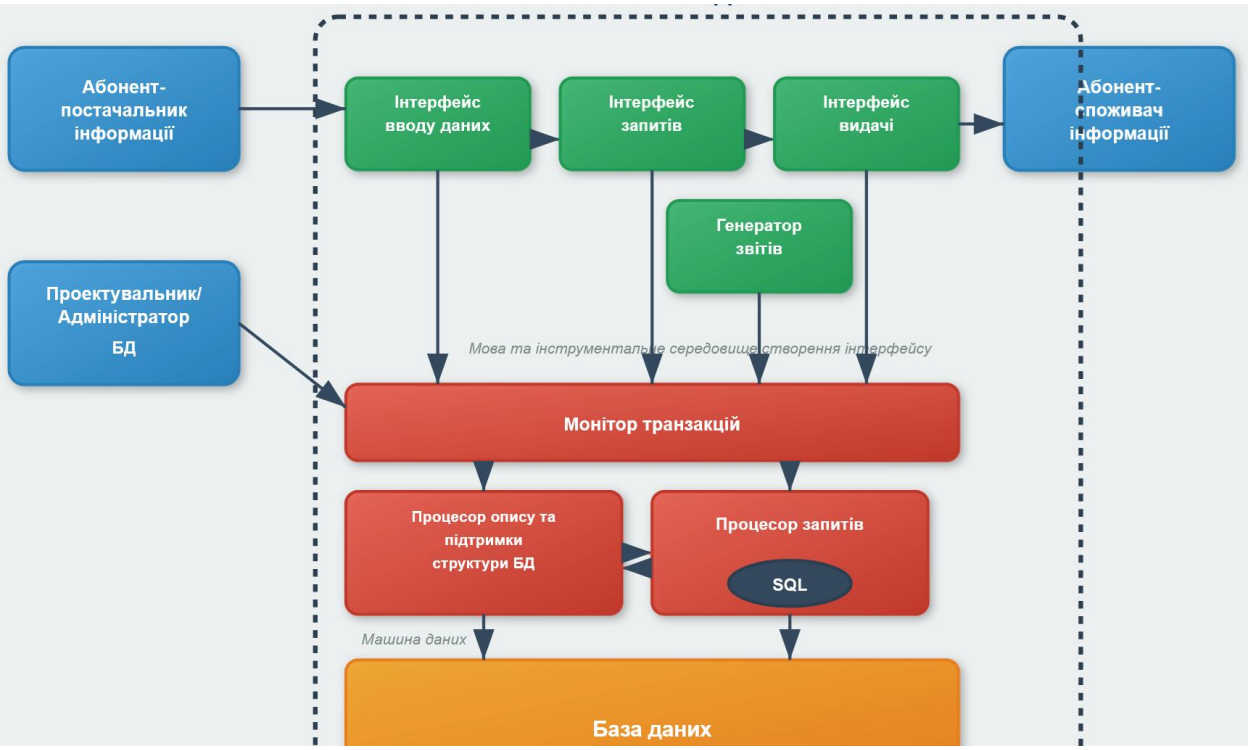

Рисунок 3.1 – Функціональна структура СКБД

На сучасному етапі розвитку інформаційних технологій однією з ключових задач у сфері управління даними є інтеграція методів інтелектуального аналізу у функціональне середовище реляційних систем керування базами даних. Така інтеграція може реалізовуватись за різними рівнями щільності взаємозв'язку між аналітичним модулем і СКБД, кожен з яких має свої переваги та обмеження.

У разі слабкого зв'язування система інтелектуального аналізу функціонує як автономна компонента, яка взаємодіє з базою даних опосередковано – шляхом експорту початкових даних зі сховища та імпорту результатів аналізу назад у нього. При середньому рівні інтеграції аналітична система використовує деякі вбудовані можливості СКБД, зокрема для виконання типових процедур підготовки даних, що часто передують основному аналізу. Такий підхід частково знижує обчислювальні витрати за рахунок делегування частини операцій базовій системі зберігання.

Найбільш тісну взаємодію забезпечує сильне зв'язування, за якого засоби інтелектуального аналізу інтегруються безпосередньо у функціональну архітектуру СКБД та розглядаються як її повноцінні елементи. У цьому випадку запити, що містять елементи аналітичної обробки, можуть бути оптимізовані на рівні механізмів зберігання, індексування та пошуку даних, характерних для конкретної реалізації СКБД. Такий підхід створює максимально зручні умови для кінцевих користувачів і прикладних розробників, хоча й потребує значних зусиль для реалізації через складність інтеграції та необхідність глибокої модифікації ядра системи.

Мова опису запитів для інтелектуального аналізу даних DMQL (Data Mining Query Language), концептуальні можливості якої представлені на рисунках 3.3–3.6, забезпечує формалізований механізм визначення параметрів задач добування знань із даних. За її допомогою можна конкретизувати набір даних, що підлягає аналізу, вказати тип аналітичного завдання – зокрема, класифікацію, виявлення асоціативних залежностей або

інші моделі. Крім того, мова дає змогу описувати семантичні ієрархії, закладені у структурі вхідних даних, а також задавати граничні значення параметрів, необхідні для контролю якості або глибини аналізу.

```
Algorithm: Знайти найближчу пару з обмеженням відстані
Input: S_I^m - множина елементів, n - мінімальна відстань індексів
Output: {pos_bsf, dist_bsf} - позиція та відстань найкращої пари

// Ініціалізація змінних
dist_bsf ← ∞        // найкраща відстань
pos_bsf ← -1        // позиція найкращої пари

// Основний цикл по всіх елементах
for i = 1 to |S_I^m| do
    current_element ← S_I^m[i]
    dist_min ← ∞    // мінімальна відстань для поточного елемента

    // Пошук найближчого елемента з обмеженням
    for j = 1 to |S_I^m| do
        if |i - j| ≥ n then    // перевірка обмеження відстані індексів
            dist ← EditDistance(S_I^m[i], S_I^m[j])

            // Раннє припинення, якщо знайдено дуже близьку пару
            if dist < dist_bsf then
                break    // перервати внутрішній цикл
            end if

            // Оновити мінімальну відстань для поточного елемента
            if dist < dist_min then
                dist_min ← dist
            end if
        end if
    end for

    // Оновити глобальний мінімум, якщо знайдено кращу пару
    if dist_min < dist_bsf then
        dist_bsf ← dist_min
        pos_bsf ← i
    end if
end for

return {pos_bsf, dist_bsf}
```

Рисунок 3.2 – Використання СКБД для інтелектального аналізу даних

Розробка DMQL стала основою для створення цілої низки спеціалізованих мов запитів, орієнтованих на задачі інтелектуального аналізу. Зокрема, у межах програмного середовища Microsoft було запропоновано стандарт OLE DB for Data Mining, що визначає відповідний інтерфейс прикладного програмування, а також мову запитів DMX (Data Mining Extensions), яка застосовується в компоненті SQL Server Analysis Services. Ці засоби розширюють традиційні можливості роботи з даними за рахунок інтеграції алгоритмів машинного навчання та інструментів побудови моделей безпосередньо в рамках середовища СКБД.

```
-- ============================================================
-- БЛОК 1: Пошук асоціативних правил для медичних даних
-- ============================================================

-- Знаходження асоціативних правил для пневмонії у пацієнтів старше 60
FIND ASSOCIATION RULES AS HealthRuleSet
    RELATED TO Salary, Age, isSmoker, Disease
    FROM HealthDB
    WHERE Disease = 'Pneumonia' AND Age > 60
    WITH SUPPORT_THRESHOLD = 0.05
    WITH CONFIDENCE_THRESHOLD = 0.07;

-- ============================================================
-- БЛОК 2: Видобуток специфічних правил з набору даних
-- ============================================================

-- Видобуток правил для аналізу зв'язку між курінням та пневмонією
MINE RULE HealthRuleSet AS
    SELECT DISTINCT
        1..n Disease AS antecedent,      -- передумова
        1..1 isSmoker AS consequent      -- наслідок
    FROM HealthDB
    WHERE antecedent.Disease = 'Pneumonia'
      AND antecedent.Age > 60
    EXTRACTING RULES WITH
        SUPPORT: 0.1,
        CONFIDENCE: 0.3;

-- ============================================================
-- БЛОК 3: Функція отримання правил з бази даних
-- ============================================================

FUNCTION GetRules(HealthDB)
RETURNS HealthRuleSet AS
BEGIN
    -- Отримання правил з підтримкою та довірою
    SELECT R INTO HealthRuleSet
    FROM HealthDB AS R
    WHERE R.Antecedent IN {
            (Disease = 'Pneumonia'),
            (Age > 60),
            (Salary > 1000)
          }
      AND R.Consequent IN {(isSmoker = TRUE)}
      AND R.Support > 0.1
      AND R.Confidence > 0.7;

    RETURN HealthRuleSet;
END;
```

Рисунок 3.3 – Приклади запитів

```
GetRules(HealthDB)
  into HealthRuleSet R
  where R.Body in {(Disease=*), (Age=*), (Salary=*)}
    and R.Body has {(Disease='Pneumonia'), (Age>60)}
    and R.Consequent in {(isSmoker=*)}
    and Support>0.1
    and Confidence>0.7

SelectRules(HealthRuleSet)
  where Body has {(Disease='Pneumonia')}
    and {(Salary>0) and (Salary<=1000)}
    and Support>0.1
    and Confidence>0.7
```

Рисунок 3.4 – Приклади запитів

```
-- ============================================================
-- БЛОК 4: Фільтрація та вибір найкращих правил
-- ============================================================

FUNCTION SelectRules(HealthRuleSet)
RETURNS FilteredRuleSet AS
BEGIN
    -- Вибір правил з високою якістю для пневмонії
    SELECT * FROM HealthRuleSet
    WHERE Antecedent HAS {(Disease = 'Pneumonia')}
      AND Consequent HAS {
            (Salary > 0),
            (Salary < 10000)
          }
      AND Support > 0.1
      AND Confidence > 0.7
      AND Lift > 1.0;  -- додатковий критерій якості
END;

-- ============================================================
-- БЛОК 5: Прогнозування асоціацій з використанням ML моделі
-- ============================================================

-- Прогнозування ризику пневмонії для конкретного пацієнта
SELECT
    PredictAssociation(
        [HealthMiningModel].[AssocLines],
        INCLUDE_STATISTICS,
        3  -- топ-3 асоціації
    ) AS PredictionResults
FROM [HealthMiningModel]
NATURAL PREDICTION JOIN (
    SELECT
        60 AS [Age],
        TRUE AS [isSmoker],
        'Pneumonia' AS [Disease]
) AS [PatientProfile];

-- ============================================================
-- БЛОК 6: Додатковий аналіз для покращення результатів
-- ============================================================

-- Аналіз кореляцій між факторами ризику
WITH RiskFactors AS (
    SELECT
        Age,
        isSmoker,
        Disease,
        Salary,
        COUNT(*) AS frequency,
        AVG(CASE WHEN Disease = 'Pneumonia' THEN 1.0 ELSE 0.0 END) AS pneumonia_rate
    FROM HealthDB
    WHERE Age > 50
    GROUP BY Age, isSmoker, Salary
)
SELECT
    *,
    pneumonia_rate * 100 AS risk_percentage
FROM RiskFactors
WHERE pneumonia_rate > 0.1
ORDER BY pneumonia_rate DESC;
```

Рисунок 3.5 – Приклади запитів

```
select
  PredictAssociation([HealthMiningModel].[AssocLines],
  INCLUDE_STATISTICS, 3)
  from [HealthMiningModel]
  natural prediction join (
    select
      60 as [Age],
      TRUE as [isSmoker],
      'Pneumonia' as [Disease]) as [AssocLines]
```

Рисунок 3.6 – Приклади запитів

Кластерна обчислювальна система являє собою сукупність взаємопов'язаних робочих станцій, інтегрованих у єдину обчислювальну

інфраструктуру за допомогою мережевих технологій, заснованих на шинній архітектурі або з використанням комутаційного обладнання. Така архітектура дозволяє створити масштабоване і надійне середовище для розподіленої обробки даних. У цьому контексті паралельні системи керування базами даних виступають інструментом, що забезпечує ефективне виконання запитів шляхом розподілу обчислювального навантаження між вузлами кластера.

Концептуальним підґрунтям організації обробки в паралельних СКБД є використання фрагментного паралелізму, який полягає в поділі вхідних даних на частини, що можуть оброблятися незалежно і паралельно. У реляційних базах даних ця ідея реалізується через горизонтальну фрагментацію, яка передбачає розподіл кортежів окремих відношень по дискових системах вузлів кластерної платформи. Такий розподіл здійснюється відповідно до певної функції фрагментації, яка визначає, які елементи відношення R повинні зберігатися на кожному конкретному вузлі. Це дозволяє оптимізувати обробку запитів, скорочуючи час доступу до даних та підвищуючи загальну продуктивність системи.

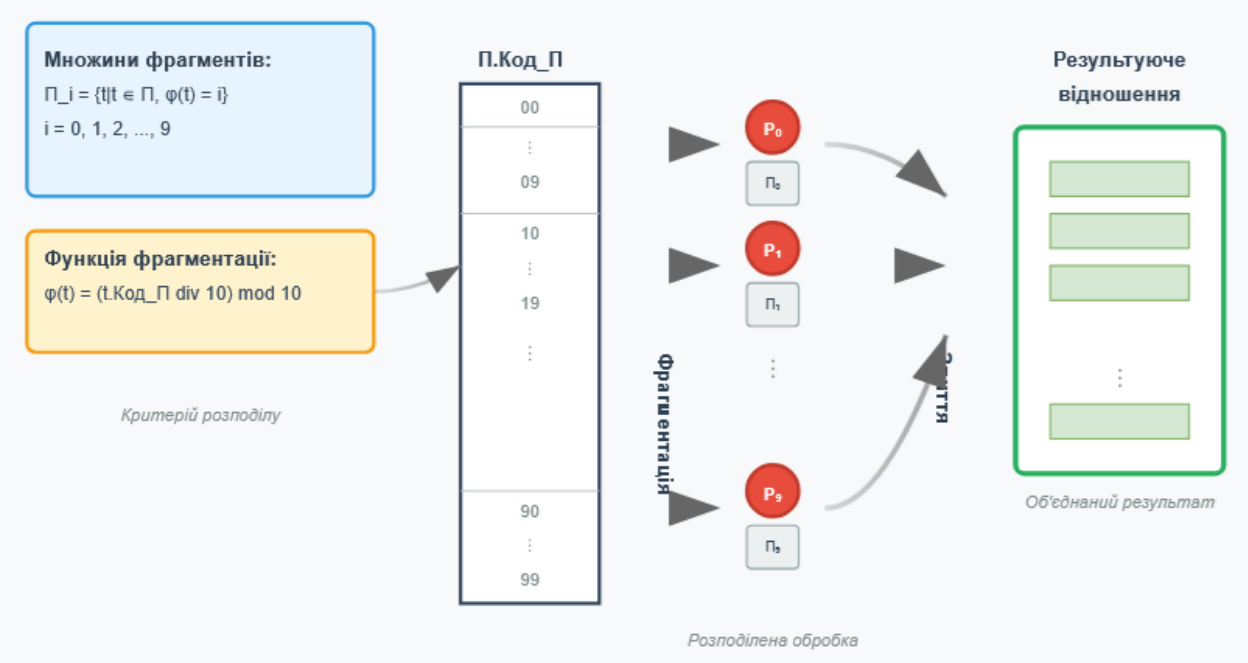

Рисунок 3. – Обробка запиту на основі фрагментного паралелізму

У паралельних системах керування базами даних обробка запиту здійснюється одночасно на всіх вузлах обчислювального кластера за участі множини паралельних агентів. Кожен з цих агентів відповідає за обробку певної частини фрагментованих відношень, які зберігаються на конкретному вузлі. Таким чином, загальний процес виконується децентралізовано, що дозволяє істотно підвищити продуктивність обробки запитів.

Виконання запиту в такій системі передбачає три послідовні фази. На початковому етапі SQL-запит надсилається користувачем на одну з машин кластера, яка тимчасово виконує роль головного вузла (host-машини). На цьому вузлі запит транслюється у послідовний фізичний план, який визначає порядок виконання базових операцій над даними. Далі цей план модифікується з метою забезпечення паралелізму: у відповідні місця структури запиту вставляються спеціальні оператори обміну – exchange, які визначають межі розподілу обробки між обчислювальними вузлами. У результаті формується паралельний фізичний план, що являє собою систему взаємодіючих агентів, кожен з яких реалізує частину загального завдання.

На завершальному етапі підготовлені паралельні агенти передаються з головної машини на відповідні вузли кластера. Там вони інтерпретуються виконавчим механізмом, який забезпечує локальну обробку даних у відповідності до заданого плану. Такий підхід дозволяє максимально ефективно використовувати ресурси обчислювальної системи та досягати значного прискорення при виконанні складних запитів у великих реляційних базах даних.

Рисунок 3.8 – Обробка запиту на основі фрагментного паралелізму

Оператори split, scatter, gather та merge, які формують функціональну основу механізму обміну в паралельному виконанні запитів, реалізуються відповідно до ітераторної моделі обробки. Кожен з цих операторів виконує спеціалізовану функцію в межах розподілу, маршрутизації та збору даних між обчислювальними вузлами.

Оператор split є бінарною конструкцією, призначеною для класифікації вхідних кортежів за приналежністю до поточного вузла. Ті з них, які асоціюються з локальним вузлом, зберігаються у вихідному буфері самого split-оператора, тоді як решта, що потребують обробки на віддалених вузлах, передаються у буфер scatter-оператора. Таким чином, відбувається логічне розділення потоку даних за принципом розподілу обчислювального навантаження.

Scatter є нульарним оператором, тобто таким, що не має вхідних операндів, і виконує обчислення функції маршрутизації, яка визначає, на який вузол слід переслати конкретний кортеж. Ці кортежі, що були класифіковані як «чужі» на попередньому етапі, передаються через відповідні канали обміну на визначені вузли системи, відповідно до заданої схеми.

Gather також є нульарним оператором і відповідає за зворотну операцію – прийом кортежів, що надходять з інших вузлів, через порт обміну. Отримані дані накопичуються у його власному буфері та згодом використовуються для подальшої обробки.

Оператор merge виконує об’єднання потоків даних і реалізується як бінарна конструкція, що приймає два вхідні джерела. Його основна функція полягає в інтеграції локальних та отриманих іззовні результатів у єдиний потік, який зберігає коректну логіку обчислень, забезпечуючи завершення етапу обміну в межах паралельного виконання запиту.

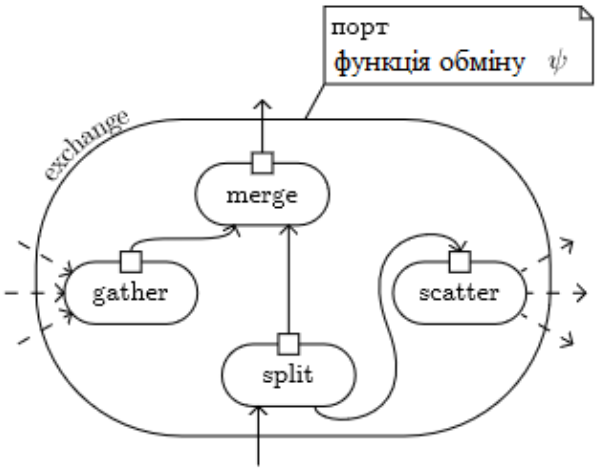

Рисунок 3.9 – Структура оператора обміну та паралельний план запиту

Кластерний аналіз являє собою багатовимірну статистичну методику, мета якої полягає у виявленні внутрішньої структури вхідних даних шляхом групування об’єктів дослідження у відносно однорідні множини, що мають назву кластери. Цей процес передбачає початкове опрацювання інформації, яка описує вибірку, а далі – побудову таких груп, у межах яких спостерігається максимальна схожість між елементами за заданими критеріями. Метод кластеризації використовується з метою узагальнення, сегментації та виявлення прихованих закономірностей у складних даних. Концептуальні та алгоритмічні засади кластерного аналізу докладно розглянуто у розділі 2.

3.1 Модифікований метод нечіткої кластеризації даних

Реалізація нечіткої кластеризації в середовищі системи керування базами даних здійснюється шляхом інтеграції алгоритму Fuzzy C-Means (FCM) у паралельну реляційну СКБД. Такий підхід передбачає представлення вхідних даних, проміжних обчислень і результатів у вигляді реляційних таблиць, які зберігаються в базі даних із заздалегідь визначеною схемою, адаптованою до вимог алгоритму. Всі обчислювальні процедури

реалізуються через SQL-запити, які виконуються над відповідними фрагментами таблиць.

З метою забезпечення паралелізму таблиці бази даних зазнають горизонтальної фрагментації, внаслідок чого кортежі розподіляються між вузлами кластерної обчислювальної системи. Кожен вузол функціонує незалежно і виконує обчислення на основі локального фрагмента таблиці, використовуючи екземпляр паралельної СКБД, запущеної на ньому.

Алгоритм FCM реалізується як процедура поступової мінімізації цільової функції JFCMJFCM, яка визначає якість нечіткого розбиття вихідного простору об'єктів. Протягом ітераційного процесу відбувається оновлення матриці центроїдів CC та матриці ступенів належності UU згідно з відповідними формулами, які визначають ступінь приналежності кожного об’єкта до кластерів.

У процесі реалізації алгоритму необхідно здійснювати агрегування координат векторів множини XX, зокрема виконувати операції обчислення сум, знаходження максимумів та інших характеристик. Проте стандартні агрегатні функції SQL обмежені у застосуванні, оскільки вони орієнтовані на обчислення за рядками таблиць, а не за стовпцями. Це створює додаткові труднощі в контексті реалізації агрегування по атрибутах в реляційній моделі даних і вимагає спеціальних процедур або альтернативних засобів обробки для досягнення коректного обчислення цільових значень.

Інтеграція інтелектуального аналізу даних у середовище реляційної системи керування базами даних має на меті надати прикладному програмісту ефективні та зручні інструменти, які дозволяють виконувати складні аналітичні операції без необхідності виходу за межі функціонального простору СКБД. Такий підхід забезпечує не лише прямий доступ до механізмів добування знань із даних, але й дає змогу здійснювати зберігання й обробку результатів аналізу в межах єдиного інформаційного середовища.

```
-- ============================================================
-- ПОВНИЙ АЛГОРИТМ FUZZY C-MEANS
-- Математичні формули та SQL реалізація
-- ============================================================

/*
МАТЕМАТИЧНІ ОСНОВИ АЛГОРИТМУ:

1. Цільова функція (мінімізується):
   J_FCM(X,k,m) = ∑∑ u_ij^m ρ²(x_i, c_j)
                  i=1 j=1

2. Оновлення центроїдів:
   ∀j,ℓ  c_jℓ = (∑ u_ij^m · x_iℓ) / (∑ u_ij^m)
                i=1            i=1

3. Оновлення коефіцієнтів членства:
   u_ij = ∑ (ρ(x_i,c_j)/ρ(x_i,c_l))^(2/(1-m))
          l=1

4. Критерій збіжності:
   max |u_ij^(s+1) - u_ij^(s)| ≤ ε
    ij

Де:
- X = {x_1, x_2, ..., x_n} - множина об'єктів
- k - кількість кластерів
- m - параметр нечіткості (зазвичай m = 2)
- u_ij - ступінь належності об'єкта i до кластера j
- c_j - центроїд кластера j
- ρ(x_i,c_j) - відстань між об'єктом i та центроїдом j
- ε - поріг збіжності
*/

-- ============================================================
-- СТРУКТУРА ДАНИХ: Створення тимчасових таблиць
-- ============================================================

-- Таблиця матриці членства U(i,j) = ступінь належності об'єкта i до кластера j
CREATE TEMP TABLE MEMBERSHIP_MATRIX (
    object_i INTEGER NOT NULL,        -- індекс об'єкта
    cluster_j INTEGER NOT NULL,       -- індекс кластера
    membership_value NUMERIC(10,6),   -- коефіцієнт u_ij
    PRIMARY KEY (object_i, cluster_j)
);

-- Таблиця параметрів алгоритму P(d,k,n,s,delta)
CREATE TEMP TABLE ALGORITHM_PARAMETERS (
    dimensions_d INTEGER,             -- розмірність простору ознак
    clusters_k INTEGER,               -- кількість кластерів
    objects_n INTEGER,                -- кількість об'єктів
    iteration_s INTEGER,              -- номер поточної ітерації
    convergence_delta NUMERIC(10,8), -- критерій збіжності
    PRIMARY KEY (iteration_s)
);

-- Допоміжна таблиця для збереження попередніх значень SV(i,l)
CREATE TEMP TABLE SAVED_VALUES (
    object_i INTEGER NOT NULL,        -- індекс об'єкта
    feature_l INTEGER NOT NULL,       -- індекс ознаки
    feature_value NUMERIC(12,6),      -- значення ознаки
    PRIMARY KEY (object_i, feature_l)
);
```

Рисунок 3.10 – Метод pgFCM

Архітектура інтеграції інтелектуального аналізу даних із відкритим вихідним кодом, призначена для функціонування в межах реляційної СКБД, передбачає наявність низки спеціалізованих компонентів. Зокрема, компонент pgMining відповідає за виконання аналітичних обчислень безпосередньо всередині СКБД, забезпечуючи інтерфейс бібліотечних функцій для реалізації алгоритмів добування знань. Паралельно функціонує компонент mcMining, який орієнтований на ефективне виконання

паралельних алгоритмів у оперативній пам’яті, оптимізованих для багатоядерних процесорів. У структурі рішення також передбачено використання стандартного низькорівневого інтерфейсу доступу до даних, який у випадку PostgreSQL реалізується через механізм Server Programming Interface (SPI).

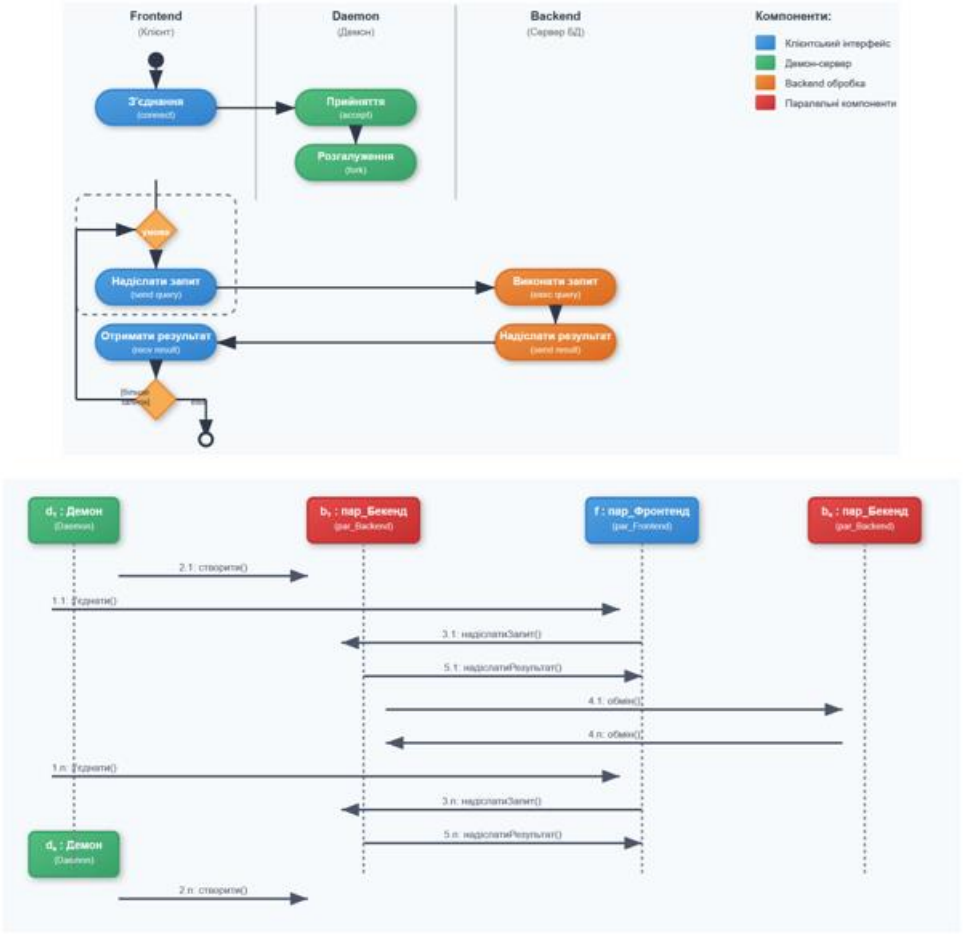

Рисунок 3.11 – Архітектура паралельної СКБД на базі PostgreSQL. Клієнт-серверна взаємодія

3.2 Реалізація архітектур

Компонент pgMining включає дві ключові підсистеми: Frontend і Backend. Підсистема Frontend виступає у ролі інтерфейсу для прикладного програміста й забезпечує засоби для ініціалізації процедур аналізу, отримання результатів і їхнього представлення у форматі JSON. Водночас

Backend відповідає за внутрішню реалізацію функціональності, до складу якої входять модулі Wrapper і Cache manager. Модуль Wrapper надає обгортки для функцій з mcMining, що дозволяє запускати паралельні алгоритми та повертати результати у структурованому вигляді для подальшого збереження в базі. Компонент Cache manager реалізує буферний механізм, який забезпечує тривале зберігання обчислених проміжних структур у оперативній пам'яті, що суттєво знижує витрати на повторне виконання операцій і підвищує ефективність у випадках повторного використання вже згенерованих результатів.

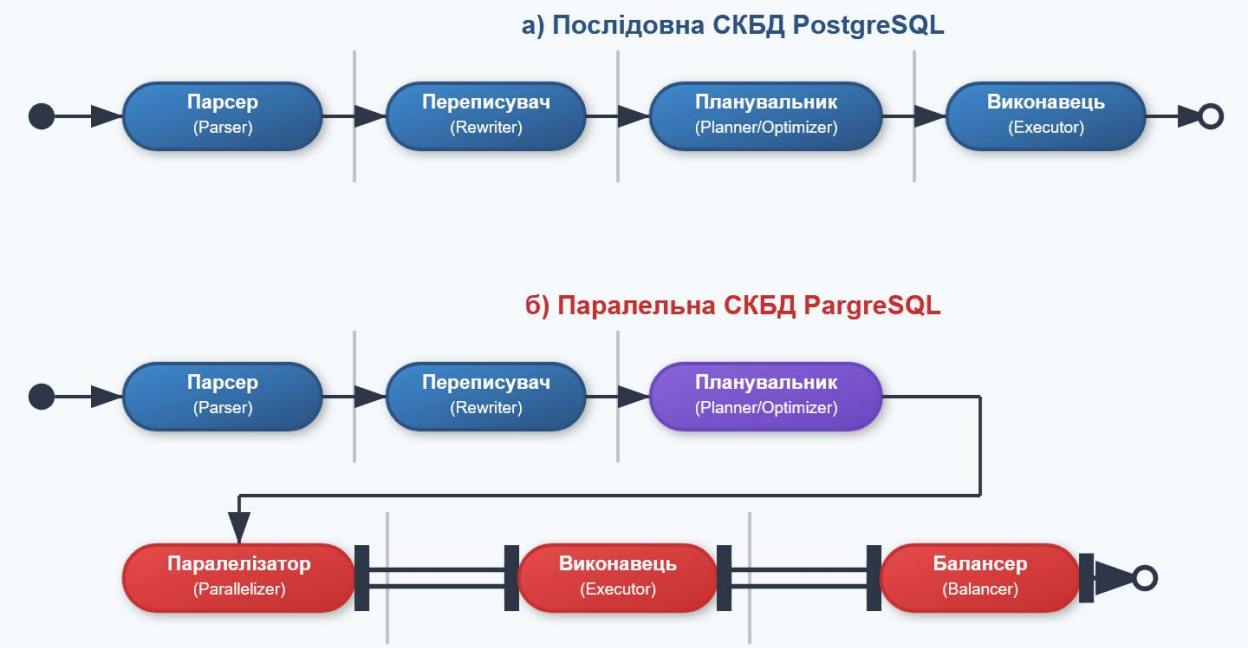

Рисунок 3.12 – Архітектура паралельної СКБД на базі PostgreSQL

Архітектурна основа системи керування базами даних PargreSQL ґрунтується на класичній моделі «клієнт–сервер», що є похідною від архітектури, реалізованої в базовій системі PostgreSQL. Така модель забезпечує чіткий розподіл функціональності між клієнтською частиною, яка ініціює запити до бази даних, і серверною частиною, що відповідає за їх обробку та повернення результатів. Взаємодія окремих процесів у PargreSQL, з урахуванням особливостей розширення системи порівняно з класичною реалізацією PostgreSQL, графічно представлена на рисунку 3.14. Ця схема

демонструє, яким чином організована координація між підсистемами, що реалізують доступ до даних, виконання паралельних обчислень та забезпечення аналітичної обробки.

Під час роботи в кластерному середовищі клієнтська програма послідовно встановлює з'єднання з кожним демоном СКБД, що функціонує на окремих вузлах системи. У результаті на кожному обчислювальному вузлі ініціалізується запуск окремого серверного компонента par_Backend, який відповідає за обробку запитів, що надходять від клієнта. Після встановлення всіх з'єднань клієнт послідовно передає сформований SQL-запит кожному з активованих компонентів.

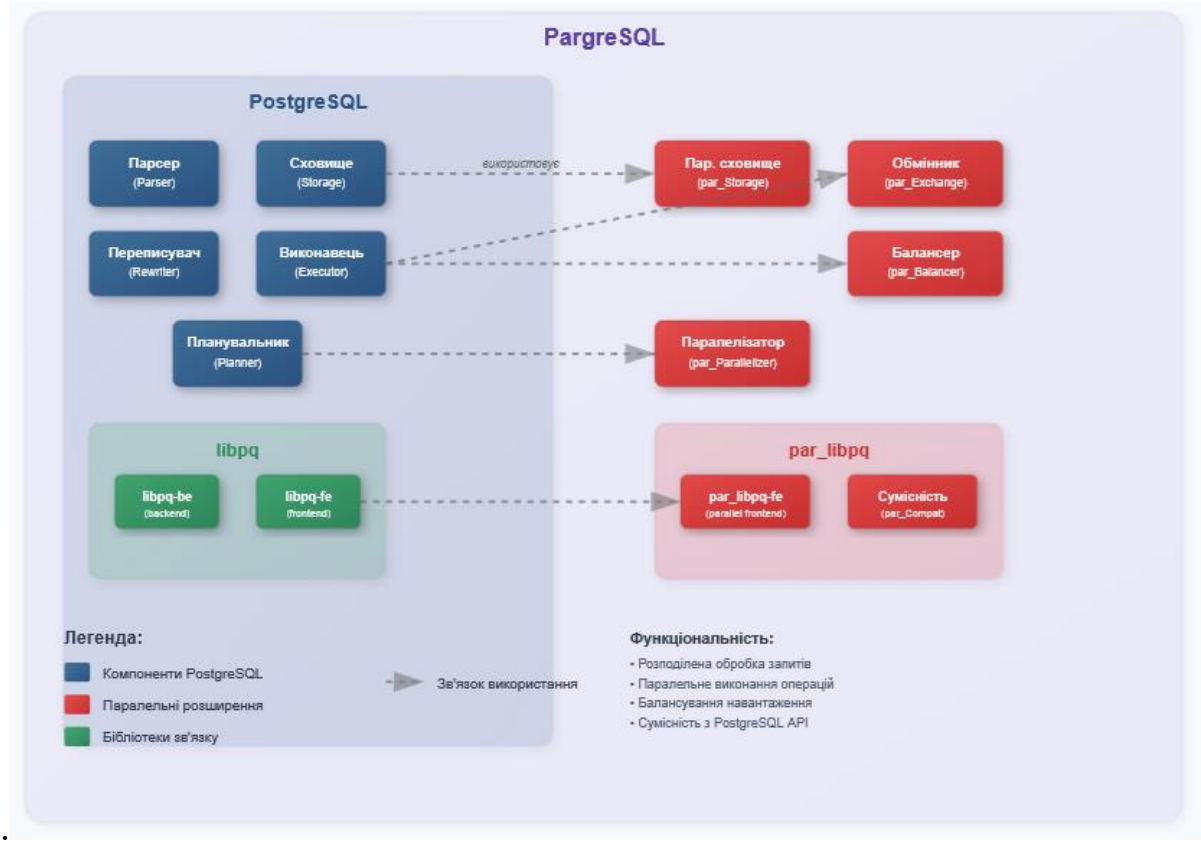

Рисунок 3.13 – Архітектура паралельної СКБД на базі PostgreSQL

Функціональна схема обробки запиту в середовищі PargreSQL представлена на рисунку 3.15. Обробка запиту в системі здійснюється через класичну послідовність етапів – parse, rewrite, plan/optimize та execute, які реалізуються відповідно до логіки, притаманної традиційній реалізації у

послідовних СКБД. Це забезпечує збереження сумісності з базовими принципами PostgreSQL і водночас дозволяє розширити можливості за рахунок використання кластерних обчислювальних ресурсів.

У процесі експериментального дослідження продуктивності системи PargreSQL було проведено випробування, що передбачали виконання SQL-запиту з операцією природного з'єднання двох відношень за спільним атрибутом. Вхідні таблиці мали значні обсяги: перше відношення містило приблизно 300 мільйонів кортежів, друге – близько 7,5 мільйона. Обидва відношення були рівномірно розподілені між вузлами кластерної системи з метою забезпечення максимальної ефективності паралельної обробки.

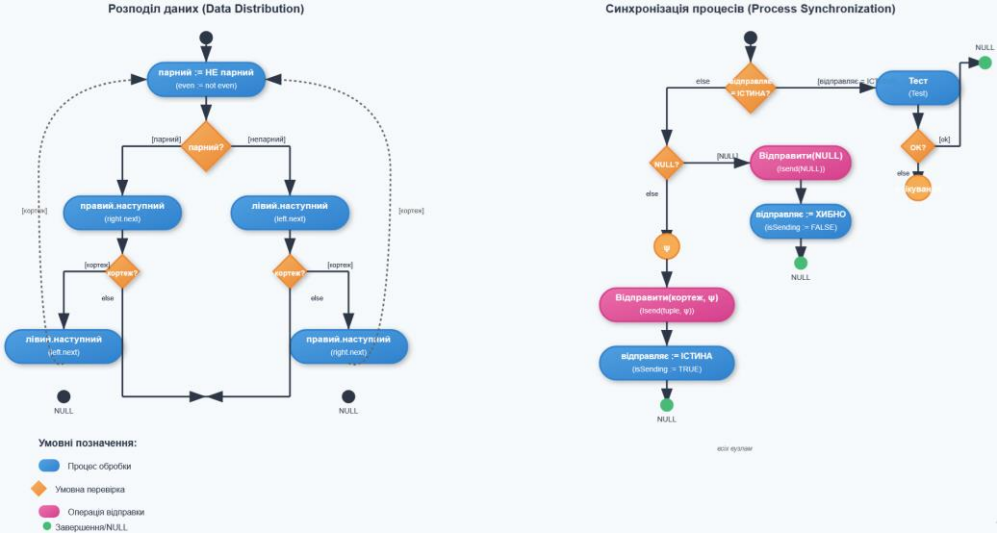

Рисунок 3.14 – Архітектура паралельної СКБД на базі PostgreSQL.

Обробка запиту

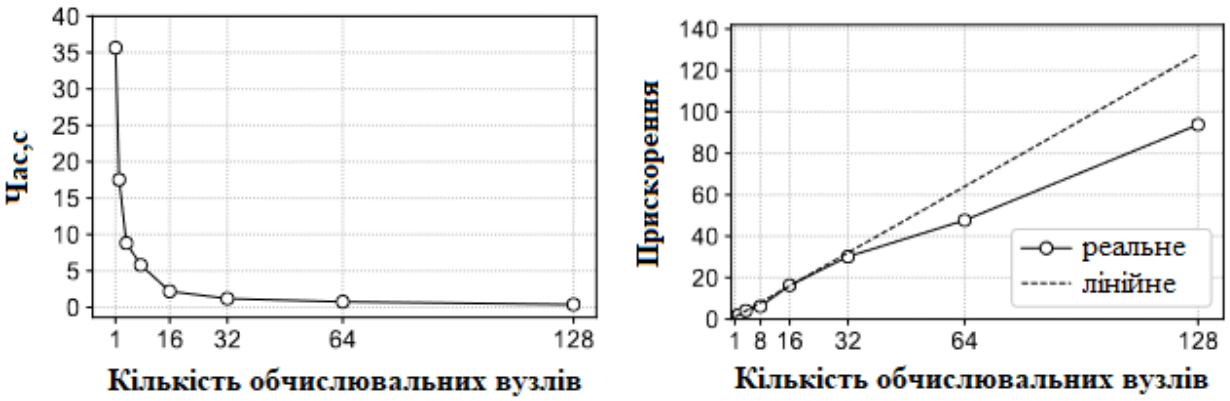

Рисунок 3.15 – Результати роботи

Результати експериментів, представлені на слайді, свідчать про те, що система PargreSQL забезпечує майже лінійне масштабування. Досягнуте прискорення в межах від 75% до 100% від ідеального лінійного приросту свідчить про високу ефективність реалізованої архітектури та здатність системи повноцінно використовувати ресурси багатовузлового обчислювального середовища при виконанні складних запитів з великою вибіркою.

ВИСНОВКИ

У роботі проаналізовано аспекти інтеграції методів інтелектуального аналізу даних у реляційні системи керування базами даних, а також окреслено підходи до розробки паралельних алгоритмів для кластерних обчислювальних систем, які функціонують на основі сучасних багатоядерних прискорювачів. Зокрема, подано опис архітектури системи та принципів реалізації інтегрованого рішення на базі відкритої СКБД PostgreSQL із використанням апаратної платформи Intel Many Integrated Core.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Flach P. A. (2012). Machine Learning: The Art and Science of Algoritms that Makes Sense of Data. Cambridge: Cambridge University Press. 291 p. https://doi.org/10.1017/CBO9780511973000

2. Obermeyer Z, Emanuel EJ. Predicting the Future–Big Data, Machine Learning, and Clinical Medicine. New England Journal of Medicine. 375(13). 2016; p. 1216-1219.

3. Rajkomar A, Dean J, Kohane I. Machine Learning in Medicine. New England Journal of Medicine. 380(14). 2019.p. 1347–1358.

4. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 542(7639).2017; p. 115–118.

5. Topol EJ. High-performance medicine: the convergence of human and artificial intelligence. Nature Medicine. 25(1) 2019. p.44–56.

6. Beam AL, Kohane IS. Big Data and Machine Learning in Health Care. JAMA. 320(11). 2018; p.1101–1102.

7. Bohr A, Memarzadeh K, editors. Artificial Intelligence in Healthcare. Academic Press; 2020.

8. Lundervold AS, Lundervold A. An overview of deep learning in medical imaging focusing on MRI. Zeitschrift für Medizinische Physik. 29(2). 2019. p.102–127.

9. W. Frawley, G. Piatetsky-Shapiro, C. Matheus Knowledge Discovery in Databases: An Overview. - AI Magazine. - 1992. pp. 213-228.

10. Kitchin Rob. The Data Revolution. United States: Sage. 2014, p. 6.

11. Piatetsky-Shapiro G, Frawley W J. Knowledge Discovery in Databases. USA: MIT Press, 1991.

12. Agrawal R., Mannila H., Srikant R., Toivonen H. and Verkamo I. Fast Discovery of Association Rules. In Advances in Knowledge Discovery and Data

Mining, eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Menlo Park, Calif.: AAAI Press, 1996, pp. 307-328.

13. Fayyad U., Piatetsky-Shapiro G., Smyth P., Advances in Knowledge Discovery and Data Mining, (Chapter 1), AAAI/MIT Press, 1996.

14. Judea Pearl, Stuart Russell. Bayesian Networks. UCLA Cognitive Systems Laboratory, Technical Report (R-277), November 2000.

15. Етапи обробки медичних даних для розв’язання задач алгоритмами машинного навчання / Дацок О.М., Дяченко В.О., Гук А.С., Фатій Л.Р. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2025. – № 2(14). – С. 100 – 116