Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів візуалізації та аналізу інформації з використанням LLM
(тема)

Виконав:
здобувач 2 року навчання
групи ІПЗм-23-3

Максим БУДНИК
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова

Керівник доц. Віра ГОЛЯН
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Буднику Максиму Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів візуалізації та аналізу інформації з використанням LLM»
Затверджена наказом по університету від 15.04. 2025р. № 290 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 01.06.2025
3. Вихідні дані до роботи Відкриті дані та інформація про системи для візуалізації даних, відкрити дані про моделі LLM, існуючі моделі LLM, мова програмування Typescript, платформа Nodejs.
4. Перелік питань, що потрібно опрацювати в роботі
Вступ, аналіз предметної галузі, огляд існуючих підходів та інструментів візуалізації даних, обгрунтування проблеми і актуалізація рішень, постановка задачі, дослідження принципів роботи LLM, порівняння моделей LLM, створення прототипу системи, обґрунтування обраних рішень, дослідження роботи LLM, аналіз результатів.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	16.04	*виконано*
2	Аналіз предметної галузі і постановка задачі	16.04 - 22.04	*виконано*
3	Теоретичне дослідження	22.04 - 30.04	*виконано*
4	Створення прототипу і експериментальне дослідження	30.04 - 12.05	*виконано*
5	Побудова промпту	12.05 - 18.05	
6	Підготовка до апробації результатів дослідження. Публікація матеріалів	18.05 - 20.05	*виконано*
7	Підготовка пояснювальної записки	20.05 - 24.05	*виконано*
8	Підготовка презентації та доповіді	24.05 - 26.05	*виконано*
9	Перевірка на плагіат	26.05	*виконано*
10	Нормоконтроль	27.05	*виконано*
11	Рецензування	09.06	*виконано*
12	Попередній захист	15.06	*виконано*
13	Занесення диплома в електронний архів	16.06	*виконано*
14	Допуск до захисту у зав. кафедри	16.06	*виконано*

Дата видачі завдання 16 квітня 2025р.

Студент (ка) _____ Максим БУДНИК
(підпис)

Керівник роботи _____ доц. Віра ГОЛЯН
(підпис) (посада, Власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 75 с., 11 рис., 21 джерело.

АНАЛІЗ ДАНИХ, АРХІТЕКТУРА ПЗ, ВІЗУАЛІЗАЦІЯ, ІНСАЙТИ, LLM, МОДЕЛІ, ОБРОБКА ДАНИХ, ПРОМПТ, ПРОМПТ-ІНЖИНІРИНГ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об'єктом дослідження є процес дослідження та розробки системи для візуалізації та аналізу даних з використанням великих мовних моделей (LLM).

Метою роботи є дослідження принципів застосування LLM для аналізу та візуалізації даних, розробка концептуальної архітектури та програмного прототипу такої системи, дослідження методів генерації інсайтів та візуалізацій, порівняння ефективності різних моделей LLM (зокрема GPT-4o, Claude 3, Gemini Pro) та дослідження оптимальних технік побудови промптів для даної задачі.

Методами дослідження є огляд літератури, системний аналіз, моделювання архітектури, аналіз інженерії промптів, розробка прототипу, експериментальне дослідження та методи порівняльного аналізу.

У результаті науково-дослідної практики проаналізовано предметну область та існуючі інструменти. Розроблено концептуальну архітектуру системи та створено програмний прототип, здатний взаємодіяти з різними LLM для автоматизованого аналізу та генерації інтерактивних візуалізацій і текстових інсайтів. Описано алгоритми обробки, аналізу та генерації візуалізацій за допомогою LLM. Проведено порівняльний аналіз обраних моделей LLM на основі експериментальних результатів. Досліджено та сформульовано принципи ефективної побудови промптів для даної задачі. Обґрунтовано вибір технологій (Next.js, AI SDK, Recharts, Zod, PostgreSQL).

Безпосередня реалізація повноцінної системи для комерційного впровадження не є метою цієї роботи. Робота є науково-дослідною практикою.

Робота базується на аналізі сучасних досліджень у галузі великих мовних моделей, візуалізації даних та програмної інженерії, зокрема на працях Vaswani A. та ін. ("Attention is all you need"), Brown T. B. та ін. ("Language models are few-shot learners"), а також дослідженнях науковців кафедри програмної інженерії ХНУРЕ.

Результати дослідження та розроблений прототип можуть бути використані як основа для подальшої розробки інтелектуальних систем аналізу даних. Запропоновані підходи до інтеграції LLM та побудови промптів можуть бути застосовані для створення інструментів, що спрощують аналіз даних для користувачів без спеціалізованих навичок програмування. Подальший розвиток може включати розширення типів візуалізацій, обробку потокових даних та оптимізацію алгоритмів.

Досліджені методи можуть бути використані в різноманітних галузях від аналізу продажів та фінансової звітності до аналізу медичних даних та освітніх результатів.

Застосування запропонованих методів може значно скоротити час на аналіз даних та підвищити якість прийнятих рішень завдяки швидкому отриманню інсайтів. Це дозволить широкому колу користувачів ефективніше використовувати наявні дані, що є критично важливим у різних сферах діяльності.

Робота робить внесок у розуміння потенціалу LLM для автоматизації аналізу та візуалізації даних, надаючи інструмент для користувачів без навичок програмування. Це відкриває нові можливості для швидкого отримання цінної інформації з даних та прийняття обґрунтованих рішень.

DATA ANALYSIS, SOFTWARE ARCHITECTURE, VISUALIZATION, INSIGHTS, LLM, MODELS, DATA PROCESSING, PROMPT, PROMPT ENGINEERING, ARTIFICIAL INTELLIGENCE

The object of this research is the process of studying and developing a system for data visualization and analysis using large language models (LLMs).

The aim of the work is to explore the principles of applying LLMs for data analysis and visualization, to develop a conceptual architecture and a software prototype of such a system, to investigate methods for generating insights and visualizations, to

compare the efficiency of various LLMs (specifically GPT-4o, Claude 3, and Gemini Pro), and to explore optimal prompt engineering techniques for this task.

The research methods include literature review, systems analysis, architectural modeling, prompt engineering analysis, prototype development, experimental research, and comparative analysis techniques.

As a result of the research practice, the subject area and existing tools were analyzed. A conceptual system architecture was developed and a software prototype was created, capable of interacting with various LLMs for automated analysis and generation of interactive visualizations and textual insights. Algorithms for processing, analyzing, and generating visualizations using LLMs were described. A comparative analysis of selected LLMs based on experimental results was conducted. Principles of effective prompt engineering for this task were examined and formulated. The choice of technologies (Next.js, AI SDK, Recharts, Zod, PostgreSQL) was justified.

The direct implementation of a fully functional commercial system is not the goal of this work. The project represents a scientific research practice.

The work is based on the analysis of modern studies in the field of large language models, data visualization, and software engineering, particularly the works of Vaswani et al. ("Attention is All You Need"), Brown et al. ("Language Models are Few-Shot Learners"), as well as research conducted by the Software Engineering Department of Kharkiv National University of Radio Electronics (KhNURE).

The research results and developed prototype can serve as a foundation for the further development of intelligent data analysis systems. The proposed approaches to LLM integration and prompt design can be applied to create tools that simplify data analysis for users without specialized programming skills. Future development may include expanding the range of visualizations, processing streaming data, and optimizing algorithms.

The explored methods can be applied across various domains - from sales and financial reporting analysis to medical data and educational performance evaluation.

The application of the proposed methods can significantly reduce the time required for data analysis and improve decision-making quality through rapid insight

generation. This enables a broad range of users to utilize available data more efficiently, which is critically important across many fields.

This work contributes to the understanding of LLM potential in automating data analysis and visualization, providing a tool for users without programming expertise. It opens new opportunities for quickly extracting valuable information from data and making informed decisions.

Завідувачу кафедри
ПІ_____
(скорочена назва кафедри)
проф. Кирилу СМЕЛЯКОВУ
(вчене звання, сласне ім'я, прізвище)

ЗАЯВА
щодо самостійності виконання кваліфікаційної роботи та можливості її публікації (та/або публікації анотації кваліфікаційної роботи) в електронному архіві відкритого доступу ElAr KhNURE

Я,_____Будник Максим Олексійович_____
(прізвище, ім'я, по батькові)

здобувач вищої освіти на другому (магістерському) рівні вищої освіти академічної групи___ІПЗм-23-3_____

кафедра_____ програмної інженерії _____,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему «Дослідження методів візуалізації та аналізу інформації з використанням LLM»_____,
(назва роботи)

що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в репозиторії "ElArKhNURE". Погоджуюся з авторським договором, відповідно до Положення про репозиторій ХНУРЕ "ElArKhNURE". Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з вимогами академічної доброчесності, згідно з якими виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата Підпис

ПЕРЕЛІК СКОРОЧЕНЬ

API - Інтерфейс прикладного програмування (Application Programming Interface)

AWS - Веб-сервіси Amazon (Amazon Web Services)

CSV - Значення, розділені комами (Comma-Separated Values)

GPT - Генеративний попередньо навчений трансформер (Generative Pre-trained Transformer)

HTML5 - Мова гіпертекстової розмітки, версія 5 (HyperText Markup Language version 5)

JSON - Нотація об'єктів JavaScript (JavaScript Object Notation)

LLM - Велика мовна модель (Large Language Model)

LSTM - Довга короткострокова пам'ять (Long Short-Term Memory)

NER - Розпізнавання іменованих сутностей (Named Entity Recognition)

NLP - Обробка природної мови (Natural Language Processing)

SQL - Мова структурованих запитів (Structured Query Language)

SVG - Масштабована векторна графіка (Scalable Vector Graphics)

TF-IDF - Частота терміну – обернена частота документа (Term Frequency-Inverse Document Frequency)

UUID - Універсально унікальний ідентифікатор (Universally Unique Identifier)

ЗМІСТ

ВСТУП

Сучасний світ стикається з експоненційним зростанням обсягів даних, що вимагає ефективних інструментів для їх аналізу та візуалізації. Існуючі рішення (наприклад, Tableau, Power BI) орієнтовані на технічних користувачів, мають складний інтерфейс або обмежену підтримку автоматизованого аналізу. Це обмежує доступність аналітики для широкого кола користувачів, особливо без навичок програмування. Великі мовні моделі (LLM) пропонують потенціал для автоматизації цих процесів, але їх інтеграція в аналітичні системи залишається недостатньо дослідженою.

Автоматизація аналізу даних за допомогою LLM може спростити отримання інсайтів, зменшити витрати часу на обробку інформації та зробити аналітику доступною для некваліфікованих користувачів. Це критично важливо для галузей, де швидкість прийняття рішень залежить від оперативного аналізу даних: фінанси, охорона здоров’я, маркетинг.

Дослідження відповідає напрямкам кафедри Програмної Інженерії ХНУРЕ в галузі штучного інтелекту, обробки даних та розробки масштабованих систем. Воно розвиває ідеї, висвітлені в роботах науковців кафедри щодо застосування нейронних мереж для класифікації текстів та ідентифікації фейкових новин [1, 2].

Метою даної науково-дослідницької роботи є дослідження можливість створення інструменту для швидкої візуалізації та аналізу даних, який би дозволив користувачам, незалежно від їхніх технічних навичок, отримувати корисну інформацію, знаходити залежності та отримувати візуалізацію своїх даних.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- проаналізувати існуючі підходи до візуалізації та аналізу даних, виявити їхні недоліки та можливості для покращення;
- розробити архітектуру прототипу системи, яка дозволяє використовувати LLM для аналізу даних з різних джерел (csv, json) та автоматизованого створення візуалізацій;
- дослідити методи аналізу та візуалізації інформації за допомогою LLM;

– описати алгоритми аналізу та візуалізації даних, використовуючи можливості LLM, та їхню взаємодію в системі;
– дослідити проблеми, пов'язані з використанням LLM для аналізу даних, такі як конфіденційність, достовірність, та обмеження;
– провести порівняльний аналіз використання різних LLM.

Об'єктом дослідження є процес обробки, аналізу та візуалізації даних з використанням великих мовних моделей.

Предметом дослідження є методи візуалізації та аналізу інформації, що використовує LLM для автоматизації процесу пошуку інсайтів та візуалізації, а також теоретичні основи її функціонування.

Методи дослідження та аналізу: У процесі розробки будуть застосовані методи системного аналізу предметної області, проектування інтерфейсів, розробки програмного забезпечення, промпт-інжиніринг, а також методи порівняльного аналізу та оцінки ефективності.

Наукова новизна полягає у:

– вперше запропоновано архітектуру системи, що інтегрує різні LLM через уніфікований API для аналізу даних;

– розроблено методику генерації інтерактивних візуалізацій на основі динамічних промитів, адаптованих під специфіку даних;

– встановлено залежність між якістю промитів та точністю результатів аналізу

В результаті виконання науково-дослідницької роботи буде запропоновано архітектуру системи для візуалізації та аналізу даних з використанням LLM, створено прототип, описані алгоритми аналізу та візуалізації, а також досліджені основні проблеми та обмеження, пов'язані з використанням LLM в цій галузі. Також було проведено порівняльний аналіз використання різних моделей для аналізу інформації.

Досліджені методи можуть бути використані в різноманітних галузях - від аналізу продажів та фінансової звітності до аналізу медичних даних та освітніх результатів, що дозволить широкому колу користувачів ефективніше

використовувати наявні дані, хоча безпосередня реалізація повноцінної системи не є метою цієї роботи.

За результатами науково-дослідної практики, основні положення та висновки роботи представлено у статті «Дослідження методів візуалізації та аналізу інформації з використанням LLM», яка прийнята до публікації в №21 студентського наукового журналу «UNIVERSUM» (випуск заплановано на 20 червня 2025 року).

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі дослідження

Галузь візуалізації та аналізу даних постійно розвивається, пропонуючи все більше інструментів для обробки, аналізу та візуалізації інформації. Розвиток технологій створює як нові можливості, так і нові виклики. Основною метою цього розділу є надання детального огляду сучасних інструментів та підходів, виявлення їхніх сильних і слабких сторін, а також визначення ключових тенденцій і проблем, що впливають на галузь.

Давайте детальніше розглянемо можливості перелічених інструментів для аналізу та візуалізації даних:

Datadog (див. рис. 1.1).

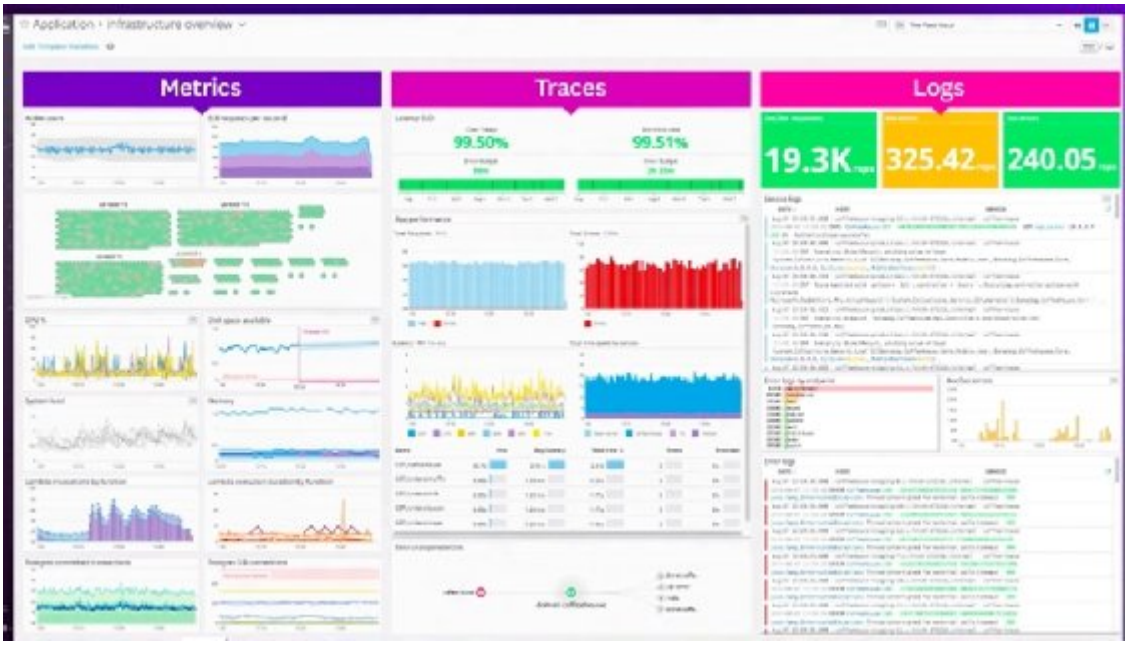

Рисунок 1.1 – Дашбоард Datadog (рисунок створено самостійно)

Datadog є потужним інструментом для моніторингу продуктивності IT-інфраструктури. Він дозволяє збирати, агрегувати та візуалізувати дані з різних джерел, включаючи сервери, бази даних, додатки та хмарні сервіси. Платформа надає велику кількість інтеграцій для збору даних. Можливості аналізу даних включають створення дашбордів з різними типами візуалізацій, таких як графіки,

діаграми та теплові карти. Datadog також має систему сповіщень, що інформує користувачів про проблеми в реальному часі. Крім того, інструмент має можливості для аналізу логів, що дозволяє виявляти та виправляти проблеми в системі.

Datadog орієнтований на моніторинг IT-інфраструктури, його можливості аналізу даних є обмеженими для завдань, що виходять за межі моніторингу, має складну систему ціноутворення, що може бути не вигідним для невеликих компаній, а також потребує великих знань для правильної конфігурації.

Grafana (див. рис. 1.2).

Рисунок 1.2 – Дашбоард Grafana (рисунок створено самостійно)

Grafana є відкритим програмним забезпеченням для інтерактивної візуалізації даних. Платформа підтримує велику кількість джерел даних, таких як Prometheus, Elasticsearch, InfluxDB та інші, з яких можна збирати дані та візуалізувати їх за допомогою різних панелей. Grafana дозволяє створювати власні дашборди, використовувати готові шаблони, налаштовувати запити до баз даних та створювати інтерактивні візуалізації. Grafana надає також розширені

можливості для налаштування сповіщень, що допомагають вчасно реагувати на важливі події.

Grafana потребує налаштування та конфігурації, що може бути складним для нетехнічних користувачів. Вона орієнтована на візуалізацію вже існуючих даних, а не на їхній автоматизований аналіз. Також Grafana потребує налаштування джерел даних.

Google Data Studio (Looker Studio) (див. рис. 1.3).

Рисунок 1.3 – Дашбоард Google Data Studio (рисунок створено самостійно)

Google Data Studio є інструментом для створення інтерактивних дашбордів та звітів. Він інтегрується з багатьма джерелами даних Google, такими як Google Sheets, Google Analytics та Google BigQuery, а також дозволяє підключати інші джерела даних. Data Studio дозволяє створювати різні види візуалізацій, такі як

діаграми, таблиці та графіки. Інтерфейс є інтуїтивно зрозумілим та дозволяє створювати власні дашборди. Google Data Studio також підтримує можливість спільної роботи над звітами та можливість ділитись ними з іншими користувачами.

Google Data Studio (Looker Studio) має обмежені можливості для створення складних аналітичних функцій, а також залежність від екосистеми Google та платформ.

Microsoft Power BI (див. рис. 1.4).

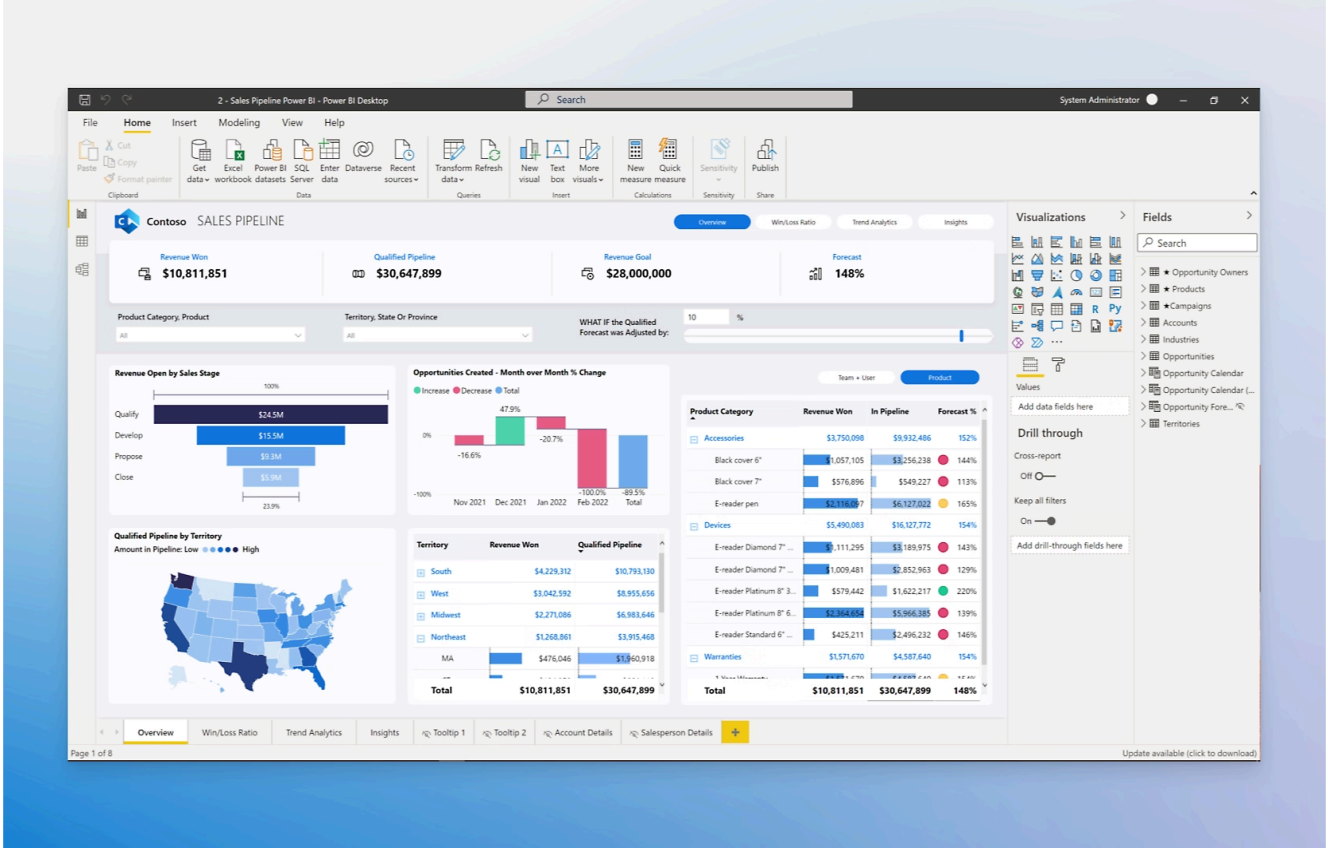

Рисунок 1.4 – Дашбоард Microsoft Power BI (рисунок створено самостійно)

Microsoft Power BI є потужною платформою для бізнес-аналітики, що дозволяє користувачам збирати, аналізувати та візуалізувати дані з різних джерел, створювати інтерактивні дашборди та звіти. Power BI має вбудовані можливості для моделювання даних та аналізу. Інструмент дозволяє інтегрувати різні джерела даних, такі як бази даних, файли та хмарні сервіси. Power BI також підтримує

можливість аналізу даних у реальному часі та можливість створення мобільних дашбордів.

Microsoft Power BI є комерційним продуктом, складний для вивчення, вимагає великої кількості ресурсів для обробки великих масивів даних, може мати обмеження для складних аналітичних задач.

Tableau (див. рис. 1.5).

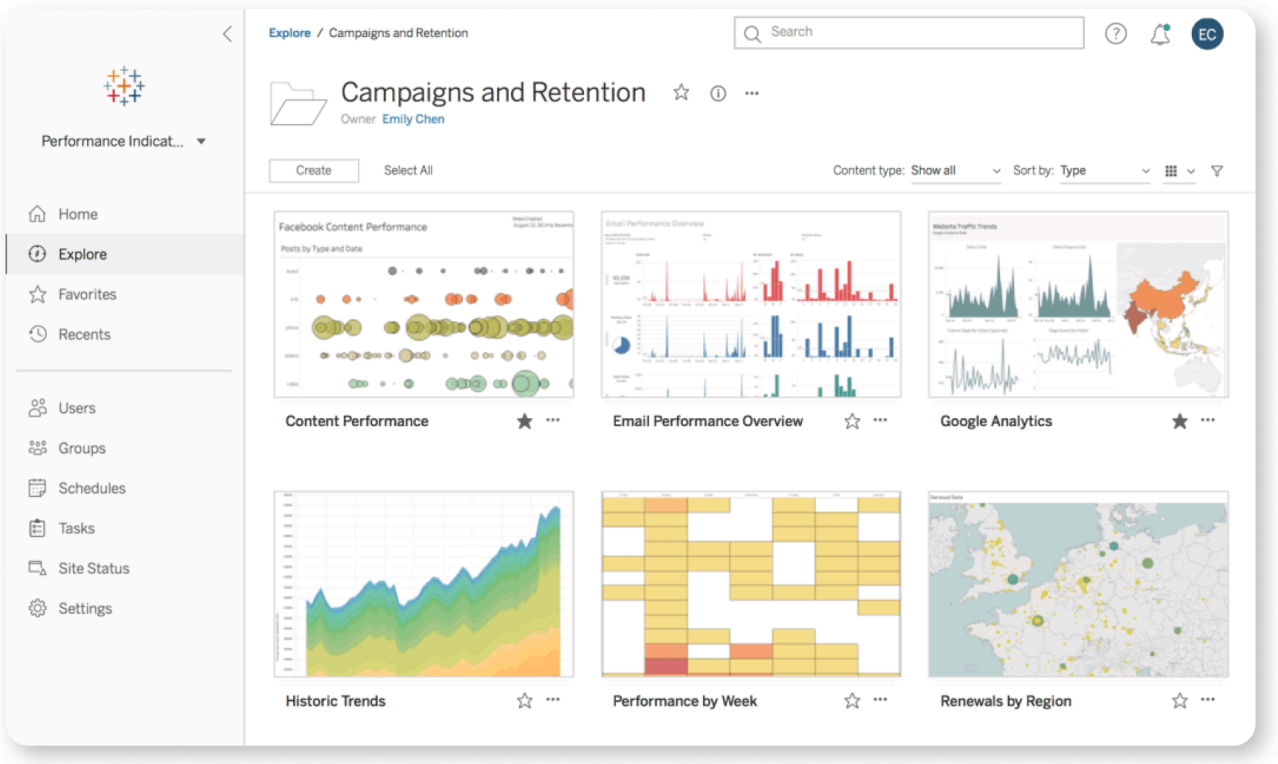

Рисунок 1.5 – Дашбоард Tableau (рисунок створено самостійно)

Tableau є потужним інструментом для інтерактивної візуалізації даних. Він дозволяє створювати широкий спектр різних візуалізацій, від базових графіків до складних карт та діаграм. Tableau підтримує інтеграцію з різними джерелами даних, включаючи бази даних, файли та хмарні сервіси. Інструмент має інтуїтивний інтерфейс для створення та кастомізації дашбордів та звітів. Tableau також надає можливості для виконання складного аналізу даних та створення розширених візуалізацій.

Tableau є дорогим комерційним продуктом, який вимагає ліцензій та може мати обмеження для роботи з великими обсягами даних. Також Tableau має довгий процес освоєння та вимагає спеціальних знань.

1.2 Підходи до аналізу даних та науково-технічна проблематика

У сучасному аналізі даних застосовуються різні методи: від традиційної статистики до нейромережевих моделей і гібридних підходів. Вибір методів значною мірою залежить від типу даних, задачі користувача та вимог до автоматизації.

Статистичні методи залишаються фундаментом для побудови гіпотез та аналізу залежностей, однак у сучасних умовах часто не дають змоги працювати з великими обсягами неструктурованих даних. У таких випадках застосовуються методи машинного навчання та автоматизовані інструменти, як-от AutoML, які дозволяють зменшити час та людське втручання в процес побудови моделей [3].

Значного поширення набувають гібридні підходи, які поєднують переваги статистичних моделей та глибокого навчання. Згідно з дослідженням [3], така інтеграція забезпечує кращу інтерпретованість моделей при збереженні високої точності.

Велика увага приділяється використанню великих мовних моделей (LLM) для задач аналізу даних, зокрема в частині генерації SQL-запитів, виявлення аномалій, автоматичної класифікації, пояснення інсайтів і побудови візуалізацій [4]. Застосування LLM для створення інтерфейсів "natural language to visualization" є новим напрямом, який поєднує генеративний ШІ, семантичний аналіз і автоматизовану побудову графіків [4].

1.3 Виявлення протиріч у наукових і експериментальних дослідженнях

Попри стрімкий розвиток технологій аналізу даних, у науковій літературі часто зустрічаються суперечливі результати. Наприклад, дослідження [5] зазначає, що висновки різних авторів щодо ефективності штучного інтелекту в обробці

великих даних часто не узгоджуються між собою, а частина моделей демонструє непослідовність результатів при повторних вимірах.

Подібні протиріччя фіксуються і в прикладних дослідженнях. У галузі біоінженерії було виявлено, що методики аналізу даних із застосуванням ML іноді дають протилежні висновки залежно від умов навчання моделі [6].

Також, у сфері генерації візуалізацій за допомогою LLM виявлено проблеми з відтворюваністю результатів та точністю. Згідно з [4], навіть найновіші моделі можуть генерувати некоректні або непрацездатні графіки, що ставить під сумнів їх повну автоматизацію без етапу валідації результатів.

1.4 Постановка науково-технічної задачі

Враховуючи виявлені протиріччя, сформульовано наступну науково-технічну задачу - розробити інструмент автоматизованого аналізу даних із застосуванням LLM, що забезпечує:

– підтримку структурованих (CSV, JSON) і текстових форматів даних;

– генерацію інсайтів і візуалізацій за допомогою природної мови;

– вбудовану перевірку достовірності результатів;

– порівняльний аналіз декількох LLM з використанням об’єктивних метрик точності, вартості та часу відгуку.

Оскільки LLM схильні до галюцинацій, необхідна додаткова валідація їхніх результатів, зокрема через впровадження контрольних сценаріїв та системи підрахунку точності (accuracy, BLEU, F1-score тощо). Також важливо враховувати різницю у вартості обчислень при використанні хмарних моделей, таких як GPT-4 чи Claude 3, що має значення для практичної реалізації [4].

2 ОГЛЯД Й АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

2.1 Огляд основних джерел

Метою цього розділу є надання детального огляду наукової літератури, що стосується розробки системи візуалізації та аналізу даних з використанням великих мовних моделей (LLM). Ми систематизуємо наявні знання, вивчимо ключові аспекти, такі як принципи роботи LLM, можливості їхнього застосування в аналізі даних, порівняння LLM-моделей (Gemini, GPT, Llama), методи візуалізації даних на JavaScript, клієнт-серверну архітектуру ПЗ, методи аналізу даних, зберігання та обробку великих даних, масштабування баз даних та мікросервісну архітектуру.

Для формування цього аналізу ми використовували наступні критерії для відбору джерел:

Авторитетність: Публікації у відомих рецензованих наукових журналах, виданнях престижних університетів та наукових організацій, матеріали авторитетних наукових конференцій.

Актуальність: Включення джерел, виданих в останні 5-10 років, щоб забезпечити актуальне бачення сучасних тенденцій і технологій.

Об'єктивність: Відбір наукових досліджень, що ґрунтуються на емпіричних даних або глибоких теоретичних розробках.

Достовірність: Відбір джерел, що мають чіткі посилання на методи дослідження, зібрані дані та проведені аналізи.

Релевантність: Вибір публікацій, що безпосередньо пов'язані з темами LLM, аналізу даних, візуалізації, архітектури ПЗ, і взаємодії між ними.

Посилання: Всі джерела мають чіткі посилання для можливості перевірки.

Для пошуку використовували бази даних Google Scholar та інші відкриті джерела, щоб знайти необхідну інформацію.

2.2 Аналіз літератури

2.2.1. Великі мовні моделі (LLM): Принципи роботи та можливості застосування.

Принципи роботи LLM LLM, такі як Gemini, GPT та Llama, базуються на архітектурі трансформерів, яка була представлена в статті "Attention is All You Need" (Vaswani et al., 2017). Трансформери відрізняються від попередніх архітектур нейронних мереж тим, що вони використовують механізми уваги (attention mechanisms), що дозволяють моделі обробляти послідовності даних паралельно, зосереджуючись на важливих частинах вхідної інформації. Механізми уваги дозволяють LLM створювати ваги, які показують, які частини вхідних даних є більш важливими для розуміння контексту. Ці ваги визначаються динамічно, що робить модель ефективною для обробки різноманітних послідовностей. Навчання LLM відбувається на величезних масивах текстових даних, що дозволяє їм розпізнавати закономірності, структуру та семантику мови. Це навчання відбувається безпосередньо на текстових даних, що робить модель адаптивною для різноманітних завдань [7].

Відповідно до Whitaker [8], ці моделі вивчають контекстні залежності через масковане або причинно-умовне мовне моделювання (Masked/Causal Language Modeling). Механізм уваги дозволяє моделі динамічно фокусуватись на важливих частинах вхідної інформації, забезпечуючи адаптивність до різноманітних задач. Дослідники зазначають, що збільшення розміру моделей покращує продуктивність, але супроводжується зменшенням приросту ефективності та зростанням обчислювальних витрат [8]. Це формує наукову задачу: знайти оптимальний баланс між розміром моделі, точністю та енергоефективністю, з урахуванням типу даних (структурованих і неструктурованих) і задач аналізу.

LLM можуть застосовуватися для аналізу даних у різних форматах, включаючи текст, структуровані дані та навіть мультимедійні дані. Вони можуть автоматизувати процеси, такі як вилучення інформації, виявлення залежностей, створення резюме, генерація звітів та переклад. Одним з ключових застосувань є

автоматизація аналізу текстів, де LLM можуть виявляти основні теми, настрій, та зв'язки між різними частинами тексту. Це особливо корисно для аналізу відгуків, документів та інших текстових масивів. LLM можуть бути використані для аналізу структурованих даних, де вони можуть виявляти закономірності, робити прогнози та надавати рекомендації. Такий аналіз може використовуватись в бізнесі, фінансах, наукових дослідженнях та багатьох інших областях [9].

2.2.2 Порівняння існуючих LLM моделей: Gemini, GPT, Llama

Gemini, розроблена Google, відзначається своєю мультимодальною архітектурою, що дозволяє моделі обробляти не тільки текст, але й інші види даних, такі як зображення, аудіо та відео. Це робить Gemini універсальним інструментом для різних завдань, де потрібно розуміти інформацію з різних джерел. Gemini використовує передові алгоритми для розуміння контексту та генерації точних відповідей, що робить її перспективною для складних завдань. Модель доступна у різних версіях (Ultra, Pro, Nano), що дозволяє її адаптувати під різні вимоги ресурсів та складності задач. Gemini Ultra - найпотужніша версія моделі, призначена для найскладніших завдань, таких як комплексний аналіз, генерація креативного контенту та розв'язування специфічних задач. Gemini Pro - версія, яка є збалансованою між точністю та продуктивністю, підходить для широкого кола завдань. Gemini Nano - оптимізована версія для роботи на мобільних пристроях та для завдань, що вимагають меншого обчислювального ресурсу [10].

Моделі GPT, розроблені OpenAI, є одними з найпопулярніших мовних моделей. GPT-4 є наступною ітерацією цієї моделі. Моделі GPT спеціалізуються на обробці та генерації тексту. GPT-4 має більший обсяг параметрів та більш широкі можливості для розуміння контексту. GPT має високу здатність до генерації природної мови, що робить її привабливою для завдань, таких як створення тексту, переклад, аналіз документів та відповіді на запитання. GPT є ефективним інструментом для обробки текстових даних [5].

Llama, розроблена Meta, є відкритою мовною моделлю, яка розповсюджується з відкритим кодом, що робить її доступною для широкого кола розробників та дослідників. Llama розроблена з акцентом на ефективність та продуктивність, що робить її ефективною для різних завдань обробки природної мови. Llama має кілька версій, кожна з яких оптимізована під різні потреби. Модель використовує ефективні методи навчання та архітектуру для зменшення обчислювальних витрат без значної втрати точност [11].

Хоча публічні архітектури цих моделей не завжди доступні, низка незалежних досліджень показала, що Claude 3 Opus іноді перевершує GPT-4 і Gemini Pro в задачах логічного висновування, аналізу даних і генерації SQL-запитів [12]. Модель Gemini вирізняється мультимодальністю й підтримкою довгого контексту (до 1M токенів у Gemini 1.5 Pro), що корисно для аналітики на великих документах. GPT-4 Turbo забезпечує високу стабільність і загальну універсальність, проте в деяких спеціалізованих задачах поступається Claude [12]. На відміну від пропрієтарних моделей, Llama 3 хоча й поступається в точності, зате є відкритою та доступною для локального розгортання. Важливо підкреслити, що результати експериментів суттєво залежать від типу даних і задач, а також від інженерії промптів.

2.2.3 Візуалізація даних за допомогою JavaScript

JavaScript є популярною мовою програмування для розробки інтерфейсів користувача, включаючи інтерактивні візуалізації. Існує багато бібліотек, що надають потужні інструменти для візуалізації даних. D3.js (Data-Driven Documents) є низькорівневою бібліотекою, яка надає максимальну гнучкість та контроль над візуалізаціями. Вона дозволяє створювати будь-які види графіків, діаграм та інтерактивних елементів, використовуючи SVG та HTML5. D3.js підходить для користувачів, які потребують максимальну гнучкість та контроль над візуалізаціями, але вимагає значних зусиль для освоєння. Chart.js є високо-рівневою бібліотекою, яка спрощує створення типових графіків та діаграм, таких як лінійні, стовпчасті, кругові діаграми. Chart.js є легкою у використанні та

підходить для швидкої візуалізації. Chart.js підходить для тих, хто хоче швидко створити стандартні візуалізації, без великих зусиль для налаштування. Plotly.js є бібліотекою для створення інтерактивних візуалізацій з можливістю масштабування, панорамування та інтерактивного перегляду. Plotly підтримує широкий спектр візуалізацій, включаючи графіки, діаграми, 3D-графіки та карти. Plotly підходить для користувачів, яким потрібні інтерактивні візуалізації та можливість дослідження даних [13].

2.2.4 Методи візуалізації даних з використанням LLM

Згідно з дослідженням Vázquez et al. [14], LLM, зокрема GPT-4, здатні автоматично генерувати код для візуалізацій (напр., з використанням бібліотек Matplotlib або Plotly) за запитами природною мовою. У більшості випадків результати є коректними, але іноді потребують ручного доопрацювання: наприклад, неправильне розташування осей, відсутність підписів або стилістичні неточності. Інтеграція LLM у аналітичні системи, як-от у проєкті Spiegel et al., дозволяє генерувати як текстові інсайти, так і графіки, підвищуючи ефективність аналізу даних у складних доменах, зокрема в медицині [15].

2.2.5 Методи аналізу даних

Для аналізу даних LLM можуть використовувати методи обробки природної мови (NLP), які дозволяють їм розуміти структуру, контекст та семантику тексту. NLP включає методи, такі як токенізація, аналіз контексту, виявлення ключових слів, теми та вилучення інформації. LLM також можуть застосовувати методи машинного навчання для виявлення закономірностей у структурованих і неструктурованих даних. Це включає методи кластеризації, класифікації та прогнозування. Методи аналізу даних у контексті LLM поєднують обробку природної мови та методи машинного навчання для створення складних аналітичних систем [16].

2.2.6 Зберігання та обробка великих даних

Зберігання великих даних вимагає спеціальних підходів, враховуючи об'єми та швидкість їхньої обробки. Традиційні реляційні бази даних часто не підходять для таких задач, тому використовуються NoSQL бази даних. NoSQL бази даних, такі як MongoDB, Cassandra, та Redis, розроблені для обробки великих обсягів неструктурованих і напівструктурованих даних, та для масштабування на великі кластери серверів. Іншим підходом є використання хмарних рішень для зберігання даних, таких як Amazon S3, Google Cloud Storage та Azure Blob Storage. Ці рішення забезпечують масштабованість, гнучкість та надійність.

Обробка великих даних вимагає використання спеціальних технологій, які дозволяють паралельно обробляти дані на великих кластерах комп'ютерів. Такі технології, як Apache Hadoop та Apache Spark, дозволяють розподіляти обчислення між вузлами кластеру, що забезпечує масштабованість та ефективність. Apache Spark є більш сучасним інструментом для обробки великих даних, що надає можливості для обробки даних у реальному часі, а також для машинного навчання та графічної обробки [17].

2.2.7 Аналіз структурованих та неструктурованих даних із LLM

LLM демонструють високу ефективність у перетворенні структурованих таблиць (CSV, JSON) на семантичні представлення та SQL-запити. Наприклад, Anthony et al. [18] зазначають, що при правильній подачі таблиці (наприклад, через “structured prompting”) GPT-4 досягає значно вищої точності при генерації запитів або узагальнень. Водночас у роботі Anderson et al. [19] описано можливість аналізу великих масивів неструктурованих даних - PDF, наукових статей, технічних звітів - із витяганням фактів та структурованої інформації. Це демонструє перевагу LLM перед класичними системами пошуку та статистичного аналізу, але вимагає додаткових механізмів перевірки на “галюцинації” та контроль якості відповідей.

2.2.8 Стратегії prompt engineering

За спостереженнями Amatriain [20], структура запиту суттєво впливає на якість відповіді LLM. Застосування методів, таких як Chain-of-Thought prompting, few-shot learning, self-reflection, дозволяє покращити точність відповідей у задачах логічного висновування, SQL-генерації та інтерпретації даних. Наприклад, запит "Проаналізуй цей датасет крок за кроком і побудуй гістограму" забезпечує більш логічну послідовність, ніж просто "Зроби графік". Використання чітких форматів для відповідей (JSON-формат, таблиця) знижує ризик неоднозначностей. Однак дослідники відзначають, що надмірно складні промпти можуть заплутати модель, що формує новий виклик у пошуку балансу між формальністю та простотою [20].

2.3 Оцінка актуальності та новизни

Аналіз літературних джерел підтверджує високу актуальність використання великих мовних моделей для автоматизації аналізу даних та створення інтерактивних візуалізацій. Поєднання можливостей LLM, таких як розуміння природної мови, обробка контексту через механізми уваги та аналіз як структурованих, так і неструктурованих даних, з сучасними JavaScript-бібліотеками для візуалізації відкриває нові можливості для створення інтуїтивно зрозумілих та ефективних аналітичних систем [1].

Порівняльний аналіз існуючих LLM-моделей показує, що кожна з них має унікальні характеристики: Gemini вирізняється мультимодальністю та здатністю обробляти довгий контекст, GPT-4 забезпечує високу стабільність та універсальність, Claude 3 Opus демонструє перевагу в логічних висновках та генерації SQL-запитів, а Llama 3 пропонує відкритість та можливість локального розгортання. Ці особливості необхідно враховувати при виборі моделі для конкретних завдань аналітичної системи.

Дослідження в області prompt engineering демонструють критичну важливість структурування запитів для отримання якісних результатів. Методи

Chain-of-Thought prompting та few-shot learning значно покращують точність відповідей, особливо при генерації коду для візуалізацій та SQL-запитів. Водночас виявлено проблему "галюцинацій" LLM, що вимагає додаткових механізмів контролю якості.

Аналіз JavaScript-бібліотек для візуалізації показує різноманітність підходів: D3.js надає максимальну гнучкість для складних візуалізацій, Chart.js забезпечує швидке створення стандартних графіків, а Plotly.js поєднує інтерактивність з широким спектром типів візуалізацій. Інтеграція цих інструментів з LLM дозволяє автоматично генерувати код візуалізацій за запитами природною мовою.

Технології обробки великих даних, такі як NoSQL бази даних (MongoDB, Cassandra), хмарні рішення (Amazon S3, Google Cloud Storage) та системи розподіленої обробки (Apache Spark), є необхідними для забезпечення масштабованості та ефективності аналітичних систем, що працюють з LLM [2].

На основі проведеного аналізу можна зробити наступні висновки:

– LLM є потужним інструментом для автоматизації аналізу даних, здатним обробляти як структуровані, так і неструктуровані дані;
– Gemini, GPT, Claude та Llama є перспективними моделями з різними перевагами: мультимодальність, стабільність, логічне мислення та відкритість відповідно;
– методи prompt engineering критично важливі для отримання якісних результатів від LLM;
– технології обробки великих даних необхідні для забезпечення продуктивності системи.

Прогалини та напрямки подальших досліджень:

– потребує вивчення оптимальні методи інтеграції різних LLM у архітектуру аналітичної системи з урахуванням балансу між точністю, швидкодією та ресурсними витратами;

– необхідно розробити ефективні стратегії боротьби з "галюцинаціями" LLM та механізми верифікації згенерованих результатів;

– потребує дослідження порівняльна ефективність різних JavaScript-бібліотек візуалізації в контексті автоматичної генерації коду через LLM;

– необхідно визначити оптимальні архітектурні рішення для інтеграції LLM з системами обробки великих даних;

– потребують розробки стандартизовані підходи до prompt engineering для задач аналізу даних та візуалізації.

3 ПОСТАНОВКА ЗАДАЧІ

Даний розділ присвячений формулюванню задачі науково-дослідницької роботи, яка полягає в дослідженні теоретичних основ системи для візуалізації та аналізу даних з використанням великих мовних моделей (LLM). Метою проекту є створення прототипу та порівняння моделей, яка дозволить користувачам ефективно обробляти великі обсяги даних, виявляти закономірності та інсайти, а також візуалізувати результати за допомогою LLM.

3.1 Опис предметної області та мети дослідження

Предметна область охоплює дослідження застосування LLM у контексті аналізу даних, зокрема їхньої здатності виявляти закономірності, генерувати інсайти та автоматизувати створення візуалізацій. Мета полягає в тому, щоб:

- дослідити можливості LLM для аналізу структурованих і неструктурованих даних;
- розробити прототип системи, яка інтегрує різні LLM для виконання аналітичних задач;
- порівняти ефективність обраних моделей (GPT, Claude, LLama, Gemini) у контексті точності, швидкості обробки та якості візуалізацій;
- визначити оптимальні стратегії адаптації LLM до специфічних потреб аналізу даних.

Система має забезпечувати:

- гнучкість інтеграції LLM: можливість підключати різні моделі через стандартизовані інтерфейси;
- автоматизацію аналізу: використання LLM для генерації SQL-запитів, класифікації даних, виявлення аномалій;
- адаптивну візуалізацію: автоматичний підбір типів графіків на основі аналізу LLM та кастомізація звітів;
- порівняльну оцінку: метрики для аналізу точності, часу відгуку та якості інсайтів кожної моделі;

3.2 Формулювання задач дослідження

Для досягнення поставленої мети необхідно вирішити наступні задачі:

Дослідження можливостей LLM у аналізі даних:

– проаналізувати існуючі LLM (GPT, Gemini, Claude тощо) щодо їхньої здатності обробляти структуровані дані (CSV, JSON), текстові записи та генерувати корисні інсайти;
– визначити критерії порівняння моделей: точність аналізу, швидкість відповіді, вартість використання API.

Розробка прототипу для інтеграції та тестування LLM:

– створити модульну систему, де кожна LLM інтегрується через уніфікований API;
– реалізувати механізми для автоматизації тестових сценаріїв (наприклад, обробка стандартних датасетів).

Генерація інсайтів та візуалізація:

– дослідити, як LLM можуть рекомендувати типи графіків на основі аналізу даних (наприклад, гістограми для розподілу, теплові карти для кореляцій);
– розробити методи автоматичного формування текстових пояснень до візуалізацій.

Порівняльний аналіз ефективності LLM:

– провести A/B тестування моделей на різних типах даних.

Оптимізація взаємодії з LLM:

– дослідити вплив prompt engineering на точність результатів;
– розробити шаблони запитів, що максимізують корисність відповідей моделей

3.3 Обґрунтування вибору методів дослідження

Для вирішення поставлених задач використовуються кілька методів. По-перше, експериментальний аналіз, який включає тестування LLM на стандартних датасетах, таких як Titanic та Iris, для оцінки їхньої здатності до

класифікації, регресії та генерації SQL-запитів. По-друге, порівняльне тестування, що передбачає вимірювання часу відгуку та точності моделей в однакових умовах. Для об'єктивної оцінки якості застосовуються кількісні метрики, зокрема F1-score, точність (accuracy) та BLEU для аналізу текстових інсайтів. Додатково проводиться якісний аналіз, під час якого оцінюється відповідність візуалізацій очікуваним результатам.

Щодо вибору технологій, для бекенду використовується Nodejs з бібліотеками vercel/ai, що забезпечують інтеграцію LLM. Для візуалізації застосовуються Nextjs та Recharts, що дозволяють створювати інтерактивні дашборди. Дані зберігаються в PostgreSQL, що забезпечує надійне збереження результатів аналізу.

3.4 Обмеження дослідження

Дослідження обмежене аналізом 3 LLM, таких як GPT-4, Claude 3 і Gemini Pro, через часові та ресурсні обмеження. Тестування проводиться на публічних датасетах, що може не повністю відображати реальні бізнес-сценарії, оскільки такі набори даних не завжди охоплюють специфічні галузеві кейси. Основна увага приділяється роботі з текстовими та структурованими даними, тоді як обробка мультимедійних форматів, таких як зображення чи аудіо, у межах цього дослідження не розглядається.

3.5 Необхідні ресурси

Дослідження передбачає доступ до API LLM, зокрема OpenAI, Anthropic і Mistral, що дозволяє тестувати та порівнювати різні моделі. Розробка ведеться в середовищі Nodejs із використанням Webstorm. Для візуалізації даних застосовуються React та Recharts що забезпечують зручний аналіз результатів. Для обчислювально-складних експериментів використовуються хмарні ресурси, зокрема Google Cloud та AWS, що дозволяють масштабувати обчислення за потреби.

4 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

У цьому розділі розглянуто архітектурні та методологічні основи використання великих мовних моделей (LLM) для аналізу даних. Ми досліджуємо особливості провідних моделей GPT-4/4o (OpenAI), Claude 3 (Anthropic) та Gemini Pro (Google) і їхню придатність для роботи з різними типами даних. Також проаналізовано підходи до обробки структурованих (таблиці, JSON) та неструктурованих даних (класифікація, NER, сентимент), роль промпт-інжинірингу (налаштування запитів) і обмеження моделей (обсяг контексту, точність, вартість, «галюцинації»). Додатково висвітлено проблему галюцинацій.

4.1 Архітектурні особливості LLM та їхній вплив на аналіз даних

Основу сучасних LLM, таких як GPT-4, Claude 3 та Llama 3, складають трансформери - нейромережі, здатні обробляти послідовності даних з урахуванням контексту через механізми уваги (attention mechanisms). Ця архітектура дозволяє моделям:

- аналізувати зв’язки між далекими елементами у структурованих даних (наприклад, кореляції між колонками CSV);
- генерувати семантичні уявлення для неструктурованих текстових даних, що є ключовим для класифікації або виявлення тем;
- адаптуватися до різних мовних шаблонів, що критично для обробки технічних звітів, соціальних медіа або наукових статей.

Дослідження показують, що LLM з параметрами від 7B до 70B можуть ефективно виконувати задачі, які раніше вимагали спеціалізованих моделей, наприклад:

- генерація SQL-запитів на основі природномовних інструкцій (точність до 85% на датасетах WikiSQL);
- виявлення аномалій у часових рядах за допомогою контекстного аналізу;
- автоматична категорізація даних (наприклад, розподіл товарів за категоріями на основі описів);

4.2 Обробка структурованих даних

Структуровані дані (таблиці, JSON, CSV) складаються з чітко визначених полів і значень. LLM не мають вбудованого «розуміння» табличної структури, тому зазвичай перед початком роботи дані переводять у текстовий формат (серіалізують). Вибір формату вводу критично впливає на результати. Дослідження показують, що при різних схемах представлення таблиць LLM демонструють неоднакову ефективність. Так, серіалізація рядок-за-рядком або стовпчик-за-стовпчиком з різними спеціальними маркерами (наприклад, TAPEX, TABBIE) може по-різному відображати структуру. Хоча LLM здатні виокремлювати базову структуру таблиці, у тривіальних задачах (наприклад, підрахунок рядків/стовпців) вони часто помиляються. При цьому потрібні спеціалізовані підказки, рекомендують зразки розмітки в запиті, розбивку даних на блоки, комбінований підхід (zero-shot та few-shot) та інші техніки, що значно підвищують якість «розуміння» таблиць. Для генерації структурованого виводу (наприклад, JSON) також важлива оптимізація промтів. Так, експеримент з GPT-4o показав: використання різних стилів підказки (JSON, YAML, гібрид CSV/префікс) дає різні результати за точністю і вартістю токенів.

З’ясовано, що JSON-підхід забезпечує найвищу точність для складних даних (точніше зберігає всі атрибути), тоді як YAML забезпечує баланс між читабельністю й ефективністю, а гібрид CSV/префікс оптимізує швидкодію і вартість при роботі з плоскими таблицями.

Ці висновки підкреслюють важливість «структурованого запитування»: сьогодні багато LLM (зокрема GPT-4 Turbo) мають спеціальні режими виводу у форматі JSON або можливість виклику функцій (function calling), щоб гарантувати коректне експортоване дерево даних. Таким чином, робота з табличними чи JSON-даними з LLM вимагає налаштування схеми запитів - правильний вибір формату вводу (серіалізації) і структури відповіді доводить свою ефективність

4.3 Аналіз текстових даних

LLM демонструють потужні результати у різних задачах обробки тексту. Класифікація тексту (наприклад, визначення теми документу чи настрою) може виконуватися через few-shot або zero-shot запити. Наприклад, моделі GPT та Claude успішно вирішують задачі аналізу сентименту – зазвичай із точністю на рівні чи вищою, ніж традиційні архітектури. У фінансовому аналізі дослідники показали, що GPT-4o, оптимізована через добре спроєктовані промти, перевершує спеціалізований FinBERT майже на 10% у задачах сентимент-аналізу новин [21].

Розпізнавання іменованих сутностей (NER) є складнішим для загальних LLM, оскільки це завдання послідовного маркування. У низці досліджень виявлено великий розрив у точності: у експерименті Zeghidi & Moncla (2024) GPT-4o досяг F1 ≈0.70, GPT-4 ≈0.67, а GPT-3.5 – лише ≈0.50 при невеликій кількості прикладів, тоді як тонко налаштований BERT для NER набрав F1 ≈0.93. Це означає, що базовий LLM не досягає кваліфікації вузькоспеціалізованих моделей у строгих NER-завданнях (виявлення кожного токена), хоча великі моделі здатні адаптуватися до нових типів сутностей за небагатьма зразками.

Загалом, LLM добре справляються з класифікацією та аналітичними завданнями у широкому діапазоні текстів (новини, твори, наукові статті), особливо якщо використовувати вбудовані механізми few-shot prompting. Проте специфічні завдання (NER, fine-grained sentiment тощо) часто вимагають додаткової обробки чи зовнішніх моделей. Наприклад, аналізування юридичних текстів або специфічних доменів може потребувати донавчання чи поєднання з іншими інструментами.

4.4 Порівняльні характеристики LLM

Порівняння GPT-4/4o, Claude 3 та Gemini за різними метриками демонструє комбінацію унікальних сильних сторін. GPT-4o mini (OpenAI) орієнтована на економічність: вона працює з контекстом до 128K токенів та демонструє високі оцінки за академічними бенчмарками. Наприклад, GPT-4o mini набрав 82.0% по MMLU (текстове розуміння) проти 77.9% у Gemini Flash та 73.8% у Claude Haiku.

У арифметичних і програмувальних тестах (MGSM, HumanEval) GPT-4o mini також випереджає конкурентів: 87.0% проти 75.5% і 71.7% у математичному MGSM та 87.2% проти 71.5% і 75.9% в HumanEval. Це показує, що GPT-моделі забезпечують відмінну якість загального розуміння та розв’язування задач.

Claude 3 орієнтована на надійність і мультидисциплінарність. Її найбільш потужна версія Opus демонструє провідні результати серед LLM у широкому спектрі завдань («веде фронт загального інтелекту»). Унікальністю Claude є увага до обробки візуальної інформації: всі моделі Claude 3 добре розпізнають діаграми та зображення.

Підмодель Haiku, наприклад, змодельована так, щоб бути максимально швидкою і доступною: вона здатна за кілька секунд переглядати великий науковий документ (~10K токенів з графіками). Sonnet компромісно поєднує високу інтелектуальність з подвійною швидкістю попередника, а Opus - максимальну точність з продуктивністю, близькою до Claude 2. Anthropic також підкреслює зростання точності: у випадкових складних тестах Opus показує удвічі кращу точність порівняно з попереднім Claude 2.1 (значно рідше помиляється).

Gemini (Google) акцентує на масштабованості та довгих контекстах. Gemini 1.5 Pro забезпечує продуктивність, порівнянну з Ultra при меншій обчислювальній вартості, і, як відзначено вище, підтримує найбільше контекстне вікно (до мільйона токенів). Тобто Gemini ідеально підходить для завдань, де необхідно «проковтнути» велику кількість інформації за один раз (наприклад, кодову базу чи об’ємний документ).

Важливим фактором є вартість використання. Згідно з офіційними даними, GPT-4o mini коштує близько 0,15 дол. за мільйон вхідних токенів і 0,60 дол. за мільйон вихідних, що удесятеро нижче порівняно з попередніми «фронтірними» моделями (і більш ніж на 60% дешевше за GPT-3.5 Turbo). Anthropic і Google не розкривають точні тарифи публічно, але традиційно їхні моделі (наприклад, Claude 3 Haiku) проєктують на мінімальні витрати для базових задач (висока швидкість при знижених витратах).

Таким чином, вибір конкретної моделі часто базується на балансі «швидкість-якість-ціна» та специфіці завдання: GPT-моделі сильні у загальному розумінні, Claude – у безпеці і роботі з мультиформатами, а Gemini – у роботі з надвеликими обсягами контексту.

4.5 Роль prompt engineering

Prompt engineering (налаштування запитів) є ключовим для підвищення продуктивності LLM. Сюди входять такі техніки: Zero-shot (лише опис завдання), Few-shot (подача прикладів зразків у запиті), Chain-of-Thought (CoT) (проміжні кроки міркувань), а також system prompts (надання інструкцій моделі щодо ролі або тону відповіді).

Дослідження показують, що додавання ланцюжка роздумів суттєво покращує результати у складних задачах. Наприклад, один лише CoT-вивід у промпті дозволив 540-мільярдній моделі GPT досягти state-of-the-art у задачі GSM8K (математичні приклади з доведеними рішеннями). При наявності декількох демонстраційних прикладів LLM «природно» починають виносити поетапні міркування, що значно підвищує точність відповіді.

Аналогічний ефект отримано при налаштуванні промтів у задачах NLP. Систематичний підбір ітеративно удосконалених промтів суттєво покращив точність GPT-4o у класифікації сентименту – модель перевершила FinBERT практично завдяки промпт-дизайну.

Стиль промту при генерації структурованих даних може змінювати баланс між точністю та вартістю: JSON-команди забезпечують найвищу точність для складних даних, але можуть бути «важчими» в плані токенів, тоді як YAML чи CSV-стилізовані промти економлять ресурси за рахунок трохи меншої точності.

Отже, роль prompt engineering є вирішальною: правильна формулювання завдання, добре підібрані зразки few-shot та стратегія розбиття міркувань (CoT) часто призводять до значного підвищення точності LLM. Натомість невдалий промт може різко знизити якість відповіді або призвести до «галюцинацій». З цієї причини розробникам рекомендують витрачати час на експерименти з

розгалуженнями промтів: зміна тональності (через system prompt), кількості прикладів і демонстраційних кроків тощо.

4.6 Обмеження LLM

Великі мовні моделі мають кілька суттєвих обмежень. По-перше, розмір контекстного вікна обмежений. GPT-4 зазвичай підтримує контекст до ~32К токенів (Turbo-версія), GPT-4o mini - до 128К, Claude 3 - базово 200К (з перспективою 1 млн токенів для обраних користувачів), а Gemini 1.5 Pro - до 1 млн токенів. Через ці обмеження модель не може обробити нескінченний потік даних; для великих завдань необхідно нарізати вхід або використовувати техніки «вікон» чи зовнішньої пам'яті.

По-друге, достовірність відповідей не гарантується. Навіть передові LLM можуть «галюціювати» - генерувати переконливу, але невірну інформацію. GPT-4 «не є повністю надійним (може піддаватися «галюцинаціям»)» і має обмежене контекстне вікно. Подібні застереження стосуються всіх LLM: вони іноді винаходять факти, які не підтверджені жодними джерелами. Галюцинація визначається як вивід, що є «вигаданим і не обґрунтований ані в контексті, ані у світових знаннях». Щоб уникнути галюцинацій, модель має бути максимально фактичною і визнавати незнання, коли воно є.

По-третє, вартість та ресурсоємність. Навіть «менші» моделі потребують значних обчислювальних ресурсів для служби. Приклад: GPT-4o mini коштує 0,15 дол. за мільйон вхідних токенів і 0,60 дол. за мільйон вихідних. Для порівняння, це у 10–20 разів дешевше, ніж попередні флагманські моделі, але все одно означає значні витрати при постійному використанні. Anthropic і Google поки не публікують точних тарифів, але модель Haiku позиціонують саме як найекономічнішу у своєму класі. Таким чином, навіть якщо сама сервісна модель безкоштовна у вигляді хмарного API, реальні проєкти мають закладати значні бюджети на оплату викликів.

Нарешті, є операційні та етичні обмеження. LLM можуть відмовлятися відповідати на запити, що потрапляють під їхні вбудовані обмеження безпеки. До

того ж ці моделі нерозкриті («чорний ящик»), тому складно зрозуміти внутрішні причини помилок. Потребують уваги ризики упередженості (bias), конфіденційності (якщо моделі отримують приватні дані) і т. ін. У сумі ці фактори визначають межі застосовності LLM і мотивують пошук додаткових рішень.

4.7 Проблема галюцинацій та способи верифікації відповідей

«Галюцинація» (hallucination) у LLM - це явище, коли модель генерує неправдивий, вигаданий або неперевірений контент. Під цим терміном розуміють вивід, «не обґрунтований ані наданим контекстом, ані реальними знаннями». Виділяють два основні типи: контекстно-залежні й екзогенні галюцинації. Перші виникають, коли відповідь не узгоджується з поданим текстом, другі - коли модель дає твердження, яких нема у жодних навчальних даних.

Галюцинації особливо небезпечні для критичних доменів. Наприклад, ChatGPT/Gemini можуть вигадувати помилкові юридичні прецеденти чи неперевірені факти з новин, що потенційно може завдати шкоди (навіть ризику для життя у медичному контексті). Через це для відповідей LLM потрібно впроваджувати механізми контролю достовірності.

Серед популярних стратегій - Retrieval-Augmented Generation (RAG): модель запитує зовнішню базу знань (наприклад, документи чи векторне сховище), отримує релевантну інформацію і ґрунтує відповідь на реальних даних. Цей підхід розділяє «знання» і «міркування»: LLM формулює текст на основі достовірних джерел.

Інший метод - самодіагностика: LLM проситься повторити відповідь чи «перепитати себе» («second guess»), що іноді допомагає виявити суперечності. Також використовують ансамблювання (кілька моделей порівнюють відповіді між собою) та детектори галюцинацій. Статистичні методи (ентропія семантики) дозволяють виявляти «вигадки» у LLM. Такі метрики дозволяють оцінити невизначеність виводу та попередити потенційну хибу.

Заохочення чесності моделі - ще одна тактика. Щоб уникнути галюцинацій, модель має бути «фактичною і визнавати незнання, коли це доречно». Тобто

краще, коли LLM прямо говорить «Я не знаю», аніж вигадує. У практиці це реалізується через уточнюючі промти (наприклад, «Приведи джерела»), обмеження за довірою чи fine-tuning на фактичність. Деякі системи навіть почали впроваджувати функцію цитування джерел (у Claude 3 заплановано опцію показувати конкретні речення з довідкових матеріалів), щоб зробити вихід прозорішим.

В цілому, боротьба з галюцинаціями - активна область досліджень. В системному аналізі її вирішують поєднанням методів (RAG, post-filtering, entropy-моніторинг тощо). У практичному застосуванні розробникам рекомендують обов'язково верифікувати відповіді LLM - наприклад, перевіряти ключові факти через пошукові системи, використовувати дублювання запитів чи утиліти для перевірки опорних джерел.

Отже, архітектурні та функціональні особливості сучасних LLM (таких як GPT-4/4o, Claude 3, Gemini) забезпечують їм високий потенціал у задачах аналізу й візуалізації даних. З іншого боку, для надійного застосування потрібні ретельне промпт-інжиніринг, розуміння обмежень моделей та додаткові методи верифікації. У поєднанні з інструментами обробки структурованих даних і графічних представлень, LLM можуть стати ядром майбутніх систем автоматизованої аналітики.

5 СТВОРЕННЯ ПРОТОТИПУ І ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

Цей розділ присвячений розробці та експериментальному дослідженню прототипу системи аналізу даних на основі великих мовних моделей (LLM). У ньому детально розглядається процес створення веб-додатку, що дозволяє користувачам завантажувати структуровані дані, проводити їх автоматичний аналіз за допомогою різних LLM та отримувати результати у вигляді інтерактивних візуалізацій і текстових інсайтів. Розділ охоплює технічну архітектуру прототипу, його функціональні можливості, особливості взаємодії з LLM API та результати експериментальної перевірки ефективності різних моделей у контексті аналізу даних.

5.1. Загальна архітектура та технологічний стек прототипу

В рамках дослідження було розроблено веб-додаток для аналізу даних з використанням великих мовних моделей (LLM). Архітектура прототипу побудована за принципом модульності, що забезпечує гнучкість при інтеграції різних LLM та можливість їх порівняльного аналізу. Для реалізації проєкту було обрано сучасний технологічний стек, що включає:

- Next.js як основний фреймворк для розробки фронтенд та бекенд частин додатку, що забезпечує серверний рендеринг та API-маршрути;
- AI SDK для уніфікованої взаємодії з різними LLM (OpenAI, Anthropic, Google), що дозволяє легко переключатися між моделями без зміни архітектури;
- ShadcnUI для створення сучасного та інтуїтивного користувацького інтерфейсу;
- Recharts для побудови інтерактивних графіків та візуалізацій;
- Zod для валідації схем даних та забезпечення типобезпеки.

Така архітектура дозволяє забезпечити масштабованість системи, а також спрощує процес тестування та порівняння різних LLM у контексті аналізу даних.

5.2. Функціональні можливості прототипу

Розроблений прототип забезпечує повний цикл аналізу даних з використанням LLM, починаючи від завантаження даних і закінчуючи створенням інтерактивних візуалізацій та текстових інсайтів. Основні функціональні можливості системи включають:

5.2.1. Домашня сторінка та управління візуалізаціями

Домашня сторінка прототипу (див рис 6.1) відображає картки раніше створених візуалізацій, що містять базову інформацію: назву аналізу, дату створення, тип даних та використану модель. Користувач може переглянути наявні візуалізації, натиснувши на відповідну картку, що забезпечує швидкий доступ до результатів попередніх аналізів.

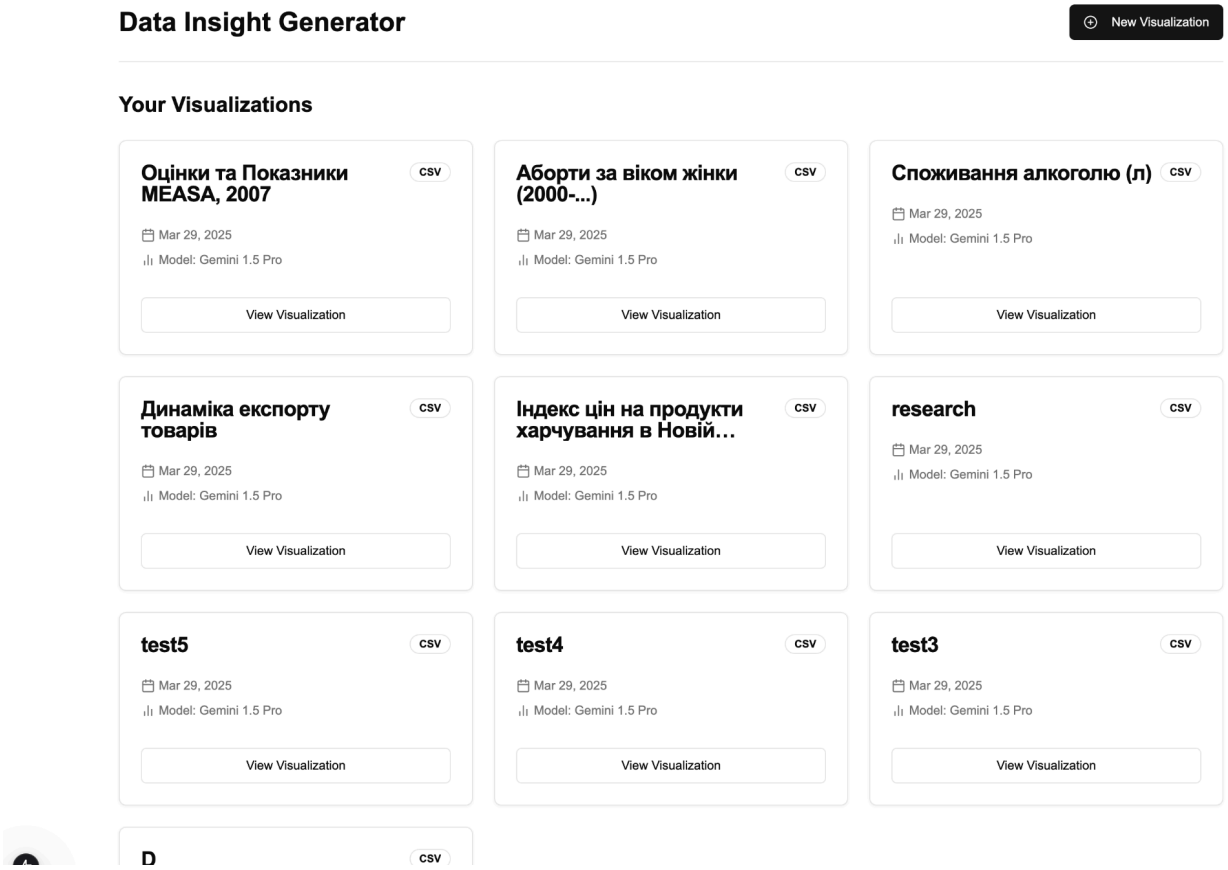

Рисунок 5.1 – Домашня сторінка прототипу (рисунок створено самостійно)

5.2.2. Створення нової візуалізації

Щоб створити нову візуалізацію, користувач натискає відповідну кнопку в шапці додатку. Після цього відкривається діалогове вікно з формою (див рис. 6.2), у якій потрібно завантажити файл даних у форматі JSON або CSV (з обмеженням до 10 МБ), обрати мовну модель для аналізу (GPT-4o від OpenAI, Claude 3 Opus від Anthropic або Gemini 1.5 Pro від Google) та вибрати мову результатів (українську або англійську).

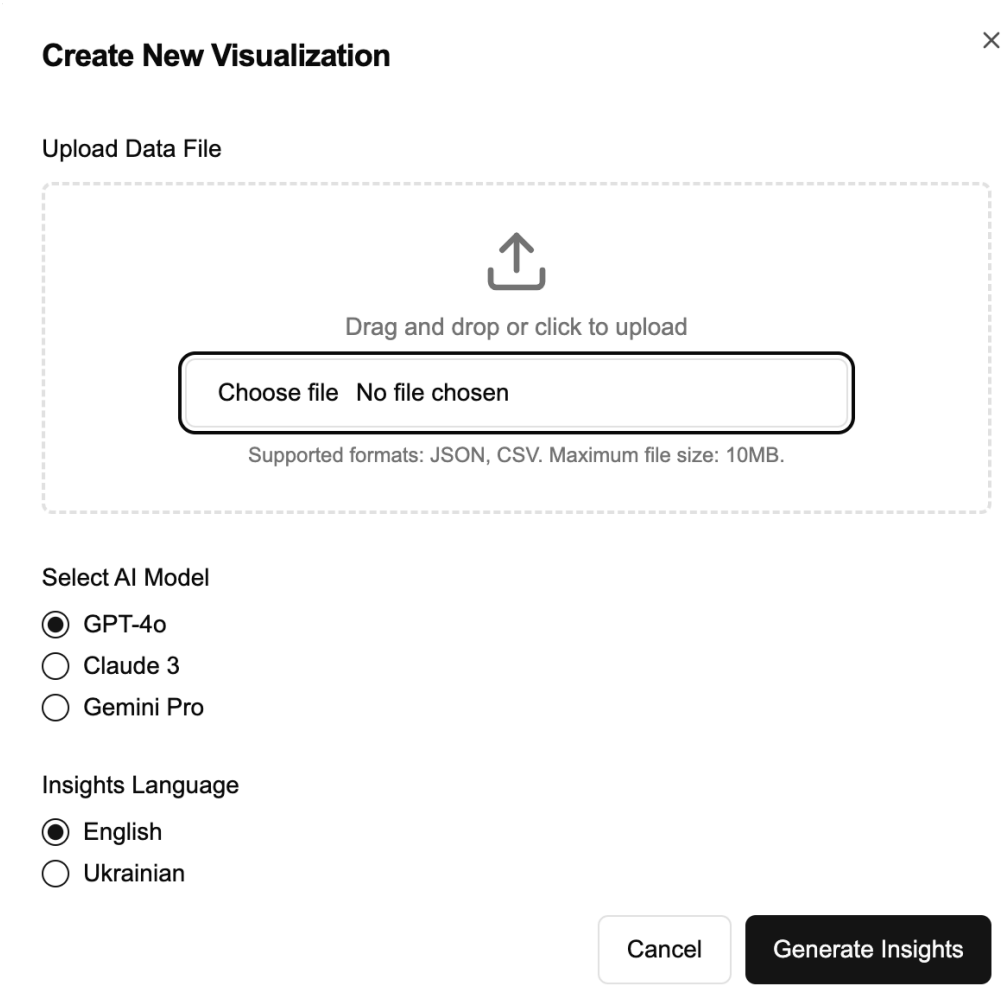

Рисунок 5.2 – Створення візуалізації (рисунок створено самостійно)

Після надсилання форми система зчитує та парсить дані, за необхідності скорочує їх, якщо перевищено ліміти токенів моделі, формує промпт з інструкціями щодо аналізу, надсилає запит до API вибраної моделі, зберігає

отримані результати у внутрішньому сховищі та перенаправляє користувача на сторінку з візуалізацією.

5.2.3. Сторінка візуалізації та аналізу даних

Сторінка візуалізації відображає результати аналізу даних у вигляді інтерактивних віджетів (див рис. 6.3) . У верхній частині сторінки розміщена інформаційна панель із заголовком аналізу, датою створення, використаною моделлю та індикатором, чи були дані скорочені через обмеження токенів.

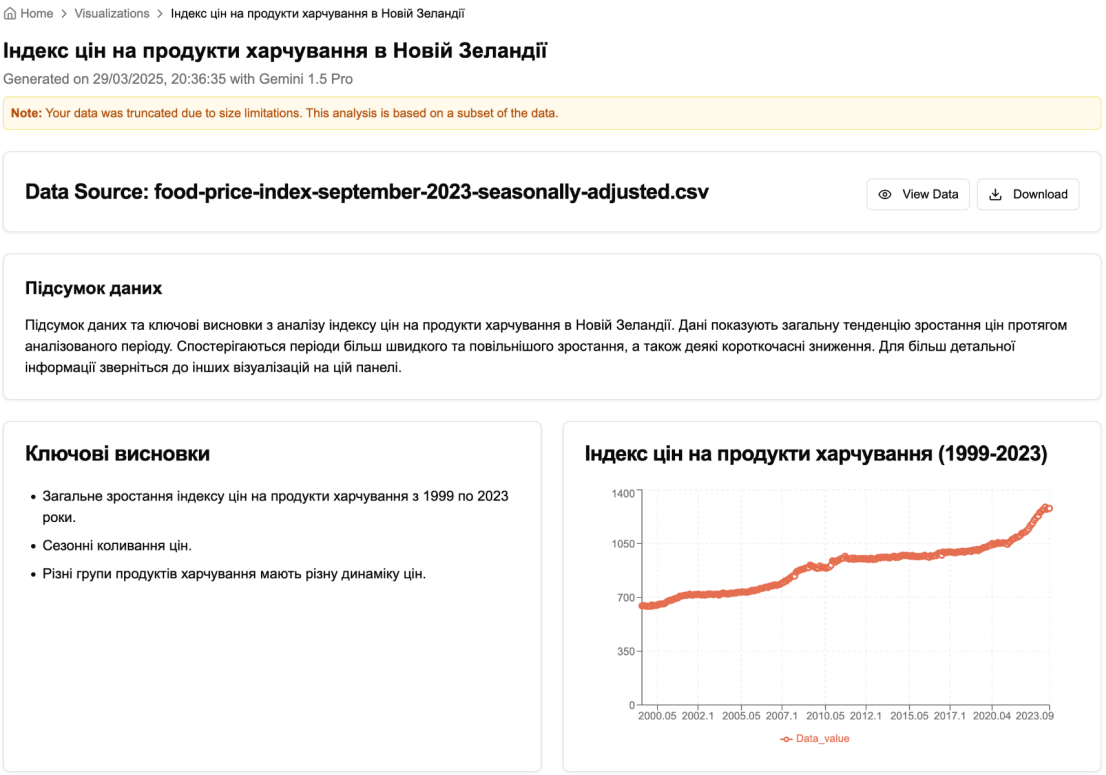

Рисунок 5.3 – Сторінка візуалізації (рисунок створено самостійно)

Основна частина сторінки містить динамічно згенеровані віджети, які можуть бути наступних типів:

– графіки (Chart Widget) – різні типи візуалізацій (стовпчикові, лінійні, кругові, площинні, точкові діаграми);
– таблиці (Table Widget) – структуровані дані з можливістю сортування;
– інсайти (Insights Widget) – текстові висновки у вигляді списку або параграфів;

– метрики (Metrics Widget) – ключові показники з індикаторами змін;
– загальний висновок (Summary Widget) – текстовий опис основних знахідок.

Кожен віджет супроводжується заголовком, описом та текстовими інсайтами, які пояснюють значення візуалізації. Додатково користувач має можливість переглянути вихідні дані або завантажити їх для подальшого використання.

5.3. Технічна реалізація взаємодії з LLM

Основною частиною прототипу є система взаємодії з LLM, яка реалізована за допомогою AI SDK та включає наступні ключові компоненти:

5.3.1. Обробка та підготовка даних

Система підтримує два основні формати даних: JSON та CSV. Для CSV-файлів використовується бібліотека csv-parse, яка конвертує текстові дані у структуровані об'єкти. Важливою частиною процесу є функція truncateData, яка забезпечує автоматичне скорочення даних у випадку, якщо їх обсяг перевищує максимальну кількість токенів, що підтримується моделлю (встановлено обмеження в 50 000 токенів або приблизно 200 000 символів).

5.3.2. Формування промптів

Для взаємодії з LLM система використовує два основні промпти:

Заголовковий промпт – короткий запит для генерації змістовного заголовка на основі перших 5 елементів даних

Промпт для аналізу даних, детальніше розглянемо його у наступному розділі.

Система використовує бібліотеку Zod для валідації відповідей від LLM, що забезпечує типобезпеку та гарантує, що згенеровані конфігурації віджетів відповідають очікуваній структурі.

5.3.3. Взаємодія з API моделей

Завдяки використанню AI SDK система підтримує уніфікований інтерфейс для взаємодії з різними LLM:

- OpenAI (GPT-4o) через пакет @ai-sdk/openai;
- Anthropic (Claude 3 Opus) через пакет @ai-sdk/anthropic;
- Google (Gemini 1.5 Pro) через пакет @ai-sdk/google;

Це дозволяє легко переключатися між різними провайдерами та моделями без необхідності змінювати код, що працює з API. Система використовує метод generateObject для отримання структурованих відповідей безпосередньо у форматі об'єктів JavaScript, що спрощує подальшу обробку та візуалізацію.

5.4. Система збереження та відображення результатів

Прототип включає модульну систему збереження даних, що дозволяє зберігати як вихідні дані (dataSource), так і результати аналізу (visualization). Кожна візуалізація отримує унікальний ідентифікатор UUID, що забезпечує можливість повторного доступу до результатів.

Важливою особливістю системи є механізм трансформації даних, який дозволяє динамічно перетворювати вихідні дані для кожного віджету. Це забезпечується через генерацію JavaScript-функцій (dataTransform), які виконуються на стороні клієнта під час рендерингу візуалізацій.

5.5. Результати експериментального дослідження

В ході експериментального дослідження прототип було протестовано на різних наборах даних, включаючи структуровані дані продажів, метрики користувацької активності, часові ряди та категоріальні дані. Проведені експерименти дозволили виявити як сильні сторони, так і обмеження різних LLM у контексті аналізу даних.

Попередні результати показують, що:

Claude 3 Opus демонструє найкращі результати в генерації змістовних текстових інсайтів та виборі найбільш доречних типів візуалізацій для різних наборів даних.

GPT-4o показує високу точність у математичних обчисленнях та генерації коректних функцій трансформації даних.

Gemini 1.5 Pro має переваги в швидкості обробки великих наборів даних, проте іноді генерує менш структуровані текстові описи.

Важливим спостереженням є те, що точність і корисність аналізу значно залежать від якості промптів та структури вихідних даних. Моделі демонструють вищу ефективність при роботі з добре структурованими даними та чіткими інструкціями.

Прототип успішно справляється із завданням автоматичного генерування різноманітних типів візуалізацій, включаючи графіки, таблиці та текстові інсайти, що підтверджує початкову гіпотезу дослідження про можливість використання LLM для аналізу даних та генерації корисних візуалізацій. Код прототипу розміщений у системі контролю версій GitHub [22].

6 ПОБУДОВА ПРОМПТУ

У рамках дослідження можливостей великих мовних моделей для аналізу та візуалізації даних критичним компонентом є розробка ефективного промпту та відповідної структури відповіді. Правильно сформований промпт дозволяє максимально використати аналітичні можливості LLM та отримати структуровані відповіді, придатні для автоматичної обробки та відображення. У цьому розділі розглянемо архітектуру промпту, розробленого для проєкту, проаналізуємо його компоненти та обґрунтуємо вибір структури відповіді LLM.

Оригінальний промпт написаний англійською мовою і його переклад українською наведено у додатку А.

6.1 Аналіз компонентів промпту

Розглянемо основні компоненти промпту та їх функціональне призначення:

6.1.1 Встановлення ролі та контексту

Промпт починається з чіткого визначення ролі: "Ви експерт з візуалізації даних". Це направляє мовну модель на виконання конкретної задачі, пов'язаної з аналізом та візуалізацією даних. Такий підхід допомагає моделі "вжитися" в роль аналітика та надавати більш релевантні відповіді в контексті аналізу даних.

6.1.2 Надання вхідних даних

У промпті переданий набір даних у форматі JSON або CSV, попередньо перетворений у структурований вигляд. Передбачена також обробка випадку, коли дані скорочені через обмеження розміру, з відповідним повідомленням: "Примітка: Дані було скорочено через обмеження розміру. Аналіз базується на підмножині даних."

Такий підхід має кілька переваг:

– забезпечує моделі доступ до реальних даних для аналізу;
– підтримує різні формати (JSON, CSV) через уніфікований інтерфейс;

– вирішує проблему з обмеженням контекстного вікна моделі через механізм скорочення даних.

6.1.3 Чіткі інструкції

Промпт містить п'ять конкретних інструкцій: проаналізувати дані для виявлення шаблонів, тенденцій та інсайтів; створити комплексну інформаційну панель візуалізації з кількома компонентами; повернути масив JSON-конфігурацій віджетів; забезпечити функцію трансформації даних для кожного віджета; використовувати певну мову (українську або англійську). Ці інструкції поетапно спрямовують роботу моделі, починаючи від аналізу даних і закінчуючи формуванням структурованої відповіді.

6.1.4 Детальний опис шаблонів віджетів

Значну частину промпту займає опис п'яти типів віджетів: діаграми (chart), таблиці (table), інсайтів (insights), метрик (metrics) і зведення (summary). Для кожного типу надається детальний шаблон JSON із описом усіх можливих полів та їх призначення. Такий підхід забезпечує однозначне розуміння моделлю очікуваного формату відповіді, демонструє приклади заповнення полів і пояснює їхнє призначення через коментарі. Особливо важливим є поле `dataTransform`, яке містить функцію JavaScript для трансформації вхідних даних. Ця функція виконується на стороні клієнта й забезпечує гнучку обробку даних для кожного віджета.

6.1.5 Важливі рекомендації

Промпт містить шість важливих рекомендацій щодо формування відповіді: надання дійсного тексту функції JavaScript, правильне форматування даних, забезпечення відповідності структури даних типу діаграми, надання змістовних інсайтів, повернення лише валідного масиву JSON-конфігурацій і включення функції dataTransform для кожного віджета. Ці рекомендації допомагають уникнути типових помилок і забезпечують високу якість результатів.

6.1.6 Чіткий формат відповіді

Промпт завершується прикладом очікуваного формату відповіді у вигляді масиву JSON-об'єктів. Це забезпечує однозначне розуміння моделлю структури очікуваної відповіді.

6.2 Оптимізація взаємодії з LLM

Оптимізація взаємодії з LLM здійснюється через декілька ключових підходів: оцінку кількості токенів для визначення розміру вхідних даних та їх скорочення у випадку перевищення встановленого ліміту, забезпечення багатомовної підтримки за допомогою динамічних мовних інструкцій, а також можливість використання різних моделей (GPT-4o, Claude 3 Opus, Gemini 1.5 Pro) для досягнення максимальної ефективності, резервування у випадку недоступності окремих моделей та адаптації до специфічних задач.

6.3 Функціональні можливості промпту

Розроблений промпт дозволяє LLM виконувати комплексний аналіз даних, включаючи виявлення шаблонів, генерацію змістовних інсайтів, виділення ключових метрик та формування узагальнених описів даних. Він підтримує різні типи візуалізацій, такі як стовпчасті, лінійні, кругові, площинні, точкові діаграми, а також створення таблиць з довільними колонками. Крім того, промпт оснащений програмним компонентом для фільтрації, агрегації, сортування, обмеження наборів даних та перетворення їх формату, що значно розширює можливості аналізу.

6.4 Висновки та рекомендації

Система промпту та структурована відповідь LLM демонструють ефективне використання великих мовних моделей для аналізу та візуалізації даних. Основними перевагами є гнучкість обробки різних форматів даних, чітка структура вихідної інформації, багатофункціональність аналітичних можливостей, а також підтримка декількох мов і моделей. Надійність забезпечується строгим

типізуванням та валідацією відповідей, а функції трансформації даних розширюють спектр застосування системи. Подальші вдосконалення можуть включати розширення можливостей візуалізацій, покращення механізмів оцінки та скорочення даних, а також розробку спеціалізованих промптів для конкретних типів аналізу.

ВИСНОВКИ

У результаті виконання даної науково-дослідної роботи було проведено комплексне дослідження, спрямоване на вивчення можливостей використання великих мовних моделей (LLM) для автоматизації процесів візуалізації та аналізу інформації.

В рамках дослідження було здійснено ретельний аналіз предметної галузі та існуючих рішень. Детально вивчено сучасні інструменти для візуалізації та аналізу даних, включаючи Datadog, Grafana, Google Data Studio, Power BI та Tableau. Виявлено їхні основні переваги та недоліки, а також ключові тенденції розвитку галузі, серед яких автоматизація аналізу, інтеграція з великими мовними моделями, підвищення інтерактивності візуалізацій та зростаюча потреба в кастомізації. Цей аналіз підтвердив актуальність обраної теми дослідження та дозволив сформулювати чіткі вимоги до розроблюваної системи.

Теоретичні основи використання великих мовних моделей було систематизовано через комплексний огляд літературних джерел. Дослідження охопило принципи роботи провідних LLM, зокрема Gemini, GPT та Llama, методи візуалізації даних засобами JavaScript бібліотек D3.js та Recharts, архітектурні підходи систем. Особливу увагу приділено методам зберігання, обробки й аналізу великих даних з використанням LLM. Теоретично обґрунтовано підходи до аналізу як структурованих даних у форматах CSV та JSON, так і текстових даних. Досліджено роль промпт-інженерії в оптимізації взаємодії з моделями та проаналізовано ефективність таких моделей як GPT-4o, Claude 3 та Gemini Pro.

На основі проведеного теоретичного дослідження розроблено концептуальну архітектуру та програмний прототип системи. Створена архітектура веб-додатку для аналізу даних базується на модульній структурі та використовує сучасні технології Next.js, AI SDK для уніфікованої взаємодії з OpenAI GPT-4o, Anthropic Claude 3 Opus та Google Gemini 1.5 Pro, а також Zod, Recharts та PostgreSQL. Розроблений прототип успішно реалізує функціональність завантаження даних у форматах CSV та JSON, автоматичну генерацію та інтерактивне відображення різноманітних візуалізацій включаючи графіки та

таблиці, створення текстових інсайтів, а також динамічну трансформацію даних безпосередньо на клієнті.

Експериментальне дослідження ефективності великих мовних моделей проводилось на різних наборах даних з метою тестування їхньої здатності автоматизувати генерацію візуалізацій та аналітичних інсайтів. Результати експериментів продемонстрували, що Claude 3 Opus виявляє високу якість у генерації змістовних текстових інсайтів та виборі найбільш доречних типів візуалізацій для конкретних даних. GPT-4o вирізняється особливою точністю в математичних обчисленнях та генерації коректних JavaScript-функцій для трансформації даних. Gemini 1.5 Pro показує значні переваги у швидкості обробки великих наборів даних. Важливим висновком стало те, що якість та корисність аналізу суттєво залежать від якості складених промптів та структури вихідних даних.

В ході дослідження було обґрунтовано ефективні методи взаємодії з великими мовними моделями для аналізу даних. Запропоновано комплексні підходи до автоматичного аналізу, що включають генерацію SQL-запитів для структурованих даних, виявлення прихованих закономірностей, класифікацію інформації, а також створення відповідних візуалізацій і змістовних текстових інсайтів. Розроблено деталізований промпт, який ефективно керує великими мовними моделями у процесі аналізу та забезпечує формування структурованої та корисної відповіді.

Практична значущість дослідження підтверджена створеним прототипом, який демонструє реальну реалізацію запропонованих підходів і підтверджує основну гіпотезу про ефективність використання великих мовних моделей для спрощення та автоматизації процесів аналізу даних. Особливу цінність система представляє для користувачів без глибоких технічних навичок, надаючи їм потужні інструменти для роботи з даними.

Результати проведеного дослідження можуть слугувати надійною основою для подальшої розробки більш складних інтелектуальних систем аналізу даних. Серед перспективних напрямків подальшого розвитку варто виділити дослідження

складніших методів машинного навчання для аналізу залежностей у даних з глибокою інтеграцією великих мовних моделей, розробку ефективних механізмів обробки потокових даних в режимі реального часу, розширення спектру підтримуваних типів візуалізацій та можливостей їх гнучкої кастомізації. Також важливими напрямками є дослідження методів масштабування системи та оптимізації алгоритмів для роботи з великими обсягами даних, а також розробка спеціалізованих промптів для конкретних типів аналізу та різних галузей застосування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Afanasieva, Iryna, Nataliia Golian, Vira Golian, Artem Khovrat, and Kostiantyn Onyshchenko. "Application of Neural Networks to Identify of Fake News." In Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2023), Volume II: Computational Linguistics Workshop, Kharkiv, Ukraine, April 20-21, 2023. CEUR Workshop Proceedings, Vol. 3396, 346–58. Aachen: CEUR-WS.org, 2023. http://ceur-ws.org/Vol-3390/paper28.pdf.

2. Amatriain, Xavier. "Prompt Design and Engineering: Introduction and Advanced Methods." arXiv preprint arXiv:2401.14423 (2024). https://arxiv.org/abs/2401.14423.

3. Anderson, Eric, Jonathan Fritz, Austin Lee, Bohou Li, Mark Lindblad, Henry Lindeman, Alex Meyer, Parth Parmar, Tanvi Ranade, Mehul A. Shah, Benjamin Sowell, Dan Tecuci, Vinayak Thapliyal, and Matt Welsh. "The Design of an LLM-powered Unstructured Analytics System." In Proceedings of the Conference on Innovative Data Systems Research (CIDR) 2025, January 2025. arXiv:2409.00847. https://arxiv.org/abs/2409.00847.

4. Anthony, Ben, Yijia Jin, and Steffen Gehrmann. "Structured Prompting for LLMs on Tabular Data." arXiv preprint arXiv:2402.01537 (2024). https://arxiv.org/abs/2402.01537.

5. Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer. "D3: Data-Driven Documents." IEEE Transactions on Visualization and Computer Graphics 17, no. 12 (December 2011): 2301–9. https://doi.org/10.1109/TVCG.2011.185.

6. Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. "Language Models are Few-Shot Learners." In

Advances in Neural Information Processing Systems 33 (NeurIPS 2020), edited by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, 1877–1901. Curran Associates, Inc., 2020. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

7. Budnyk, M. O. "2025 M PI IPZ-23-3 Budnyk M.O. Diploma Project: Source Code and Materials." GitHub repository. Accessed October 4, 2024. https://github.com/max-nure/diploma/tree/main/2025_%D0%9C_%D0%9F%D0%86_%D0%86%D0%9F%D0%97-23-3_%D0%91%D1%83%D0%B4%D0%BD%D0%B8%D0%BA_%D0%9C_%D0%9E.

8. Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters." Communications of the ACM 51, no. 1 (January 2008): 107–13. https://doi.org/10.1145/1327452.1327492.

9. Dhillon, Sarinder Kaur, Mogana Darshini Ganggayah, Siamala Sinnadurai, Pietro Lio, and Nur Aishah Taib. "Theory and Practice of Integrating Machine Learning and Conventional Statistics in Medical Data Analysis." Diagnostics 12, no. 10 (2022): 2526. https://doi.org/10.3390/diagnostics12102526.

10. Gemini Team, Google. "Gemini: A Family of Highly Capable Multimodal Models." Google, December 2023. https://storage.googleapis.com/deepmind-media/gemini/gemini_1_report.pdf.

11. James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An Introduction to Statistical Learning: With Applications in R. New York: Springer, 2013.

12. Kang, Ji-Won, and Sun-Yong Choi. "Comparative Investigation of GPT and FinBERT's Sentiment Analysis Performance in News Across Different Sectors." Electronics 14, no. 6 (2025): 1090. https://doi.org/10.3390/electronics14061090.

13. Kevian, Danny, Usama Syed, Xiang Guo, et al. "Capabilities of Large Language Models in Control Engineering: A Benchmark Study." arXiv preprint arXiv:2404.03647 (2024). https://arxiv.org/abs/2404.03647.

14. Konstantinou, Charalampos, and Yuze Wang. "Statistical and Machine

Learning Analysis for the Application of Microbially Induced Carbonate Precipitation as a Physical Barrier to Control Seawater Intrusion." Journal of Contaminant Hydrology 263 (2024): 104337. https://doi.org/10.1016/j.jconhyd.2024.104337.

15. Nazarenko, Dmytro, Iryna Afanasieva, Nataliia Golian, and Vira Golian. "Investigation of the Deep Learning Approaches to Classify Emotions in Texts." In Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021), Volume I: Main Conference, Lviv, Ukraine, April 22-23, 2021. CEUR Workshop Proceedings, Vol. 2870, 206–24. Aachen: CEUR-WS.org, 2021. http://ceur-ws.org/Vol-2870/paper15.pdf.

16. OpenAI. "GPT-4 Technical Report." arXiv preprint arXiv:2303.08774 (2023). https://arxiv.org/abs/2303.08774.

17. Sohail, Shahab Saquib, Dັญญา Østvik Madsen, Yahya Himeur, and Mohiuddin Ashraf. "Using ChatGPT to Navigate Ambivalent and Contradictory Research Findings on Artificial Intelligence." Frontiers in Artificial Intelligence 6 (2023): 1195797. https://doi.org/10.3389/frai.2023.1195797.

18. Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. "Llama: Open and Efficient Foundation Language Models." arXiv preprint arXiv:2302.13971 (2023). https://arxiv.org/abs/2302.13971.

19. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need." In Advances in Neural Information Processing Systems 30 (NIPS 2017), edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 5998–6008. Red Hook, NY: Curran Associates, Inc., 2017. https://papers.nips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

20. Vázquez, Pere-Pau. "Are LLMs Ready for Visualization?" In 2024 IEEE 17th Pacific Visualization Symposium (PacificVis) - Workshops, 343–52. IEEE, 2024. https://doi.org/10.1109/PacificVis60374.2024.00049.

21. Whitaker, Miles. "Technical Principles of Large Language Models: From

Transformer Architectures to Future Challenges." Journal of Computer Science and Software Applications 5, no. 4 (2025).

22. Ye, Yifan, Haotian Li, Chris Bryan, Fan Du, Yun Wang, and Huamin Qu. "Generative AI for Visualization: State of the Art and Future Directions." Visual Informatics 8, no. 1 (March 2024): 1–20. https://doi.org/10.1016/j.visinf.2023.12.004.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

1. Afanasieva, Iryna, Nataliia Golian, Vira Golian, Artem Khovrat, and Kostiantyn Onyshchenko. "Application of Neural Networks to Identify of Fake News." In Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2023), Volume II: Computational Linguistics Workshop, Kharkiv, Ukraine, April 20-21, 2023. CEUR Workshop Proceedings, Vol. 3396, 346–58. Aachen: CEUR-WS.org, 2023. http://ceur-ws.org/Vol-3390/paper28.pdf.

15. Nazarenko, Dmytro, Iryna Afanasieva, Nataliia Golian, and Vira Golian. "Investigation of the Deep Learning Approaches to Classify Emotions in Texts." In Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021), Volume I: Main Conference, Lviv, Ukraine, April 22-23, 2021. CEUR Workshop Proceedings, Vol. 2870, 206–24. Aachen: CEUR-WS.org, 2021. http://ceur-ws.org/Vol-2870/paper15.pdf.