

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій і робототехніки

(повне найменування інституту, назва факультету (відділення))

Кафедра автоматики, електроніки та телекомунікацій

(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до магістерської роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: «Система оцінки якості роботи підприємства з використанням
елементів штучного інтелекту»

Виконав: студент 6 курсу, групи 601-ТТ

спеціальності 172 «Електронні

комунікації та радіотехніка»

шифр і назва напрямку підготовки, спеціальності

Говтяниця А.О.

(прізвище та ініціали)

Керівник Боряк Б.Р.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Полтава – 2025 рік

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
Інститут Навчально-науковий інститут інформаційних технологій та робототехніки
Кафедра Автоматики, електроніки та телекомунікацій
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 172 «Електронні комунікації та радіотехніка»

ЗАТВЕРДЖУЮ

**Завідувач кафедри автоматки,
електроніки та телекомунікацій**

_____ О.В. Шефер

“ _____ ” вересня 2024 р.

ЗАВДАННЯ НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Говтяниці Андрію Олексійовичу

1. Тема проекту (роботи) **«Система оцінки якості роботи підприємства з використанням елементів штучного інтелекту»**
керівник проекту (роботи) Боряк Богдан Радиславович, к.т.н.
затверджена наказом вищого навчального закладу від “09” серпня 2024 року № 818-ф,а
2. Строк подання студентом проекту (роботи) 20.12.2024.
3. Вихідні дані до проекту (роботи) Технічна документація мови програмування Python.
Опис робочих процесів комунікаційного відділу по роботі з клієнтами підприємств сфери обслуговування. Забезпечити автоматизоване оцінювання якості роботи працівників комунікаційного відділу по роботі з клієнтами за допомогою сучасних комп’ютерних технологій та елементів штучного інтелекту.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз застосування мовних моделей штучного інтелекту у сферах людської діяльності.
Імплементация елементів штучного інтелекту для автоматизації робочих процесів підприємств.
Аналіз робочих процесів комунікаційного відділу по роботі з клієнтами підприємств сфери обслуговування.
Вибір комп’ютерних технології для реалізації застосунку для автоматизації якості оцінювання роботи працівників комунікаційного відділу по роботі з клієнтами.
Особливості архітектури серверного застосунку.
Реалізація серверного застосунку.
Дослідження і оцінювання критеріїв якості роботи серверного застосунку.
Особливості розгортання серверного застосунку.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових плакатів):
 - 1) Мета та задачі дослідження
 - 2) Аналіз застосування мовних моделей у сферах людської діяльності

- 3) Порівняльна характеристика швидкодії роботи мовних моделей штучного інтелекту
- 4) Аналітичний огляд комп'ютерних технологій
- 5) Архітектура серверного застосунку
- 6) Результат роботи застосунку
- 7) Дослідження і оцінювання критеріїв якості роботи серверного застосунку
- 8) Особливості CI/CD серверного застосунку
- 9) Висновки

7. Дата видачі завдання 02.09.2024 р.

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів магістерської роботи	Термін виконання етапів роботи			Примітка (плакати)
1	Аналіз застосування мовних моделей штучного інтелекту у сферах людської діяльності.	07.10.24		10%	Пл. 1
2	Імплементация елементів штучного інтелекту для автоматизації робочих процесів підприємств.	16.10.24	I	25%	Пл. 2
3	Аналіз робочих процесів комунікаційного відділу по роботі з клієнтами підприємств сфери обслуговування.	05.11.24		40%	Пл. 3,4
4	Вибір комп'ютерних технологій для реалізації застосунку для автоматизації якості оцінювання роботи працівників комунікаційного відділу по роботі з клієнтами.	12.11.24	II	60%	Пл. 5
5	Особливості архітектури серверного застосунку. Реалізація серверного застосунку.	19.11.24		70%	Пл. 6, 7
6	Дослідження і оцінювання критеріїв якості роботи серверного застосунку.	26.11.24		80%	Пл. 8
7	Особливості розгортання серверного застосунку.	11.12.24		90%	Пл. 9
8	Оформлення магістерської роботи	19.12.24	III	100%	

Магістрант _____ Говтяниця А.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Боряк Б.Р.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

кваліфікаційної роботи

Система оцінки якості роботи підприємства з використанням елементів штучного інтелекту

Структура роботи. Робота складається зі вступу, трьох розділів, висновків та списку використаних джерел; містить 66 сторінку основного тексту.

Ключові слова: штучний інтелект, мовні моделі, контакт-центр, аналіз діалогів, якість обслуговування, автоматизація, NLP, серверний застосунок.

У магістерській роботі проведено аналіз застосування мовних моделей штучного інтелекту в різних сферах діяльності та досліджено можливості їх використання для автоматизації процесів підприємства. Визначено основні підходи до впровадження ШІ в роботу комунікаційних відділів закладів сфери обслуговування, зокрема для оцінювання якості взаємодії з клієнтами.

Розроблено автоматизовану систему аналізу діалогів контакт-центру, що включає обробку аудіозаписів, їх транскрипцію та оцінювання за визначеними критеріями (ввічливість, ефективність, відповідність стандартам). Запропоновано архітектуру серверного застосунку, структуру бази даних та інтеграцію з API штучного інтелекту для обробки текстових даних.

Мета дослідження – розробка автоматизованої системи оцінки якості роботи підприємства на основі аналізу комунікацій із клієнтами з використанням методів штучного інтелекту.

Об’єкт дослідження – процес оцінювання якості роботи підприємства на основі взаємодії з клієнтами.

Предмет дослідження – методи та засоби використання штучного інтелекту для аналізу діалогів і формування рекомендацій щодо покращення якості обслуговування.

ABSTRACT

master's thesis

System for evaluating the quality of enterprise performance using artificial intelligence elements

Structure of the work. The thesis consists of an introduction, three chapters, conclusions and references; it contains 66 pages of the main text.

Keywords: artificial intelligence, language models, contact center, dialogue analysis, service quality, automation, NLP, server application.

In the master's thesis, an analysis of the application of artificial intelligence language models in various fields of activity was carried out, and the possibilities of their use for automation of enterprise processes were investigated. The main approaches to the implementation of AI in the operation of communication departments of service enterprises, particularly for evaluating the quality of customer interactions, were identified.

An automated system for analyzing contact center dialogues was developed, which includes processing of audio recordings, their transcription, and evaluation based on defined criteria such as politeness, efficiency, and compliance with service standards. The architecture of a server-side application, a database structure, and integration with artificial intelligence APIs for processing textual data were proposed.

The purpose of the research is to develop an automated system for evaluating the quality of enterprise performance based on the analysis of customer communications using artificial intelligence methods.

The object of research is the process of evaluating the quality of enterprise performance based on customer interaction.

The subject of research is the methods and tools of applying artificial intelligence for dialogue analysis and generation of recommendations to improve service quality.

ЗМІСТ

Вступ.....	5
Розділ 1. Аналіз застосування мовних моделей штучного інтелекту у сферах людської діяльності.....	7
1.1. Огляд використання мовних моделей ШІ у сучасних сферах діяльності	10
1.2. Порівняльна характеристика засобів штучного інтелекту	14
1.2.1. Аналіз та порівняння моделей штучного інтелекту від OpenAI та Google.....	16
1.2.2. Порівняльна характеристика швидкодії роботи мовних моделей штучного інтелекту.....	20
1.3. Імплементація елементів штучного інтелекту для автоматизації робочих процесів підприємств	23
Розділ 2. Методи впровадження елементів штучного інтелекту в робочі процеси підприємств.....	27
2.1. Аналіз робочих процесів комунікаційного відділу по роботі з клієнтами підприємств сфери обслуговування	28
2.2. Вибір комп'ютерних технологій для реалізації застосунку для автоматизації якості оцінювання роботи працівників комунікаційного відділу по роботі з клієнтами	33
Розділ 3. Особливості архітектури серверного застосунку	37
3.1. Опис реалізації серверного застосунку.....	45
3.2. Дослідження і оцінювання критеріїв якості роботи серверного застосунку	50
3.3. Особливості розгортання серверного застосунку.....	55
Висновки	63
Література	65

Вступ

Актуальність теми. На сьогодні автоматизація та оптимізація робочих процесів стали ключовими факторами успіху в умовах високої конкуренції, а здатність адаптувати новітні технології до реальних бізнес-задач дозволяє компаніям залишатися ефективними та конкурентоспроможними.

Розвиток штучного інтелекту, зокрема його мовних моделей, відкриває нові горизонти для аналізу та оцінювання якості взаємодії з клієнтами. У контексті закладів громадського харчування, які характеризуються високим рівнем комунікації з клієнтами, така система є не лише інструментом автоматизації, але й способом покращення обслуговування та підвищення задоволеності споживачів. Інтеграція ШІ у роботу контакт-центрів дозволяє автоматизувати аналіз діалогів між операторами та клієнтами, виявляючи ключові моменти, що впливають на якість послуг, такі як швидкість реакції, ввічливість та відповідність стандартам обслуговування.

Об'єкт дослідження. Процес оцінювання якості роботи підприємства на основі взаємодії з клієнтами.

Предмет дослідження. Використання штучного інтелекту для аналізу діалогів та формування рекомендацій щодо підвищення якості обслуговування.

Мета дослідження. Розробка автоматизованої системи оцінки якості роботи підприємства, що інтегрує алгоритми штучного інтелекту для аналізу комунікацій з клієнтами.

Задачі дослідження:

1. Дослідити процес оцінювання якості роботи підприємства на основі взаємодії з клієнтами.
2. Проаналізувати сучасні системи автоматизованої оцінки якості роботи контакт-центрів.
3. Вивчити можливості інтеграції алгоритмів штучного інтелекту для аналізу діалогів між клієнтами та операторами.

4. Розробити структуру бази даних для зберігання даних розмов та результатів оцінювання.
5. Створити проєкт, що інтегрує API штучного інтелекту для обробки текстових даних із транскрипцій.
6. Налаштувати процес автоматизації завантаження аудіозаписів, їхньої транскрипції та отримання оцінок за допомогою OpenAI API.
7. Розробити алгоритм обробки текстових даних для оцінювання якості розмов за ключовими параметрами, такими як ввічливість, ефективність та відповідність стандартам.
8. Провести тестування роботи автоматизованої системи на реальних даних, отриманих із записів діалогів у контакт-центрі.
9. Виконати аналіз результатів роботи системи, включаючи порівняння оцінок ШІ із оцінками, наданими експертами.
10. Запропонувати рекомендації щодо покращення роботи системи та адаптації її під специфіку підприємств громадського харчування.

Розділ 1. Аналіз застосування мовних моделей штучного інтелекту у сферах людської діяльності

Штучний інтелект (ШІ) пройшов довгий шлях свого розвитку від абстрактної концепції до прикладної науки. Спочатку ідеї про створення розумних машин розвивалися в філософських працях ще в давнину, однак практична реалізація стала можливою лише в середині ХХ століття, коли було винайдено електронно-обчислювальні машини. Перші комп'ютери були спроможні виконувати математичні розрахунки, що заклало фундамент для обчислювальних наук і розвитку математичної логіки. Прорив у формуванні основ ШІ відбувся завдяки працям Алана Тьюринга, британського математика і логіка, який у 1950 році запропонував так званий «Тест Тьюринга». Цей тест був задуманий для перевірки здатності машини демонструвати поведінку, яка нагадує людське мислення. Вважається, що машина проходить тест, якщо людина, спілкуючись з нею через текстовий інтерфейс, не може визначити, чи спілкується вона з людиною чи з машиною.

Протягом 1950-х і 1960-х років інтерес до ШІ зростав, і дослідники в різних університетах почали розробляти програми для імітації різних аспектів інтелектуальної поведінки, таких як розв'язання задач, маніпуляції мовою та розпізнавання образів. Одним із перших і відомих досягнень у цій галузі стала програма «Логік Теоріст» (Logic Theorist), розроблена в 1955 році Аланом Ньюеллом і Гербертом Саймоном, яка була здатна автоматично доводити математичні теореми. Це стало важливим кроком, оскільки показало, що машини можуть розв'язувати задачі, які раніше вважалися виключно людською прерогативою.

У 1960-х роках фокус досліджень змістився до символічного підходу, який передбачав використання символів і маніпулювання ними для моделювання людських розумових процесів. Однак у 1970-х роках через обмеженість обчислювальних потужностей і низьку ефективність ШІ у вирішенні реальних проблем інтерес до цієї галузі дещо згас. Лише з 1980-х років, із появою нових алгоритмів і значним зростанням обчислювальних

потужностей, відбулося відродження ШІ. Значну роль у цьому процесі відіграли експертні системи — комп'ютерні програми, які накопичували знання в певних сферах і могли давати поради або вирішувати задачі, покладаючись на базу правил і знань.

Паралельно з розвитком комп'ютерної техніки зростало значення теоретичних основ ШІ, зокрема теорії графів, теорії ймовірностей та статистичних методів. Теорія графів стала основою для багатьох застосувань ШІ, дозволяючи моделювати складні структури і зв'язки між даними. Графи представляють об'єкти у вигляді вузлів, а зв'язки між ними у вигляді ребер. Це стало основою для таких напрямків, як соціальні мережі, системи рекомендацій, маршрутизація та аналіз мереж. Зокрема, алгоритми на графах, такі як алгоритм Дейкстри для пошуку найкоротшого шляху, знайшли широке застосування у навігаційних і логістичних системах.

Теорія ймовірностей стала фундаментом для розвитку статистичних методів і методів машинного навчання. Вона дозволяє системам ШІ приймати рішення в умовах невизначеності, використовуючи ймовірнісні моделі для прогнозування результатів. Наприклад, метод Байєсівської класифікації, заснований на теоремі Байєса, застосовується для обробки даних, де є неоднозначність. Іншим важливим інструментом у статистиці є методи регресії, які дозволяють прогнозувати значення на основі наявних даних. Згодом ці інструменти стали основою для створення нейронних мереж та інших адаптивних систем.

У 1980-х роках з'явилися перші багатошарові нейронні мережі, що стало важливим кроком у розвитку технології глибокого навчання. На відміну від попередніх моделей, багатошарові мережі можуть навчатися через корекцію своїх внутрішніх параметрів, зменшуючи похибки шляхом обчислення зворотного поширення. Ця методика, відома як алгоритм зворотного поширення помилок (backpropagation), дозволила значно підвищити точність моделей, особливо при аналізі великих і складних наборів даних.

Протягом 2000-х і 2010-х років розвиток ШІ активно продовжувався завдяки появі нових підходів, таких як глибоке навчання та навчання з підкріпленням. Глибоке навчання, яке базується на багатошарових нейронних мережах, дозволяє обробляти великі масиви різноманітних даних, зокрема зображення, текст, аудіо. Системи на основі глибокого навчання широко застосовуються у таких сферах, як розпізнавання образів, обробка природної мови та медична діагностика. Навчання з підкріпленням, де модель навчається через отримання зворотного зв'язку від навколишнього середовища, стало основою для створення систем, здатних самостійно навчатися і вдосконалюватися в умовах невизначеності.

Загалом, сучасний ШІ поєднує в собі методи, які дозволяють автоматизованим системам аналізувати дані, робити прогнози та приймати рішення. Ці системи можуть бути адаптовані під конкретні задачі, такі як аналіз діалогів, як у випадку оцінки якості роботи співробітників контактних центрів, що безпосередньо відповідає темі магістерської роботи. Завдяки постійному вдосконаленню алгоритмів і зростанню обчислювальних потужностей ШІ стає невід'ємною частиною багатьох галузей, від бізнесу і медицини до освіти та соціальних сервісів.

Штучний інтелект (ШІ) нині стрімко розвивається, і однією з його найбільш перспективних галузей є мовні моделі. Вони дозволяють розуміти, генерувати та аналізувати природну мову, що має велике значення для автоматизації багатьох процесів у різних сферах людської діяльності. Мовні моделі, як частина штучного інтелекту, застосовуються у різноманітних контекстах, таких як бізнес, освіта, медицина, юридична сфера, фінанси та інші. У цьому розділі розглянемо основні напрями використання мовних моделей у цих сферах, їхні можливості, а також виклики, що постають перед їхнім впровадженням.

Згідно з [1] Мовна модель – це тип штучного інтелекту, розроблений для обробки та генерування природної мови. Основна мета мовних моделей полягає в тому, щоб навчитися розуміти та продукувати текст, який виглядає і

звучить так, ніби його написала людина. Визначення мовної моделі можна окреслити як статистичну або машинну модель, що має здатність передбачати ймовірність наступних слів у послідовності, виходячи з попереднього контексту.

Однією з найбільш значущих сфер застосування мовних моделей є бізнес-аналітика. У сучасних умовах обсяги даних, які генерують підприємства, зростають експоненціально, і необхідність їхньої ефективної обробки стає все більш актуальною. Мовні моделі ШІ забезпечують автоматизацію рутинних процесів, таких як аналіз звітів, генерація бізнес-прогнозів, опрацювання клієнтських запитів та навіть аналіз споживчих настроїв на основі текстових даних з соціальних мереж. Завдяки здатності ШІ до машинного навчання, такі моделі можуть адаптуватися до нових умов і даних, що робить їх універсальним інструментом для сучасного бізнесу. Важливо зазначити, що використання мовних моделей не лише підвищує продуктивність підприємства, але й допомагає знизити витрати на людські ресурси, особливо в контексті аналізу великих масивів інформації.

1.1. Огляд використання мовних моделей ШІ у сучасних сферах діяльності

У сфері освіти мовні моделі штучного інтелекту знаходять застосування для створення адаптивних систем навчання, які підлаштовуються під індивідуальні потреби учнів. Можливості таких систем включають автоматичну генерацію навчальних матеріалів, проведення тестування знань, а також надання індивідуальних рекомендацій для поліпшення освітнього процесу. Штучний інтелект може допомагати в автоматизації адміністративних завдань, таких як перевірка письмових робіт, забезпечуючи точність та швидкість оцінювання. Крім того, завдяки розвитку мовних технологій у сфері перекладу, зростає доступність навчальних матеріалів різними мовами, що сприяє глобалізації освітніх процесів.

У медицині мовні моделі ШІ відіграють важливу роль у підвищенні точності діагностики та покращенні комунікації між пацієнтами і медичними працівниками. Згідно з [2] більшість застосунків штучного інтелекту в медицині зчитують дані певного типу, чи то числові, наприклад, частота серцевих скорочень або кров'яний тиск, так і на основі зображень, наприклад, МРТ або зображення зразків біопсійної тканини, як вхідні дані. Потім алгоритми вивчаються на даних і виробляють або ймовірність, або класифікацію. Вони дозволяють здійснювати обробку великої кількості медичних даних, включаючи електронні медичні картки та наукові публікації, для покращення діагностичних рішень та розробки персоналізованих планів лікування. Згідно з [3] стало можливим результатом може бути ймовірність утворення артеріального згустку з урахуванням даних про частоту серцевих скорочень і кров'яного тиску. Це відкриває нові можливості для підвищення доступності медичних послуг, особливо у віддалених регіонах.

У юридичній сфері вони здатні автоматизувати процеси аналізу великих масивів правових документів, прискорюючи процес підготовки юридичних висновків і контрактів. Згідно з [4] На штучний інтелект покладають сподівання в тому, щоб допомогти покращити судову практику: виявляти типові правові ситуації, розробляти алгоритми дій (наприклад, визначитися з гарантійним строком або строком давності, встановити вид порушення суб'єктивних прав і свобод, обрати спосіб захисту тощо), зіставляти зі зразком (судовим прецедентом), абстрагуватися від обставин, фактів, документів, речей та інших доказів, які не мають відношення до предмета розгляду, не охоплені предметом спору або не відображають обраний позивачем спосіб захисту, або не передбачені відповідною нормою матеріального права, на котру посилається позивач, виявляти нетипову поведінку суду за звичайних умов, так звані «аномальні» судові рішення, обробляти значний обсяг інформації, готувати проекти судових рішень тощо. За допомогою мовних моделей можна швидко знаходити необхідні правові норми та судові прецеденти, що значно зменшує час, необхідний для правових досліджень.

Також, мовні моделі можуть використовуватися для поліпшення якості юридичних консультацій, оскільки вони дозволяють аналізувати запити клієнтів та надавати відповідні рекомендації на основі аналізу законодавства та судової практики.

Широко розповсюджені мовні моделі ШІ у фінансовій сфері для вирішення різноманітних завдань, пов'язаних із аналізом ринкових тенденцій, прогнозуванням курсу акцій, а також управлінням ризиками. Згідно з [5] Наразі багато клієнтів банків скаржаться на те, що вони не отримують консультації від банків належного рівня. Згідно з дослідженням KPMG, 63% клієнтів відчувають недоліки у рівні консультативної підтримки, а 66% тих, хто отримав консультацію, вважають, що вона була неефективною. Ця проблема турбує не тільки клієнтів, але також становить занепокоєння для 45% керівників банків. Вони вважають, що їхні компанії опинилися нижче середнього рівня щодо безпечного розгортання технологій та розробки рішень для задоволення потреб клієнтів.

Отже, засоби штучного інтелекту можуть суттєво підвищити ефективність консультативної підтримки банківських клієнтів завдяки автоматизації аналізу запитів та наданню персоналізованих рекомендацій. Використання чат-ботів на основі обробки природної мови дозволяє надавати оперативну підтримку в режимі реального часу, зменшуючи навантаження на працівників. Системи штучного інтелекту можуть аналізувати історію взаємодій з клієнтом, прогнозувати його потреби та пропонувати оптимальні фінансові рішення. Окрім цього, впровадження аналітичних платформ, які працюють з великими обсягами даних, сприяє ідентифікації проблемних аспектів обслуговування, що дозволяє банкам удосконалювати свої стратегії та відповідати очікуванням клієнтів.

Також варто відзначити, що системи ШІ можуть бути корисними у визначенні фейкової інформації, адже можуть оперативно перевіряти великі об'єми даних. Згідно з [6] Машинне навчання стає необхідним інструментом у боротьбі з фейковим контентом, оскільки воно дозволяє розробляти моделі

та алгоритми для автоматичного виявлення та фільтрації фейкової інформації на основі аналізу великих обсягів даних та виявлення характерних ознак, що сприяє підвищенню рівня достовірності та надійності інформації, яку споживачі отримують.

У рамках розробки системи оцінки якості роботи підприємства було обрано заклад громадського харчування як основний об'єкт для аналізу та тестування. Це рішення обумовлене тим, що заклади громадського харчування характеризуються високим рівнем взаємодії з клієнтами, великою кількістю транзакцій, а також необхідністю оперативного реагування на замовлення та зворотний зв'язок від споживачів. Таким чином, впровадження елементів штучного інтелекту у цей сектор дозволяє не тільки автоматизувати численні процеси, але й підвищити якість обслуговування та ефективність роботи персоналу.

Одним із ключових елементів системи оцінки якості роботи в закладі громадського харчування є аналіз роботи операторів колл-центру, які відповідають за підтримку клієнтів та оформлення замовлень. Мовні моделі штучного інтелекту можуть бути використані для моніторингу комунікацій між операторами та клієнтами. Це дозволяє автоматично оцінювати рівень обслуговування на основі таких критеріїв, як тон розмови, швидкість реакції на запити, правильність надання інформації та відповідність стандартам обслуговування.

Застосування мовних моделей для аналізу діалогів в колл-центрі дозволяє не тільки виявляти можливі проблеми у комунікації, але й надавати рекомендації для поліпшення взаємодії з клієнтами. Це може бути корисним для підвищення рівня задоволеності клієнтів, а також для більш точного контролю за якістю роботи операторів. Наприклад, система може виявляти випадки, коли оператор не врахував важливу інформацію від клієнта, або якщо комунікація була надто повільною чи неефективною.

Крім того, аналіз колл-центру може допомогти в оптимізації процесів оформлення замовлень. Штучний інтелект може ідентифікувати типові

проблеми, які виникають під час обробки замовлень, такі як помилки в записі даних або затримки у підтвердженні замовлень. Це дозволяє закладу оперативно реагувати на проблеми і вдосконалювати процеси роботи, що в кінцевому результаті сприяє покращенню якості обслуговування клієнтів.

1.2. Порівняльна характеристика засобів штучного інтелекту

Штучний інтелект (ШІ) базується на моделюванні інтелектуальних процесів людини за допомогою алгоритмів, математичних методів та комп'ютерних технологій. Його робота ґрунтується на здатності системи аналізувати дані, виявляти закономірності, приймати рішення та адаптуватися до нових умов на основі отриманих знань. Структура ШІ включає кілька ключових компонентів, таких як алгоритми навчання, обробка даних, моделі представлення знань, а також методи прийняття рішень.

Основою роботи ШІ є математичні обчислення, що дозволяють формалізувати інтелектуальні завдання. Лінійна алгебра відіграє важливу роль у представленні та обробці даних у вигляді векторів і матриць. Наприклад, обчислення ваг у нейронних мережах базується на операціях множення матриць та використанні сигмоїдальних або інших активаційних функцій. Теорія ймовірностей та статистика використовуються для побудови моделей машинного навчання, особливо у задачах класифікації, регресії та оцінювання ризиків. Метод градієнтного спуску, широко застосовуваний у навчанні нейронних мереж, дозволяє оптимізувати функцію втрат, поступово коригуючи параметри моделі для мінімізації похибки.

Графи є важливою частиною роботи ШІ, адже вони дозволяють структурувати та аналізувати взаємозв'язки між об'єктами. Графи використовуються у задачах обробки природної мови, рекомендаційних системах та пошукових алгоритмах. Наприклад, у графових нейронних мережах кожен вузол графа може бути представлений як окрема сутність із властивостями, а ребра між ними визначають їхні взаємозв'язки. Такі

структури допомагають моделювати складні системи, як-от соціальні мережі чи транспортні мережі.

ШІ також включає механізми навчання, які поділяються на три основні категорії: навчання з учителем, без учителя та підкріплювальне навчання. У навчанні з учителем моделі оптимізуються на основі наданих міток у даних, використовуючи такі алгоритми, як підтримувальні векторні машини або дерева рішень. Навчання без учителя спрямоване на виявлення прихованих структур у даних без попередньої класифікації, наприклад, через методи кластеризації або зменшення розмірності, як-от метод головних компонент. Підкріплювальне навчання моделює процес прийняття рішень, де агент взаємодіє із середовищем, отримуючи винагороду або штрафи за свої дії, що дозволяє оптимізувати стратегію поведінки.

Для представлення знань у системах ШІ використовуються семантичні мережі, онтології та логічні моделі. Логічні підходи, зокрема булева алгебра та предикативна логіка, застосовуються для формалізації правил, які визначають взаємозв'язки між об'єктами. Семантичні мережі дозволяють будувати ієрархічні структури, у яких поняття пов'язані між собою через визначені відношення, що сприяє автоматизації процесів пошуку інформації або прийняття рішень.

Робота систем ШІ також базується на великих обсягах даних, які є основою для тренування моделей. Для забезпечення точності та узагальненості результатів необхідна ретельна попередня обробка даних, включно з очищенням, нормалізацією та виділенням ключових характеристик. Бази даних та хмарні обчислення дозволяють зберігати та обробляти великі масиви інформації, необхідні для роботи ШІ.

Окрім цього, системи ШІ інтегрують методи оптимізації, такі як генетичні алгоритми чи методи рою часток, які допомагають вирішувати складні задачі, що потребують пошуку найкращих рішень у багатовимірних просторах. Інша важлива складова – алгоритми обробки природної мови, які дозволяють системам ШІ розуміти текст і мову людини. Вони використовують

методи марковських процесів, рекурентні нейронні мережі та трансформери, щоб розпізнавати контекст і сенс повідомлень.

У сукупності ці елементи формують складну архітектуру, яка дозволяє ШІ адаптуватися до нових завдань, виявляти приховані закономірності та забезпечувати ефективну взаємодію з користувачем. Завдяки їх поєднанню штучний інтелект стає не лише інструментом аналізу, а й активним учасником прийняття рішень у різних сферах діяльності.

1.2.1. Аналіз та порівняння моделей штучного інтелекту від OpenAI та Google

У сучасному світі технологій мовні моделі є важливим елементом для багатьох сфер діяльності, що вимагають обробки природної мови. Серед найвідоміших і найрозповсюдженіших моделей є GPT-4 та його варіанти від OpenAI. Також важливо зазначити активну конкуренцію з боку інших технологічних гігантів, таких як Google, яка розвиває свою власну мовну модель під назвою Gemini. У цьому підрозділі розглянемо порівняння між різними версіями GPT від OpenAI, а також проведемо порівняння з конкуруючими рішеннями.

OpenAI пропонує кілька версій моделей GPT-4 та GPT-3.5, які відрізняються між собою за продуктивністю, обсягом пам'яті та ефективністю обробки даних. Ці моделі адаптовані для вирішення різних завдань: від генерації тексту високої складності до забезпечення інтерактивних обговорень у реальному часі. Версії GPT-4 зазвичай мають більш розвинені можливості аналізу, що дозволяє їм краще розуміти контекст та створювати більш точні й релевантні відповіді. У свою чергу, моделі GPT-3.5 забезпечують швидкість та оптимізацію обчислювальних ресурсів, що робить їх ефективними для задач, які не потребують глибокої обробки. Користувачі можуть обирати відповідну модель залежно від специфіки своїх потреб, враховуючи баланс між якістю результатів та швидкістю виконання. Приклад графічного інтерфейсу ChatGPT зображено на рисунку 1.

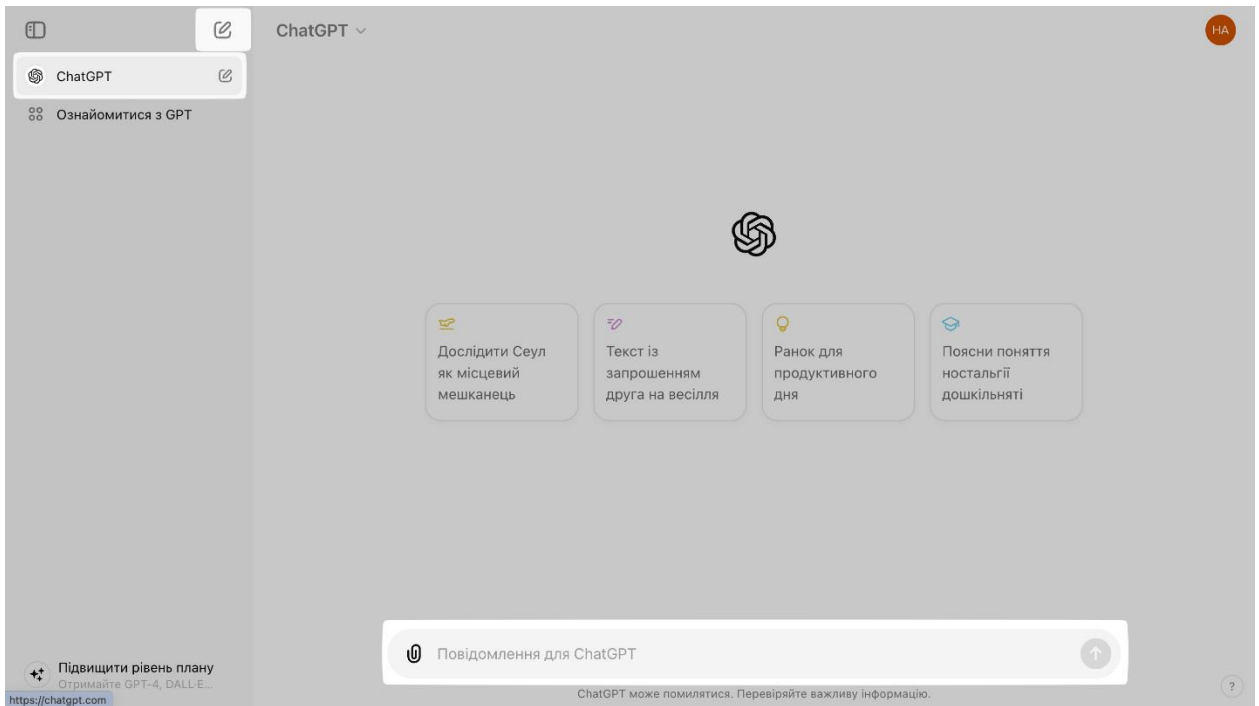


Рисунок 1. Приклад зображення графічного інтерфейсу ChatGPT.

- GPT-4o і GPT-4o mini: Це оптимізовані версії GPT-4, створені для досягнення більшої ефективності за рахунок зниження ресурсів. GPT-4o mini є більш компактною моделлю, що робить її придатною для застосувань, де важлива швидкість обробки та низькі вимоги до апаратних ресурсів. Незважаючи на зменшені розміри, GPT-4o mini забезпечує високу якість розуміння та генерації тексту, що робить її корисною для менш вимогливих завдань, таких як автоматичні відповіді на запити або базовий аналіз тексту.
- o1-preview і o1-mini: Моделі цієї серії належать до нового покоління мовних моделей, яке OpenAI експериментально вводить на ринок. Вони акцентують увагу на збалансованому поєднанні швидкості та точності, надаючи розробникам можливість використовувати моделі з різними рівнями потужності для конкретних завдань. o1-mini, як і GPT-4o mini, пропонує меншу обчислювальну складність, що знижує час обробки запитів і робить її оптимальним вибором для швидких застосувань.
- GPT-4 Turbo і GPT-4: GPT-4 є основною версією моделі від OpenAI, яка призначена для завдань високої складності, що потребують глибокого розуміння тексту та здатності аналізувати великі обсяги даних. Згідно з

[7] Постійні оновлення GPT-4 покращили його здатність розв'язувати складні математичні рівняння та працювати з абстрактним числовим аналізом, що робить цю модель ключовим вибором для галузей, які залежать від точності. Вона підтримує високий рівень контексту, що робить її незамінною для складних діалогових систем або дослідницьких завдань. GPT-4 Turbo, у свою чергу, є вдосконаленою та пришвидшеною версією, що працює з більшими обсягами даних за короткий час, при цьому підтримуючи практично той самий рівень якості обробки тексту. Це робить її ідеальним вибором для задач, де важлива і точність, і швидкість обробки.

- GPT-3.5 Turbo: Хоча ця модель є попередником GPT-4, вона продовжує залишатися популярною завдяки її високій ефективності у багатьох сферах. GPT-3.5 Turbo забезпечує значно менші витрати на обчислення порівняно з GPT-4, що робить її оптимальною для рутинних завдань, таких як автоматичне створення текстів або базовий аналіз даних.

Серед головних конкурентів OpenAI на ринку мовних моделей виділяється Google зі своєю новою мовною моделлю Gemini. Згідно з [8] Gemini є результатом масштабних спільних зусиль команд Google, включаючи наших колег з Google Research. Модель була створена з нуля як мультимодальна, що означає її здатність узагальнювати та безперешкодно розуміти, працювати з різними типами інформації, такими як текст, код, аудіо, зображення та відео, і поєднувати їх між собою. Вона здатна обробляти широкий спектр завдань, від генерації тексту до автоматизації діалогів і аналізу великих обсягів даних. Gemini відрізняється особливою інтеграцією з екосистемою Google, що робить її привабливою для бізнесу, який вже використовує інші сервіси компанії. Крім того, Gemini демонструє високу продуктивність при взаємодії з контекстно насиченими задачами, такими як аналіз голосових запитів та обробка багатомовних текстів, що дає їй перевагу у глобальних ринках. Приклад графічного інтерфейсу Gemini зображено на рисунку 2.

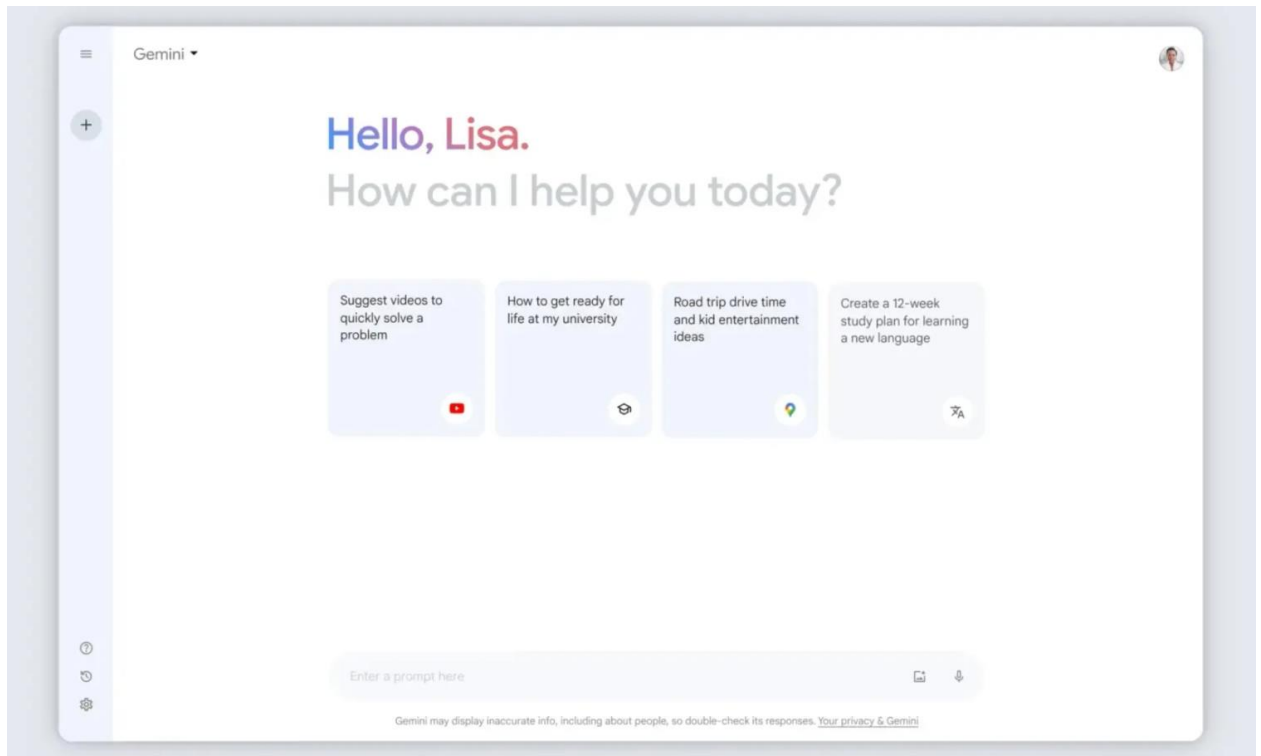


Рисунок 2. Приклад графічного інтерфейсу Gemini.

У порівнянні з моделями OpenAI, Gemini пропонує значну інтеграцію з пошуковими алгоритмами Google, що надає їй певні переваги у завданнях, пов'язаних з доступом до свіжих даних. Водночас моделі GPT-4 Turbo та GPT-4 забезпечують кращу здатність до генерації природного тексту в контексті діалогів та більш глибокого аналізу, що робить їх кращими для таких завдань, як ведення розмов чи написання складних документів.

Обидві компанії, OpenAI і Google, прагнуть оптимізувати свої мовні моделі для різноманітних завдань, забезпечуючи кращу продуктивність та точність. Основними викликами для моделей від OpenAI є забезпечення швидкості та ефективності обробки запитів, оскільки вони покладаються на великі обсяги даних для підвищення точності. Gemini, зі свого боку, може зіткнутися з проблемами інтеграції у сферу генерації високоякісного тексту, хоча вона має перевагу в аналізі пошукових даних та багатомовній підтримці.

Вибір між моделями залежить від специфіки завдання: для інтеграції в системи аналізу роботи колл-центрів, моделі GPT-4 Turbo можуть запропонувати більш гнучкі рішення завдяки своїй здатності обробляти великі

масиви діалогів, тоді як Gemini може бути корисною для пошукових завдань або аналізу тенденцій на основі веб-даних.

1.2.2. Порівняльна характеристика швидкодії роботи мовних моделей штучного інтелекту

Мовні моделі штучного інтелекту, такі як GPT-4 від OpenAI, Gemini від Google та Claude 3 від Anthropic, відображають сучасні досягнення у сфері штучного інтелекту, що дає змогу вирішувати широкий спектр завдань, пов'язаних із аналізом текстових даних, генерацією змістовних відповідей та обробкою мультимодальних інформаційних потоків. Відмінності у їхній продуктивності, підходах до обробки даних та інтерпретації результатів зумовлюють їхню спеціалізацію у певних контекстах застосування. GPT-4 демонструє стабільну продуктивність у більшості текстових завдань завдяки розвиненому механізму розуміння контексту, що забезпечує високу точність та релевантність відповідей. Ця модель ефективно адаптується до різних сценаріїв, таких як автоматизація бізнес-процесів, написання текстів або взаємодія через діалогові системи. Завдяки можливості інтеграції у хмарні сервіси та використанню оптимізованих ресурсів, GPT-4 часто є вибором для завдань, які вимагають високого рівня гнучкості та стабільності.

Gemini від Google, яка вирізняється своєю орієнтацією на мультимодальність, є однією з найбільш перспективних моделей для задач, пов'язаних із обробкою даних різного типу. Ця модель інтегрує текстову інформацію із зображеннями, аудіо та іншими медіа, що дозволяє їй вирішувати комплексні завдання, включаючи аналіз візуальних даних у поєднанні з текстовими описами або побудову систем автоматичного перекладу з урахуванням невербальних компонентів. Gemini також надає значну перевагу у завданнях пошукової оптимізації завдяки тісній інтеграції з пошуковою екосистемою Google, що забезпечує доступ до найактуальніших даних. Однак складність мультимодальної обробки може спричинити

зростання часу виконання окремих запитів, що робить цю модель менш придатною для задач, які потребують негайного результату.

Claude 3 від Anthropic є прикладом моделі, орієнтованої на деталізацію та пояснення. Вона демонструє виняткові результати у задачах, що вимагають не лише розв'язання, але й глибокого обґрунтування відповіді, таких як написання коду, розв'язання складних математичних задач або створення детальних інструкцій. Claude 3 також відома своєю здатністю інтегрувати логічні аргументи у відповідях, що робить її особливо цінною у сфері освіти та технічного консалтингу. Порівняно з GPT-4, ця модель часто надає більше пояснень, що корисно для навчальних завдань, але її продуктивність може бути обмежена у випадках, коли потрібно обробляти значні обсяги даних за короткий проміжок часу.

Вибір між цими моделями залежить від багатьох факторів, включаючи обсяг даних, формат інформації, що обробляється, та необхідний рівень точності. GPT-4 є оптимальним вибором для задач, які вимагають універсальності, швидкості та стабільності. Вона підходить для інтерактивних систем, створення контенту та обробки великих текстових масивів, де важливим є збереження контексту. Gemini, у свою чергу, стає незамінною для мультимодальних задач, де поєднуються текст, візуальна інформація та інші види даних. Завдяки своїй гнучкості та мультимодальній підтримці вона ідеально підходить для аналізу даних у динамічних середовищах. Claude 3 вирізняється своєю здатністю до аналітичного мислення та деталізації, що робить її особливо ефективною у спеціалізованих сферах, таких як навчання, технічний аналіз та написання складних інструкцій.

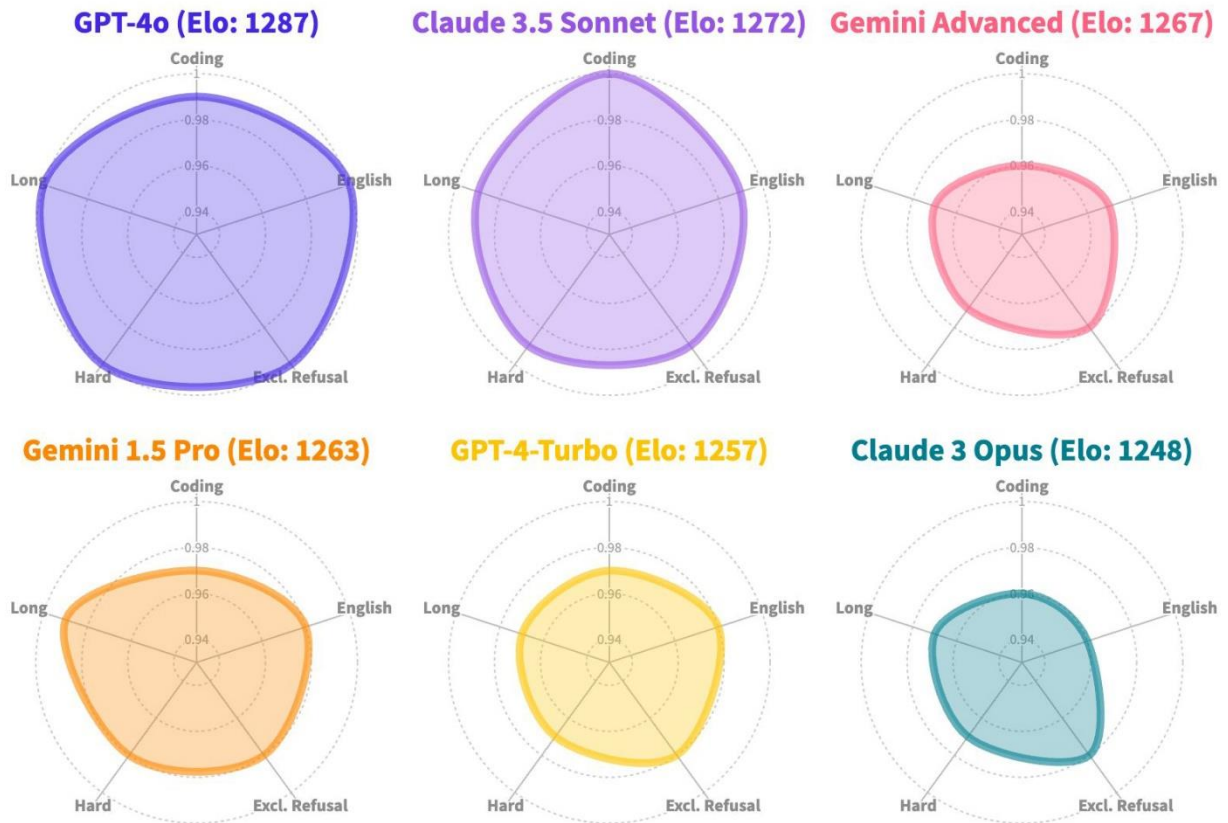


Рисунок 3. Порівняння продуктивності LLM моделей

На рисунку 3 зображено діаграми особливості кожної мовної моделі. GPT-4 демонструє найвищий показник Elo (1287), що свідчить про її універсальність та високу продуктивність у всіх представлених категоріях. Claude 3.5 Sonnet (Elo: 1272) наближається до результатів GPT-4, проте має невеликі відхилення у продуктивності, особливо в задачах із довгими текстами. Gemini Advanced (Elo: 1267) та Gemini 1.5 Pro (Elo: 1263) показують схожу ефективність, але з меншою гнучкістю в роботі з кодуванням та обробкою складних завдань, порівняно з Claude та GPT. GPT-4 Turbo (Elo: 1257) відрізняється оптимізацією швидкодії, але не досягає такого рівня якості, як стандартний GPT-4. Claude 3 Opus (Elo: 1248) демонструє хорошу продуктивність, однак поступається іншим моделям у здатності працювати з довгими текстами та складними задачами. Загалом, зображення наочно ілюструє сильні сторони кожної моделі та допомагає зрозуміти їхню ефективність у різних аспектах.

1.3. Імплементация елементів штучного інтелекту для автоматизації робочих процесів підприємств

Імплементация штучного інтелекту (ШІ) для автоматизації робочих процесів підприємств відкриває широкі можливості для підвищення ефективності та оптимізації багатьох бізнес-операцій. ШІ стає центральним елементом сучасних підприємств, дозволяючи автоматизувати як рутинні завдання, так і складні аналітичні процеси. Одним із важливих аспектів є його інтеграція в управління бізнес-процесами, де ШІ може сприяти оптимізації документообігу, бухгалтерського обліку та HR-процедур. Використовуючи потужності машинного навчання, такі системи автоматично обробляють великі обсяги інформації, зокрема рахунки чи кадрові дані, тим самим значно зменшуючи потребу в людському втручанні і підвищуючи швидкість та точність обробки.

Згідно з [9] глобальний світовий ВВП від використання інтелектуальних технологій складає 15,73 трлн.дол. Водночас, згідно прогнозів експертів, цей показник до 2030р. зросте ще мінімум на 14% власне завдяки штучному інтелекту. Таким чином, впровадження штучного інтелекту стає ключовим драйвером економічного зростання, що значно посилює продуктивність та інноваційний потенціал різних галузей. Очікуване збільшення глобального ВВП свідчить про масштабний вплив інтелектуальних технологій на трансформацію світової економіки.

Автоматизація процесів на основі роботизованих технологій ШІ дозволяє знизити навантаження на персонал і зменшити ймовірність людських помилок. ШІ може бути використаний для виконання таких завдань, як введення даних або оновлення записів, що не тільки підвищує продуктивність, але й сприяє ефективнішому управлінню ресурсами підприємства. Це дозволяє співробітникам зосереджуватися на стратегічних завданнях, залишаючи рутинні операції на автоматизовані системи.

Не менш важливою є роль штучного інтелекту в аналітиці та прогнозуванні. Здатність ШІ аналізувати великі обсяги даних та виявляти

приховані закономірності дозволяє підприємствам більш точно прогнозувати попит, фінансові ризики або ефективність виробничих процесів. Така аналітика стає критично важливою в умовах конкурентного ринку, де швидкість реакції на зміни може визначати успіх або невдачу бізнесу. Прогнозування на основі машинного навчання також надає можливість для оптимізації управління ланцюгами постачання, дозволяючи знизити витрати і покращити взаємодію з постачальниками.

Штучний інтелект також докорінно змінює взаємодію підприємств з клієнтами. Завдяки можливостям обробки природної мови, підприємства можуть використовувати чат-боти для автоматизації спілкування з клієнтами, що не тільки прискорює процес вирішення запитів, але й дозволяє запропонувати клієнтам персоналізовані послуги або продукти. Системи рекомендацій на основі ШІ здатні аналізувати попередні покупки клієнтів або їхні запити для надання індивідуальних пропозицій, що сприяє підвищенню задоволеності клієнтів і їхньої лояльності до компанії.

Важливою сферою для впровадження штучного інтелекту є управління ланцюгами постачання, де алгоритми ШІ можуть аналізувати складні взаємозв'язки між попитом і пропозицією, оптимізуючи запаси та доставку продукції. Завдяки можливостям аналізу різноманітних факторів, включаючи погодні умови та ринкові зміни, такі системи допомагають приймати більш зважені рішення щодо планування постачання та оптимізації витрат. Згідно з [10] Автоматизація у логістиці не тільки знижує оперативні витрати та збільшує безпеку на робочих місцях, але й відкриває нові можливості для оптимізації бізнес-процесів. Впровадження автоматизованих систем зберігання та внутрішньоскладської логістики, забезпечує компаніям стратегічну перевагу у конкурентній боротьбі на ринку.

Крім того, ШІ активно застосовується у виробничих процесах для моніторингу якості продукції та контролю за обладнанням. Використання алгоритмів машинного навчання та комп'ютерного зору дозволяє виявляти дефекти на ранніх стадіях виробництва, що знижує кількість браку і підвищує

ефективність. Планування виробничих процесів також може бути автоматизованим за допомогою ШІ, що дає змогу зменшити простір обладнання і оптимізувати використання виробничих потужностей.

Питання безпеки підприємств також не залишаються без уваги. Згідно з [11] Способами застосування ШІ для захисту від кіберзагроз є виявлення аномалій у мережі, аналіз поведінкових моделей користувачів, розпізнавання атак на основі машинного навчання, автоматизоване відновлення після інцидентів (рис. 4). ШІ аналізує нормальну діяльність, щоб швидко виявити та реагувати на відхилення, які можуть бути індикаторами кібератак. Аналіз поведінкових моделей користувачів дозволяє відрізнити нормальну діяльність від підозрілої, навіть якщо кіберзлочинці використовують інноваційні способи атак. У міру розвитку загроз у сфері кібербезпеки, системи ШІ забезпечують підвищену адаптивність до нових видів атак, тим самим підвищуючи загальний рівень захисту даних підприємства.

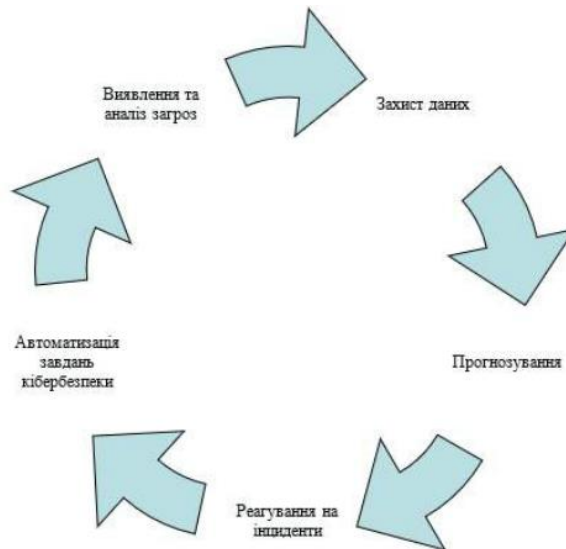


Рисунок 4. Демонстрація застосування штучного інтелекту в кібербезпеці

Штучний інтелект сприяє ухваленню рішень у реальному часі. Використання інтелектуальних систем дозволяє автоматизувати процеси аналізу даних і надавати керівництву підприємства актуальну інформацію для стратегічних рішень. Це може бути особливо корисно у фінансовій сфері для управління ризиками або у виробничих процесах для оптимізації

завантаження обладнання. Такі системи допомагають знизити невизначеність та мінімізувати ризики в умовах швидкозмінного ринку.

У контексті маркетингових процесів ШІ дозволяє підприємствам не лише аналізувати поведінку споживачів, але й персоналізувати рекламні стратегії, підвищуючи їхню ефективність. За допомогою машинного навчання підприємства можуть прогнозувати, які продукти чи послуги будуть найбільш цікавими для клієнтів, що дозволяє значно підвищити ефективність рекламних кампаній і оптимізувати витрати на маркетинг.

На стратегічному рівні, алгоритми штучного інтелекту допомагають компаніям у розробці довгострокових планів розвитку та виходу на нові ринки. ШІ може аналізувати внутрішні дані компанії, ринки та конкурентів, щоб надавати обґрунтовані рекомендації для ухвалення рішень щодо інвестицій, розвитку нових продуктів та бізнес-стратегій. Це сприяє підвищенню конкурентоспроможності компанії та забезпечує її гнучкість в умовах глобальних змін.

Отже, впровадження штучного інтелекту в робочі процеси підприємств забезпечує комплексну оптимізацію багатьох аспектів бізнесу. Його застосування дозволяє підприємствам підвищити ефективність, знизити витрати, покращити якість обслуговування клієнтів і посилити захист від кіберзагроз, що робить ШІ ключовим елементом успішної автоматизації сучасних підприємств.

Розділ 2. Методи впровадження елементів штучного інтелекту в робочі процеси підприємств

Впровадження елементів штучного інтелекту (ШІ) у робочі процеси підприємств стає важливою складовою їхньої модернізації та підвищення конкурентоспроможності. Основними методами інтеграції ШІ є автоматизація рутинних завдань, аналітичне прогнозування та адаптивні системи управління. Використання автоматизації дозволяє оптимізувати повторювані процеси, такі як обробка даних, сортування інформації та контроль за виконанням стандартних операцій. Завдяки цьому підприємства можуть знизити витрати часу і ресурсів, спрямовуючи зусилля на стратегічний розвиток.

Аналітичні можливості ШІ забезпечують підприємства потужними інструментами для аналізу великих обсягів даних. Завдяки методам машинного навчання можна виявляти приховані закономірності, оцінювати ризики та передбачати ринкові тенденції. Наприклад, в системах управління запасами ШІ допомагає передбачати попит на продукцію, запобігаючи надмірному накопиченню або дефіциту товарів. Таким чином, аналітичне прогнозування сприяє підвищенню ефективності процесів планування.

Адаптивні системи управління, що базуються на ШІ, дозволяють підприємствам оперативно реагувати на зміни зовнішнього середовища. Такі системи забезпечують автоматичне налаштування параметрів виробничих і логістичних процесів залежно від поточних умов. Наприклад, у сфері енергетики ШІ використовується для адаптивного управління енергоспоживанням, що дозволяє знижувати витрати і зменшувати вплив на довкілля. Інтеграція таких систем дає можливість підвищити гнучкість і адаптивність бізнес-процесів.

Важливою складовою впровадження ШІ є етап навчання систем та їх тестування у реальних умовах. Для цього підприємства застосовують пілотні проєкти, в рамках яких проводиться перевірка ефективності обраних рішень та їх адаптація до специфіки роботи компанії. Успішне виконання пілотних

проектів створює основу для масштабування технологій ШІ на інші аспекти діяльності.

2.1. Аналіз робочих процесів комунікаційного відділу по роботі з клієнтами підприємств сфери обслуговування

Штучний інтелект (ШІ) стає ключовим інструментом для оптимізації бізнес-процесів у різних галузях, сприяючи підвищенню ефективності, зниженню витрат та покращенню якості продукції та послуг. Його впровадження дозволяє підприємствам адаптуватися до сучасних викликів та залишатися конкурентоспроможними на ринку.

У виробничій сфері ШІ активно використовується для контролю якості та автоматизації процесів. Згідно з [12] компанія BMW впровадила системи штучного інтелекту на своєму заводі в Спартанбурзі, США, для контролю якості виробництва автомобілів. Ці системи аналізують зображення компонентів у реальному часі, виявляючи можливі дефекти та відхилення від стандартів, що дозволяє оперативно виправляти помилки та знижувати витрати на виробництво. Завдяки цьому компанія економить понад 1 мільйон доларів на рік.

У логістиці та управлінні ланцюгами постачання ШІ допомагає оптимізувати маршрути доставки, прогнозувати попит та управляти запасами. Компанія DHL використовує алгоритми машинного навчання для аналізу великих обсягів даних, що дозволяє ефективніше планувати маршрути, знижувати витрати на паливо та покращувати обслуговування клієнтів. Зокрема, ШІ допомагає прогнозувати попит на послуги доставки та оптимізувати розподіл ресурсів, що підвищує загальну ефективність логістичних процесів.

У фінансовому секторі ШІ застосовується для виявлення шахрайських операцій, управління ризиками та автоматизації обслуговування клієнтів. Банки використовують алгоритми машинного навчання для аналізу транзакцій у реальному часі, що дозволяє виявляти підозрілі операції та запобігати

шахрайству. Крім того, чат-боти на основі ШІ забезпечують клієнтам цілодобову підтримку, відповідаючи на запити та надаючи консультації з фінансових питань. Це підвищує рівень задоволеності клієнтів та знижує навантаження на персонал.

У сфері охорони здоров'я ШІ використовується для діагностики захворювань, аналізу медичних зображень та розробки індивідуальних планів лікування. Медичні заклади впроваджують системи штучного інтелекту, які аналізують результати обстежень та допомагають лікарям у постановці точних діагнозів. Наприклад, алгоритми ШІ можуть виявляти ознаки раку на ранніх стадіях, що підвищує шанси на успішне лікування. Також ШІ допомагає оптимізувати розподіл ресурсів у лікарнях та прогнозувати потреби в медичних послугах.

Штучний інтелект (ШІ) у сфері обслуговування, особливо у кафе та ресторанах, сприяє значній трансформації бізнес-процесів, підвищуючи ефективність, знижуючи витрати і покращуючи клієнтський досвід. Інтеграція технологій ШІ у цій галузі дозволяє оптимізувати роботу персоналу, забезпечити персоналізований сервіс і вдосконалити управління ресурсами.

Одним із ключових напрямів використання ШІ є автоматизація процесів прийому замовлень. Наприклад, інтерактивні меню на основі штучного інтелекту, які доступні через мобільні додатки або інформаційні кіоски в закладах, дозволяють клієнтам самостійно обирати страви, знижуючи навантаження на офіціантів. Такі системи також можуть рекомендувати страви на основі уподобань клієнта або історії замовлень, створюючи персоналізований досвід. Згідно з [14] мережа ресторанів McDonald's активно використовує технології ШІ у своїх терміналах самообслуговування, які пропонують клієнтам додаткові товари на основі попередніх виборів та часу доби.

ШІ також ефективно впроваджується у процеси управління запасами та постачанням. Алгоритми машинного навчання аналізують дані про продажі, сезонність і попит, щоб прогнозувати потреби у продуктах та мінімізувати

харчові відходи. Згідно з [15] компанія Domino's Pizza використовує ШІ для оптимізації запасів на основі аналізу замовлень, погодних умов та святкових днів, що дозволяє уникнути нестачі або перевитрат ресурсів.

Таким чином, впровадження ШІ у кафе та ресторанах дозволяє автоматизувати процеси, покращити обслуговування клієнтів та оптимізувати управління ресурсами, що сприяє підвищенню конкурентоспроможності закладів у сучасних умовах.

У магістерській роботі взято за мету створити систему аналізу роботи комунікаційного відділу закладу харчування. Проект аналізу роботи кол-центру для закладу громадського харчування спрямований на покращення якості обслуговування клієнтів шляхом автоматизованого аналізу телефонних розмов. Основною метою розробки є оцінка того, наскільки коректно оператори гарячої лінії взаємодіють із клієнтами, чи дотримуються вони стандартів ввічливості та чи вирішують усі проблеми, з якими звертаються споживачі. Це дозволить підвищити рівень задоволеності клієнтів, оптимізувати роботу комунікаційного відділу та забезпечити відповідність наданих послуг високим стандартам якості.

Технічне завдання проекту передбачає створення системи, яка здійснює автоматизований аналіз записів телефонних розмов. Система має використовувати алгоритми обробки природної мови (NLP) для розпізнавання тексту з аудіозаписів і подальшої оцінки за визначеними критеріями. Основними функціональними компонентами проекту є модуль для розпізнавання голосу, аналітичний модуль для визначення ключових фраз і тональності розмови, а також модуль генерації звітів для управлінського персоналу. Алгоритми аналізу будуть побудовані з урахуванням специфіки мови та емоційної складової розмови.

Розробка базується на попередньо визначених скриптах роботи комунікаційного відділу, які є основою для оцінки якості обслуговування. У процесі взаємодії оператори зобов'язані дотримуватися затверджених протоколів, що включають порядок привітання, уточнення проблеми клієнта,

надання інформації чи допомоги, а також завершення розмови з підведенням підсумків. Ефективність роботи оцінюється за кількома критеріями, зокрема відповідністю сценарію, емоційною ввічливістю, швидкістю вирішення запиту та задоволеністю клієнта.

Процеси, які здійснюються інфокомунікаційним відділом, безпосередньо впливають на репутацію закладу громадського харчування, адже саме цей підрозділ є точкою першого контакту клієнта з компанією. Від того, наскільки оперативно і якісно оператор вирішує проблему клієнта, залежить не лише збереження лояльності, але й створення позитивного іміджу бренду в цілому. Крім того, аналіз телефонних розмов дозволяє виявляти типові проблеми, з якими стикаються клієнти, та адаптувати внутрішні бізнес-процеси для їх усунення.

Результати роботи системи аналізу передбачають формування комплексних звітів для керівництва, які міститимуть як індивідуальні оцінки роботи операторів, так і загальні показники ефективності відділу. На основі цих звітів керівництво зможе приймати обґрунтовані рішення щодо корекції скриптів, навчання персоналу та впровадження додаткових інструментів для покращення обслуговування клієнтів.

Аналіз роботи комунікаційного відділу закладу громадського харчування на основі скриптів і заданих критеріїв дозволяє оцінити ефективність роботи менеджерів, забезпечити дотримання стандартів якості та виявити шляхи для вдосконалення. Система аналізу базується на структурованому підході до оцінки розмов, що охоплює такі аспекти, як структура діалогу, активне слухання, пропозиція супутніх товарів, робота із запереченнями, задоволення клієнтів та аналіз емоційного стану.

Ключовим елементом є структура розмови, яка визначає логічний порядок комунікації менеджера з клієнтом. Початок розмови включає привітання та представлення компанії, що встановлює професійний тон і створює позитивне враження. Наприклад, звернення на кшталт «Доброго дня, мене звати [ім'я]. Як я можу вам допомогти?» формує основу для довіри та

відкритого спілкування. Подальше виявлення потреб клієнта через відкриті запитання дозволяє зібрати необхідну інформацію для подальшої пропозиції відповідних товарів або послуг. Завершення розмови включає підсумок, подяку та пропозицію додаткової допомоги, що сприяє завершенню контакту на позитивній ноті.

Активне слухання є важливою складовою успішного спілкування. Менеджери мають підтверджувати розуміння клієнтських запитів через повторення або перефразування ключових моментів. Наприклад, уточнення «Отже, вам потрібно доставити обід на 14:00, правильно?» демонструє увагу та розуміння потреб клієнта. Уникнення перебивань і використання підтверджувальних фраз, таких як «Розумію вас», сприяють створенню комфортної атмосфери для клієнта.

Пропозиція супутніх товарів і допродажі мають вирішальне значення для збільшення середнього чеку. Використання підходів cross-sell і up-sell дозволяє менеджерам пропонувати клієнтам додаткові товари, які доповнюють їхнє основне замовлення. Наприклад, пропозиція десерту зі знижкою до обіду або більшого набору товарів за вигідною ціною є ефективними стратегіями, що підвищують доходи закладу та задовольняють потреби клієнтів.

Робота із запереченнями вимагає від менеджера не лише впевненості, але й вміння підкреслити переваги послуг чи продуктів. Використання технік, таких як метод «сендвіча», допомагає знизити рівень невдоволення клієнта, наприклад: «Я розумію, що ціна може здаватися високою, але наші продукти мають відмінну якість, яка вас задовольнить». Такий підхід дозволяє не лише розв'язати заперечення, але й зміцнити впевненість клієнта у правильності його вибору.

Задоволення клієнта та післяпродажне обслуговування є завершальним етапом, який впливає на формування лояльності. Проактивний контакт після покупки, запит на зворотний зв'язок і оперативна реакція на скарги формують у клієнта відчуття турботи та уваги до його потреб. Наприклад,

зателефонувати після доставки замовлення з уточненням, чи все було доставлено вчасно та чи задоволений клієнт, є прикладом такого підходу.

Система аналізу розмов за вказаними критеріями дає можливість не лише виявити сильні та слабкі сторони роботи менеджерів, але й надавати конструктивні рекомендації для покращення. Визначення рівня відповідності стандартам через оцінки за шкалою дозволяє відстежувати прогрес і забезпечувати безперервний розвиток персоналу. Таким чином, використання структурованих скриптів та оцінювання роботи менеджерів сприяє підвищенню якості обслуговування, що є важливим фактором у конкурентному середовищі закладів громадського харчування.

2.2. Вибір комп'ютерних технологій для реалізації застосунку для автоматизації якості оцінювання роботи працівників комунікаційного відділу по роботі з клієнтами

Розробка застосунку для автоматизації оцінювання якості роботи працівників комунікаційного відділу вимагає ретельного підходу до вибору комп'ютерних технологій. Це стосується не лише вибору мов програмування, але й забезпечення їхньої відповідності технічним вимогам проекту, таким як обробка тексту, взаємодія через API та інтеграція з платформами штучного інтелекту. Для серверного застосунку, який взаємодіє з зовнішніми API для завантаження, транскрибування та аналізу даних, вирішальними критеріями є продуктивність, легкість розробки, підтримка бібліотек і модулів, а також можливість масштабування.

Однією з ключових задач при виборі мови програмування є оцінка її придатності для роботи зі штучним інтелектом та API. У цьому контексті Python виявляється найбільш оптимальним варіантом завдяки своїм численним перевагам. Перш за все, Python є мовою з широким спектром бібліотек і модулів, які забезпечують інтеграцію зі штучним інтелектом та обробкою природної мови, наприклад, OpenAI, SpeechRecognition, Transformers та інші. Python має простий синтаксис, що значно знижує поріг

входу для розробників і прискорює процес розробки. Це є особливо важливим для проектів, які мають суворі часові рамки.

Інші мови програмування, такі як Java, C++ або JavaScript, також можуть бути використані для розробки серверних застосунків. Наприклад, Java характеризується високою продуктивністю, а її екосистема дозволяє працювати з великими обсягами даних і складними обчисленнями. C++ забезпечує надзвичайно високу продуктивність, але значно ускладнює процес розробки через складність синтаксису та необхідність врахування деталей низькорівневої роботи. JavaScript, будучи популярною мовою для веб-розробки, також може бути використаний для серверної частини через середовище Node.js, але його можливості у сфері обробки природної мови та інтеграції з ШІ обмежені у порівнянні з Python.

Порівняльний аналіз мов програмування демонструє, що Python є найбільш придатним вибором для цього проекту завдяки його гнучкості, багатству бібліотек та активній спільноті. У таблиці 1 наведено основні критерії для оцінки мов програмування та їх переваги.

Таблиця 2.1 – Порівняння мов програмування

Критерій	Python	Java	C++	JavaScript (Node.js)
Простота розробки	Дуже висока	Середня	Низька	Висока
Підтримка бібліотек для ШІ	Дуже широка	Обмежена	Обмежена	Обмежена
Інтеграція через API	Проста	Проста	Складна	Проста
Продуктивність	Середня	Висока	Дуже висока	Середня
Масштабованість	Висока	Дуже висока	Дуже висока	Висока
Складність підтримки коду	Низька	Середня	Висока	Низька

Python виділяється своєю простотою у використанні, розвиненими можливостями інтеграції з API та багатством бібліотек для роботи з обробкою тексту та природної мови. Хоча Java та C++ демонструють переваги у продуктивності, складність їхнього використання та менша підтримка сучасних бібліотек III роблять їх менш придатними для швидкої реалізації цього проекту. JavaScript підходить для створення веб-інтерфейсів, але його можливості у сфері машинного навчання значно обмежені.

У процесі реалізації застосунку для автоматизації оцінювання якості роботи працівників комунікаційного відділу було використано кілька ключових бібліотек Python. Вони забезпечують ефективну взаємодію із зовнішніми сервісами, обробку запитів через API та інтеграцію з платформами штучного інтелекту. Серед них особливе місце займають бібліотеки `openai` і `requests`.

Бібліотека `openai` є офіційним інструментом для взаємодії з мовними моделями, розробленими OpenAI, такими як GPT-4. Вона надає зручний інтерфейс для виконання запитів до моделей, що дозволяє швидко інтегрувати їх у різноманітні проекти, включно із задачами обробки природної мови.

До переваг цієї бібліотеки можна віднести:

- Бібліотека має інтуїтивно зрозумілий синтаксис, що дозволяє легко надсилати запити до мовної моделі та отримувати відповіді у форматі JSON.
- Вона дозволяє налаштовувати параметри запиту, такі як довжина відповіді, рівень креативності (`temperature`) та обмеження використання токенів.
- Завдяки підтримці різних типів запитів, `openai` може бути використана для генерації тексту, аналізу діалогів, резюмування тексту та інших задач.
- Бібліотека забезпечує зручну роботу з API ключами та включає механізми для обробки помилок у запитах, що підвищує надійність застосунку.

У цьому проекті `openai` має використовуватись для надсилання тексту розмов, транскрибованого за допомогою стороннього сервісу, до мовної

моделі GPT. Відповідь моделі містить аналіз якості розмови, який використовується для подальшого оцінювання роботи працівників.

Бібліотека `requests` є одним із найпоширеніших інструментів у Python для роботи з HTTP-запитами. Вона використовується для взаємодії з API зовнішніх сервісів, забезпечуючи простий і зрозумілий спосіб надсилання GET, POST, PUT та DELETE-запитів.

Ключові особливості та переваги це:

- Простий синтаксис дозволяє легко формувати запити та обробляти відповіді у форматі JSON, XML або текст.
- Бібліотека підтримує налаштування заголовків, параметрів запиту, аутентифікації, таймаутів та проксі, що робить її універсальною для різних сценаріїв.
- Вбудовані механізми дозволяють обробляти помилки HTTP-запитів, такі як таймаути або відмови у доступі.
- Вона легко інтегрується з іншими бібліотеками та модулями Python, що робить її зручним вибором для створення серверних застосунків.

Поєднання бібліотек `openai` та `requests` дозволяє ефективно реалізувати серверний застосунок, який інтегрується із зовнішніми сервісами для обробки розмов та оцінювання якості роботи працівників. `openai` забезпечує роботу зі штучним інтелектом, тоді як `requests` відповідає за надійність і гнучкість обміну даними через API, створюючи фундамент для автоматизованої системи оцінювання.

Розділ 3. Особливості архітектури серверного застосунку

Архітектура серверних застосунків є фундаментом для забезпечення надійності, продуктивності та масштабованості сучасних інформаційних систем. Існує кілька основних типів архітектури, які використовуються залежно від завдань і специфіки проєкту, кожна з яких має свої переваги й обмеження. Серед найпоширеніших архітектур виділяють монолітну, клієнт-серверну, мікросервісну, серверлесс і подієво-орієнтовану архітектури.

Монолітна архітектура є однією з найстаріших і найпростішою у плануванні та реалізації моделей побудови програмного забезпечення. У межах цієї архітектури всі компоненти застосунку об'єднуються в єдиний цілісний модуль, який виконується як єдиний процес. Цей підхід є популярним для невеликих проєктів та застосунків, що мають стабільну функціональність і не потребують частого масштабування.

Монолітна архітектура передбачає, що всі компоненти системи, такі як інтерфейс користувача, бізнес-логіка та база даних, інтегровані в один виконуваний модуль. Це забезпечує тісну взаємодію між компонентами, зменшуючи затримки у комунікації. Для запуску монолітного застосунку зазвичай достатньо єдиного серверного середовища.

В основі монолітної архітектури лежить концепція єдиного кодового базису, що спрощує управління залежностями та зменшує кількість технічних інтеграцій. Вона зазвичай добре працює з традиційними моделями розробки, які передбачають чіткий життєвий цикл розробки, починаючи з визначення вимог, і завершуючи впровадженням.

До переваг використання такої архітектури можна віднести:

- Монолітна архітектура є відносно простою у впровадженні, оскільки не вимагає складного налаштування для комунікації між компонентами. Усі частини системи працюють в одному середовищі, що зменшує складність розгортання.

- Усі компоненти інтегровані, тому розробники можуть легко тестувати функціональність застосунку в єдиному середовищі. виправлення помилок не потребує оновлення багатьох окремих сервісів.
- Для невеликих проєктів моноліт дозволяє швидко перейти від початкової концепції до робочого прототипу, що особливо корисно для стартапів і пілотних проєктів.
- Завдяки тому, що застосунок запускається як єдиний модуль, немає потреби налаштовувати окремі середовища для кожного сервісу, як у випадку мікросервісів.
- Єдина точка входу для клієнтів спрощує управління аутентифікацією, доступом до ресурсів і загальною архітектурою додатку.

До недоліків використання такої системи відносяться:

- Усі компоненти застосунку масштабуються одночасно, незалежно від того, чи всі з них цього потребують. Це призводить до надмірного використання ресурсів.
- У міру зростання застосунку внесення змін стає складнішим, оскільки будь-яка модифікація може впливати на всі інші компоненти.
- Оскільки всі частини системи інтегровані, помилка в одній частині може викликати збій усього застосунку.
- Зі збільшенням розміру застосунку, кодова база стає важкою для розуміння та підтримки, що вимагає значних витрат часу та ресурсів.
- У разі змін чи додавання функцій потрібно перезапустити весь застосунок, що може призвести до простоїв.

Монолітна архітектура є популярним вибором для невеликих бізнесів, стартапів або додатків з чітко визначеною функціональністю. Вона також підходить для внутрішніх корпоративних систем, які не потребують значного обсягу інтеграцій з іншими сервісами.

Наприклад, система управління взаємовідносинами з клієнтами (CRM) для малого бізнесу, створена як монолітний застосунок, може ефективно

обробляти всі ключові функції: зберігання клієнтських даних, управління замовленнями та створення звітів.

Клієнт-серверна архітектура є одним із класичних підходів у проєктуванні інформаційних систем, що забезпечує чіткий розподіл функцій між двома основними учасниками: клієнтом та сервером. Цей підхід став основою для багатьох сучасних веб-систем і додатків, забезпечуючи ефективність роботи, гнучкість у розробці та масштабованість.

Клієнт-серверна модель базується на взаємодії двох сторін. Клієнт — це програма або пристрій, який ініціює запит до сервера для отримання певної послуги, наприклад, доступу до даних або виконання обчислень. Сервер — це потужний обчислювальний ресурс, який відповідає за обробку запитів і повернення відповідей клієнту.

Ключовою особливістю цієї архітектури є централізація обробки даних. Сервери зазвичай керують базами даних, виконують складні обчислення та забезпечують функціонування основної бізнес-логіки, тоді як клієнтська сторона відповідає за відображення даних та взаємодію з користувачем.

Перевагами використання клієнт-серверної архітектури вважаються:

- Вся інформація зберігається та обробляється на сервері, що забезпечує високий рівень контролю над нею. Це особливо важливо для систем, які працюють із конфіденційними даними, наприклад, банківські додатки.
- Сервер може обслуговувати одночасно велику кількість клієнтів, і за потреби до нього можна додавати нові ресурси (наприклад, більше процесорної потужності або пам'яті).
- Централізація дозволяє легко контролювати доступ до даних, застосовувати політики безпеки та протидіяти загрозам.
- Клієнт-серверна архітектура дозволяє використовувати різні типи клієнтських пристроїв, такі як комп'ютери, смартфони чи планшети, що забезпечує широку доступність для користувачів.
- Оновлення сервера дозволяє миттєво внести зміни в систему без необхідності модифікації клієнтських програм.

Серед недоліків є:

- Сервер є критичною точкою системи: якщо він виходить з ладу, то вся система перестає функціонувати.
- Велика кількість запитів від клієнтів може призвести до перевантаження сервера, що потребує додаткового масштабування інфраструктури.
- Оскільки клієнт взаємодіє із сервером через мережу, час відповіді залежить від пропускної здатності мережі та її стабільності.
- Сервер може стати об'єктом кібератак, таких як DDoS-атаки, що може призвести до тимчасової недоступності послуг.
- У разі відсутності стабільного мережевого підключення клієнти не зможуть отримувати доступ до сервера.

Ця архітектура є основою для багатьох сучасних систем, таких як веб-сайти, інтернет-магазини, банківські платформи, CRM-системи та хмарні сервіси. Наприклад, інтернет-магазин використовує клієнт-серверну модель для взаємодії між клієнтським браузером, сервером баз даних та сервером обробки замовлень. Клієнт переглядає товари, надсилає замовлення, а сервер обробляє транзакції та забезпечує доставку.

Мікросервісна архітектура — це сучасний підхід до побудови програмних систем, який ґрунтується на розподілі системи на невеликі, незалежні сервіси, кожен з яких відповідає за виконання окремої функції. Цей підхід дозволяє підвищити гнучкість, продуктивність і масштабованість системи, що є особливо важливим для складних і динамічних проєктів.

Мікросервіси — це автономні компоненти, які виконують окремі функції. Вони взаємодіють між собою через чітко визначені інтерфейси, зазвичай за допомогою API або повідомлень. Кожен мікросервіс може бути розроблений, розгорнутий і масштабований незалежно, що дозволяє розробникам зосереджуватися на конкретних завданнях без потреби змінювати всю систему.

Головна особливість мікросервісної архітектури — це децентралізація. Кожен сервіс може мати власну базу даних і бути написаний на різних мовах

програмування, якщо це відповідає його завданням. Це дозволяє використовувати найкращі технології для вирішення конкретних проблем.

Серед переваг такої архітектури можна віднести:

- Кожен мікросервіс може бути масштабований незалежно, залежно від його навантаження. Це дозволяє оптимально використовувати ресурси.
- Команди розробників можуть працювати над різними сервісами без необхідності враховувати весь проєкт. Це пришвидшує розробку та впровадження змін.
- Помилка в одному мікросервісі не впливає на роботу інших. Це підвищує надійність системи загалом.
- Оскільки мікросервіси є автономними, вони можуть використовувати різні технології, які найкраще підходять для конкретного завдання.
- Оновлення одного мікросервісу не вимагає змін у всій системі, що зменшує ризики під час впровадження нових функцій.

А недоліками вважаються:

- Збільшення кількості мікросервісів ускладнює управління взаємодією між ними, що потребує додаткових зусиль для моніторингу, логування та забезпечення безпеки.
- Оскільки сервіси взаємодіють через мережу, це може спричинити затримки, особливо при інтенсивному обміні даними.
- Кожен мікросервіс вимагає власного середовища виконання, що збільшує витрати на сервери та налаштування.
- Перевірка коректної взаємодії між великою кількістю мікросервісів може бути складною, особливо якщо між ними використовуються асинхронні механізми зв'язку.
- Якщо кожен сервіс має власну базу даних, це може створювати складнощі у забезпеченні цілісності та актуальності даних.

Мікросервісна архітектура найбільш ефективна для великих і складних систем, які повинні обслуговувати значну кількість користувачів або обробляти великі обсяги даних. Наприклад, такі платформи, як Netflix,

Amazon і eBay, активно використовують мікросервіси для забезпечення стабільності та швидкості своїх систем.

Наприклад, у випадку e-commerce платформи, один мікросервіс може відповідати за управління каталогом товарів, інший — за обробку платежів, ще один — за обробку замовлень. Це дозволяє масштабувати лише ті сервіси, які зазнають найбільшого навантаження, наприклад, під час сезонних розпродажів.

Подієво-орієнтована архітектура (Event-Driven Architecture, EDA) є підходом до побудови програмних систем, який базується на реакції на події. У цій моделі система складається з компонентів, що спілкуються між собою через повідомлення про події. Подія може бути будь-якою зміною стану системи, наприклад, додавання нового замовлення, натискання кнопки користувачем або отримання сигналу від зовнішнього сенсора.

Подієво-орієнтована архітектура побудована навколо концепції обміну подіями між різними компонентами системи. Основними елементами цього підходу є відправники подій, обробники подій, шина подій і сховище подій. Відправник подій (Event Producer) відповідає за генерацію подій, які є реакцією на певні дії або зміни стану системи. Наприклад, система моніторингу може відправляти подію про перевищення допустимого рівня температури.

Обробник подій (Event Consumer) отримує ці події та виконує відповідні дії, такі як надсилання сповіщень, оновлення стану системи або виконання бізнес-логіки. Для передачі подій між відправниками та обробниками використовується шина подій (Event Bus) — спеціальна інфраструктура, яка забезпечує маршрутизацію повідомлень. Вона може бути реалізована через черги повідомлень, наприклад, Apache Kafka або RabbitMQ, що дозволяє системі працювати асинхронно і забезпечувати високу продуктивність.

Ще одним важливим елементом подієво-орієнтованої архітектури є сховище подій (Event Store), яке використовується для зберігання всіх подій системи. Це забезпечує можливість відстежувати історію змін, аналізувати

минулі дії або відновлювати стан системи у разі необхідності. Така організація взаємодії між компонентами дозволяє створювати динамічні, гнучкі системи, які можуть швидко реагувати на зміни у навколишньому середовищі чи вимоги користувачів.

Переваги подієво-орієнтованої архітектури:

- Компоненти системи працюють незалежно один від одного, обробляючи події у своєму темпі. Це дозволяє зменшити затримки та забезпечити високу продуктивність.
- Нові обробники подій можуть бути додані без змін існуючих компонентів, що робить систему легко масштабованою.
- У разі збою одного з компонентів інші можуть продовжувати працювати, що підвищує надійність системи.
- Завдяки запису подій, система зберігає історію своїх змін, що дозволяє аналізувати дії користувачів і поведінку системи.
- Подієво-орієнтована архітектура полегшує інтеграцію різнорідних систем, оскільки всі вони можуть реагувати на події через загальний канал.

Недоліки подієво-орієнтованої архітектури:

- Взаємодія через події ускладнює дизайн системи та потребує розуміння асинхронних процесів.
- Система вимагає надійної та продуктивної шини подій, а її налаштування та обслуговування можуть бути складними.
- Відстеження подій і аналіз їхньої взаємодії можуть бути складними через розподілений характер системи.
- Якщо обробник не працює належним чином, подія може бути оброблена кілька разів або втрачена, що потребує механізмів перевірки.

Подієво-орієнтована архітектура найкраще підходить для систем, які потребують високоасинхронного зв'язку, таких як IoT (Інтернет речей), системи моніторингу, обробка потоків даних або мікросервісні архітектури.

Наприклад, у сфері електронної комерції така архітектура дозволяє реагувати на події, як-от оформлення замовлення, оновлення складу чи надсилання сповіщень про стан доставки. Інший приклад — система фінансового моніторингу, яка реагує на підозрілі транзакції в режимі реального часу.

Компонентно-орієнтована архітектура — це підхід до побудови програмних систем, що базується на розділенні застосунку на окремі незалежні компоненти. Кожен компонент виконує чітко визначену функцію і взаємодіє з іншими через стандартизовані інтерфейси. Ця архітектура спрямована на модульність, гнучкість та легкість у підтримці, що робить її особливо актуальною для систем, які потребують постійного розвитку або інтеграції з іншими сервісами.

Компоненти в системі є незалежними блоками, які можуть бути замінені, оновлені або масштабовані без значного впливу на інші частини системи. Вони можуть реалізовувати бізнес-логіку, працювати з даними або взаємодіяти з зовнішніми сервісами. Взаємодія між компонентами зазвичай здійснюється через API або події, що дозволяє зберігати слабку зв'язність між частинами системи.

Ця архітектура часто використовує подієво-орієнтований підхід для асинхронного обміну даними між компонентами, а також стандартні протоколи, такі як HTTP або gRPC, для синхронної комунікації. Вона підтримує інтеграцію зовнішніх сервісів, таких як API, бази даних або хмарні платформи, забезпечуючи масштабованість і адаптивність системи.

Компонентно-орієнтована архітектура забезпечує високу гнучкість завдяки можливості незалежного розвитку і масштабування кожного компонента. Наприклад, якщо система має модуль для обробки аудіофайлів, його можна оновити або замінити, не впливаючи на модулі, які відповідають за аналіз тексту або створення звітів.

Також важливою перевагою є зменшення ризиків: збій в одному компоненті не обов'язково призводить до зупинки всієї системи. Завдяки

слабкій зв'язності можна зберігати стабільність навіть у разі проблем з однією частиною системи. Крім того, компонентно-орієнтована архітектура полегшує інтеграцію нових функцій або зовнішніх сервісів, дозволяючи адаптувати систему до змін у бізнес-вимогах або технологічному середовищі.

Попри численні переваги, компонентно-орієнтована архітектура має свої виклики. Насамперед, це складність управління залежностями та взаємодією між компонентами. Кожен компонент потребує чітко визначених інтерфейсів, що вимагає додаткового часу на проектування та тестування.

Ще одним недоліком є підвищена складність моніторингу і діагностики. У розподіленій системі потрібно контролювати роботу кожного компонента окремо, а також аналізувати, як вони взаємодіють один з одним. Це вимагає впровадження інструментів для моніторингу, журналювання та трасування запитів.

Проте компонентно-орієнтована архітектура є ідеальним рішенням для систем, які потребують гнучкості, адаптивності та легкої інтеграції з іншими сервісами. Її використання дозволяє ефективно розподіляти ресурси, забезпечувати стабільність роботи та швидко впроваджувати нові функції.

3.1. Опис реалізації серверного застосунку

Система побудована на основі компонентно-орієнтованої архітектури, яка інтегрує елементи подієво-орієнтованого підходу. Це проявляється у використанні модулів для обробки даних (аудіофайлів, текстових транскриптів) та організації асинхронних процесів, таких як взаємодія з API сторонніх сервісів (Vinotel, OpenAI), а також реалізації циклічних завдань (запланованих подій). Основний обмін даними між компонентами здійснюється через виклики методів, що дозволяє забезпечити узгодженість і незалежність частин системи.

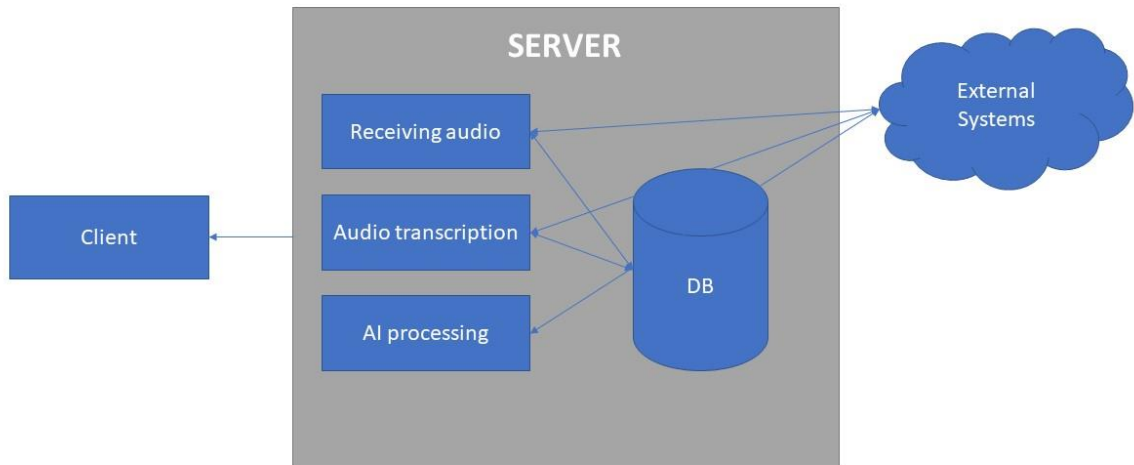


Рисунок 5. Схема моделі серверного застосунку

Вибір цієї архітектури зумовлений специфікою завдань, які вирішує система. Основна мета — обробка значного обсягу даних, зокрема аудіо- та текстових файлів, з подальшим їхнім аналізом за допомогою алгоритмів штучного інтелекту. Для таких задач важливі масштабованість, гнучкість і незалежність компонентів.

Компонентно-орієнтований підхід дозволяє легко модифікувати систему, додаючи нові функції або змінюючи існуючі без значного впливу на інші частини. Це особливо важливо для динамічних проєктів, які можуть змінюватися залежно від нових вимог.

Асинхронність і подієво-орієнтований підхід забезпечують оптимальне використання ресурсів, оскільки різні частини системи працюють незалежно, не блокуючи одна одну. Це критично для інтеграції із зовнішніми сервісами, такими як Vinotel чи OpenAI, де затримки можуть бути непередбачуваними.

Інтеграція із зовнішніми API і збереження оброблених даних у Google Drive та інших сховищах дозволяє користувачам отримувати доступ до результатів у зручному форматі. Крім того, хмарні сервіси зменшують навантаження на локальні ресурси.

Серверний застосунок функціонує як централізована система, що інтегрує різні модулі для автоматизації обробки телефонних розмов. Центральною складовою є база даних, яка забезпечує збереження та управління ключовою інформацією, включаючи списки телефонних номерів для виключення з аналізу, текстові результати транскрипції розмов та журнали подій. Використання MySQL як бази даних забезпечує ефективну організацію даних і швидкий доступ до них через SQL-запити. Наприклад, перед аналізом система перевіряє, чи номер телефону є у списку виключень. Якщо номер відсутній, його можна додати до цього списку динамічно (рис. 6).

#	ID	PhoneNumber	ActionDate	InsertDate
1	1	096...	2024-12-11 09:08:00	2024-12-11 16:18:38
2	2	093...	2024-12-09 14:03:00	2024-12-11 16:18:38
3	3	067...	2024-11-11 16:23:00	2024-12-11 16:18:38
4	4	093...	2024-12-02 10:13:00	2024-12-11 16:18:38
5	5	093...	2024-01-30 10:31:00	2024-12-11 16:18:38
6	6	096...	2024-12-10 17:46:00	2024-12-11 16:18:38
7	7	093...	2024-11-18 14:37:00	2024-12-11 16:18:38
8	8	068...	2024-12-02 17:41:00	2024-12-11 16:18:38
9	9	051...	2024-11-19 11:54:00	2024-12-11 16:18:38
10	10	093...	2024-11-22 09:07:00	2024-12-11 16:18:38
11	11	064...	2024-12-10 10:50:00	2024-12-11 16:18:38
12	12	073...	2024-12-09 17:47:00	2024-12-11 16:18:38
13	13	093...	2024-11-07 11:11:00	2024-12-11 16:18:38
14	14	093...	2024-11-06 15:51:00	2024-12-11 16:18:38
15	16	093...	2024-10-30 14:19:00	2024-12-11 16:18:38
16	17	093...	2024-11-03 10:05:00	2024-12-11 16:18:38
17	18	067...	2024-12-10 17:04:00	2024-12-11 16:18:38
18	19	093...	2024-10-25 09:22:00	2024-12-11 16:18:38
19	20	067...	2024-10-30 11:26:00	2024-12-11 16:18:38
20	22	067...	2024-12-10 16:56:00	2024-12-11 16:18:38
21	23	067...	2024-11-08 13:39:00	2024-12-11 16:18:38
22	24	068...	2024-11-04 09:17:00	2024-12-11 16:18:38

Рисунок 6. Приклад таблиці excluded_numbers у БД

Модуль взаємодії з Vinotel API відповідає за завантаження аудіозаписів дзвінків. Цей процес включає отримання даних про дзвінки, їх фільтрацію за певними критеріями, такими як тривалість або статус запису, а також завантаження відповідних файлів. Інтеграція з API Vinotel дозволяє автоматизувати збір даних і гарантує, що система працює з актуальною інформацією. На рисунку 7 зображено частину функції, яка завантажує аудіофайли до локальної файлової системи. Завантажені аудіофайли проходять багаторівневий процес обробки, починаючи з етапу їх передачі у модуль транскрипції, який організовано для інтеграції з хмарними сервісами.

Цей процес включає взаємодію з API транскриптора, за допомогою якого застосунок завантажує відповідні аудіофайли, ініціює процес автоматизованого розпізнавання мови та отримує текстовий варіант результатів розмов. Подальша обробка отриманих текстових даних передбачає їх структурування та організацію у визначені каталоги, де вони сортуються за критеріями, такими як назва проекту або дата дзвінка, забезпечуючи зручність доступу та аналізу.

Ключовою особливістю цього підходу є його ефективність у роботі з великими обсягами інформації та здатність автоматизувати тривалі ручні операції, зменшуючи тим самим потенційні помилки. Отримані транскрипти стають основою для глибшого аналізу, зокрема через залучення штучного інтелекту для визначення емоційного тону, відповідності стандартам обслуговування, а також надання рекомендацій для вдосконалення комунікаційних процесів. Завдяки застосуванню хмарних технологій, які забезпечують масштабованість, система стає гнучкою та адаптивною, що дозволяє інтегрувати її у широкий спектр бізнес-процесів. На рисунку 8 зображено приклад структури папок та файлів із аудіозаписами телефонних розмов.

Процес автоматизованої обробки аудіофайлів, інтегрований із сучасними хмарними сервісами, дозволяє підвищити ефективність аналізу комунікаційних даних та забезпечити їхнє зручне сортування.

```

143 while True:
144     logger.info(
145         f"Attempting to download audio file for project '{project_name}' with generalCallID '{call_data['generalCallID']}'")
146     result = send_request('stats/call-record.json', {
147         'generalCallID': call_data['generalCallID'],
148         'key': binotel_key,
149         'secret': binotel_secret
150     })
151
152     logger.info(f"API response: {result}")
153
154     if result['status'] == 'success':
155         call_record_content = requests.get(result['url']).content
156
157         if call_data['callType'] == '0':
158             call_record_filename = f"{call_data['generalCallID']}_ \
159                 f"{record_datetime.strftime('%d-%m-%Y_%H-%M')}_ \
160                 f"{call_data['externalNumber']}_{call_data['internalNumber']}_incoming.mp3"
161         else:
162             call_record_filename = f"{call_data['generalCallID']}_ \
163                 f"{record_datetime.strftime('%d-%m-%Y_%H-%M')}_ \
164                 f"{call_data['internalNumber']}_{call_data['externalNumber']}_outcoming.mp3"
165
166         with open(os.path.join(project_directory, call_record_filename), 'wb') as file:
167             file.write(call_record_content)
168
169         logger.info(f"Audio file saved: {call_record_filename}")
170         break
171     else:
172         logger.warning(
173             f"Failed to download audio file. REST API error {result['code']}: {result['message']}")
174         attempts += 1
175         time.sleep(5)
176
177     if attempts >= 10:
178         logger.error("Max retry attempts reached. Skipping to next call.")
179         break

```

Рисунок 7. Частина функції, яка завантажує аудіофайли

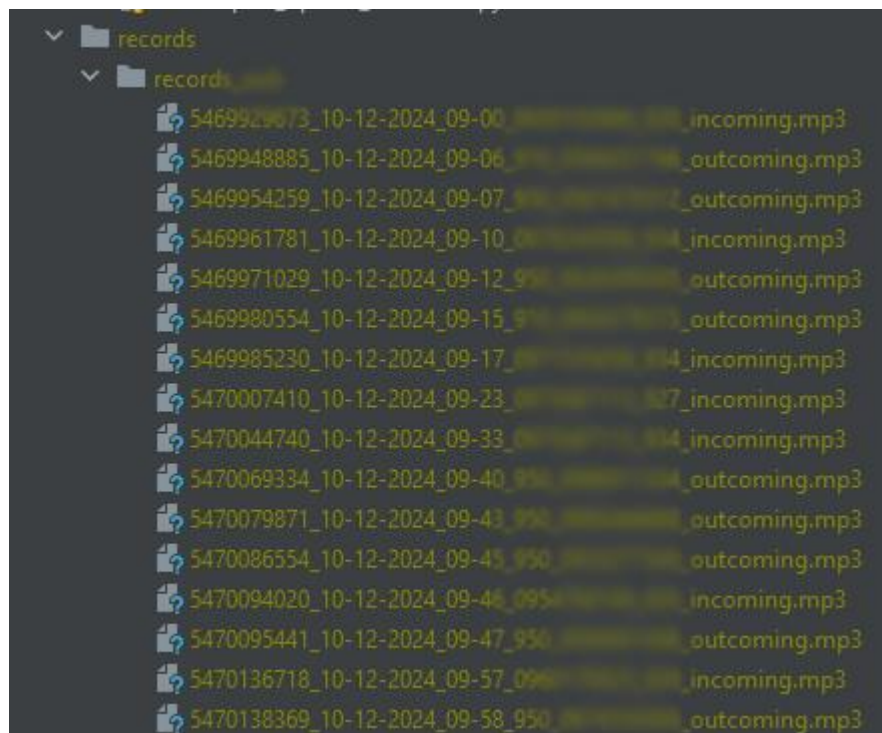


Рисунок 8. Організація збереження файлів із записами розмов

Після отримання тексту розмов вони обробляються за допомогою OpenAI. Моделі штучного інтелекту аналізують розмови, визначаючи ключові аспекти, такі як тональність, рівень ввічливості оператора, відповідність

сценаріям взаємодії з клієнтом. В результаті аналізу система формує оцінку якості обслуговування, яка включає кількісний бал і текстові коментарі для покращення роботи оператора. Ці результати автоматично додаються до Google Sheets через відповідний API, що полегшує їх подальший аналіз та звітування. Приклад такого звіту зображено на рисунку 9.

№	Дата та час розмови	Код оператора	Номер телефону	Тип дзвінка	Оцінка	Коментар від ШІ
1	18-12-2024 10-10	910	0954	Вихідний дзвінок	5	Під час роз
2	18-12-2024 11-38	910	0986	Вихідний дзвінок	6	Розмова бу
3	18-12-2024 11-02	950	0970	Вихідний дзвінок	7	З розмови і
4	18-12-2024 09-59	950	0681	Вихідний дзвінок	9	Розмова бу
5	18-12-2024 09-14	950	0982	Вихідний дзвінок	6	У ці розмо
6	18-12-2024 14-03	910	0996	Вихідний дзвінок	8	Розмова бу
7	18-12-2024 09-43	910	0968	Вихідний дзвінок	4	Оператор н
8	18-12-2024 10-00	931	0385	Вхідний дзвінок	7	Розмова бу
9	18-12-2024 10-13	910	0978	Вихідний дзвінок	9	Розмова оп
10	18-12-2024 09-07	950	0976	Вихідний дзвінок	6	Розмова не
11	18-12-2024 10-58	950	0669	Вихідний дзвінок	8	Розмова бу
12	18-12-2024 13-41	934	0412	Вхідний дзвінок	5	Розмова бу

Рисунок 9. Приклад оформлення звіту аналізу розмов від ШІ

Особливістю застосунку є його здатність автоматично запускатися щодня ввечері, виконуючи всі необхідні операції. Процес розпочинається з очищення робочих директорій, після чого завантажуються нові дзвінки, обробляються текстові дані, а результати зберігаються у вигляді структурованих звітів. Використання модульного підходу у розробці дозволяє легко адаптувати або розширювати функціональність, зберігаючи стабільність та надійність системи.

3.2. Дослідження і оцінювання критеріїв якості роботи серверного застосунку

Оцінка якості серверного застосунку для системи аналізу телефонних розмов базується на ряді ключових аспектів, що визначають його ефективність і стабільність. Серед основних критеріїв — швидкодія, надійність, оптимальне використання ресурсів, масштабованість та інтеграційна здатність. Кожен з них безпосередньо залежить від специфіки архітектури та компонентів системи.

Швидкодія серверного застосунку є основою для забезпечення його ефективності. В контексті аналізу телефонних розмов це включає час, витрачений на отримання аудіофайлів, їхнє завантаження до сервісу

транскрипції, обробку тексту мовною моделлю OpenAI та збереження результатів у базі даних або хмарному сховищі. Наприклад, завантаження файлів аудіо до сервісу транскрипції передбачає отримання URL-адреси завантаження, передачу файлу, ініціацію процесу розпізнавання та моніторинг статусу до завершення обробки. Це створює критичну точку, де час виконання може варіюватися в залежності від пропускну здатності мережі та продуктивності API зовнішнього сервісу.

Інтеграція з OpenAI також є значним чинником, що впливає на продуктивність. Взаємодія включає передачу великих текстових файлів для аналізу, отримання відповідей, включаючи оцінку якості обслуговування та генерацію коментарів. Цей процес може супроводжуватися очікуванням завершення асинхронного виконання запитів до API OpenAI, що вимагає оптимізації таймаутів і обробки помилок. У випадках затримки або відмови необхідно повторювати спроби, що впливає на загальний час виконання.

Крім того, час витрачається на очищення та підготовку середовища. Наприклад, видалення старих даних із директорій для зберігання аудіофайлів і текстових розшифровок забезпечує уникнення накопичення зайвих даних, що може уповільнити роботу файлової системи. Такі операції автоматизуються перед запуском основних процесів.

Окремим викликом є робота з базою даних. З'єднання, зберігання та перевірка даних, як-от телефонні номери або результати аналізу, вимагають мінімізації затримок через оптимізацію SQL-запитів і структури бази.

Бази даних є фундаментальною частиною будь-якої інформаційної системи. Серед сучасних СУБД (систем управління базами даних) виділяються такі, як MySQL, SQLite, MariaDB, PostgreSQL та інші. Кожна з них має унікальні характеристики, що визначають їхню придатність для певних проєктів.

MySQL є однією з найпопулярніших реляційних систем управління базами даних, яка забезпечує високу продуктивність, стабільність і гнучкість. Її розроблено як відкриту платформу з активною спільнотою, що гарантує

регулярні оновлення і широке коло доступних інструментів. Однією з ключових переваг MySQL є її оптимізація для обробки великих обсягів даних, що робить її ідеальним вибором для веб-додатків та серверних застосунків, як у цьому проєкті. MySQL також забезпечує сумісність з багатьма мовами програмування, включаючи Python, який використовується у даній системі.

SQLite, у свою чергу, є легковажною вбудованою базою даних, яка часто використовується для мобільних або невеликих додатків. Вона не вимагає окремого серверного середовища і дозволяє зберігати дані в одному файлі. Проте обмежені можливості масштабування SQLite роблять її непридатною для великих систем, які обробляють значні обсяги транзакційних даних.

MariaDB, як форк MySQL, зберігає більшість її функціональних можливостей, водночас пропонуючи додаткові покращення у сфері продуктивності та безпеки. Її відкритий код і розвиток спільнотою роблять її привабливим вибором для багатьох підприємств. Однак через специфічні налаштування та відмінності в синтаксисі вона може потребувати додаткового часу на адаптацію, що в даному випадку не є виправданим.

PostgreSQL є потужною об'єктно-реляційною СУБД, яка пропонує розширені функції роботи з даними, включаючи підтримку JSON, складних типів даних і повнотекстового пошуку. PostgreSQL також забезпечує високу надійність і відповідність стандартам ACID. Однак у порівнянні з MySQL, PostgreSQL може бути дещо складнішою у налаштуванні та управлінні, що робить її менш оптимальною для швидкого розгортання середовищ.

У контексті реалізації автоматизованої системи оцінки якості роботи підприємства було обрано MySQL як основну базу даних, оскільки вона поєднує в собі високу продуктивність, підтримку складних транзакцій і стабільність. Інтеграція MySQL з бібліотекою `mysql-connector-python` дозволяє легко виконувати SQL-запити, забезпечуючи ефективну обробку даних, таких як телефонні номери, результати транскрипції та оцінки ІШ. Крім того, MySQL ідеально підходить для організації централізованої бази даних із

можливістю масштабування в разі зростання обсягів інформації, що зберігається.

Таблиця 3.1 — Порівняння СУБД

Критерій	MySQL	SQLite	MariaDB	PostgreSQL
Тип СУБД	Реляційна	Вбудована	Реляційна	Об'єктно-реляційна
Підтримка транзакцій	Так	Обмежена	Так	Так
Масштабованість	Висока	Низька	Висока	Дуже висока
Продуктивність	Висока	Середня	Вища за MySQL	Висока
Простота налаштування	Середня	Дуже висока	Середня	Низька
Відкритий код	Так	Так	Так	Так
Сфера застосування	Веб-додатки, серверні системи	Мобільні додатки, невеликі проєкти	Альтернатива MySQL, веб-сервіси	Складні обчислення, аналітика, великі системи

У таблиці 3.1 наведено детальне порівняння раніше описаних СУБД. Таким чином, вибір MySQL базується на її здатності задовольнити поточні та майбутні потреби системи, забезпечуючи оптимальний баланс між продуктивністю, масштабованістю та простотою інтеграції.

Ефективне зберігання даних також має враховувати потреби в масштабованості, оскільки зростання обсягів даних може викликати перевантаження серверів або підвищення часу відповіді на запити. Проте це можна покращити включаючи використання індексів, зменшення обсягу даних, що обробляються, і оптимізацію запитів.

Індекси є одним із найбільш ефективних інструментів для підвищення швидкодії запитів до БД. Індекс — це структура даних, яка дозволяє БД

швидше знаходити необхідні записи, зменшуючи кількість операцій читання. Наприклад, якщо таблиця містить мільйони записів, а запит часто використовує фільтрацію за певними стовпцями (наприклад, `phone_number` або `created_date`), створення індексів для цих стовпців дозволить значно зменшити час виконання запиту. На рисунку 10 зображено налаштування індексів для поля `phone_number`.

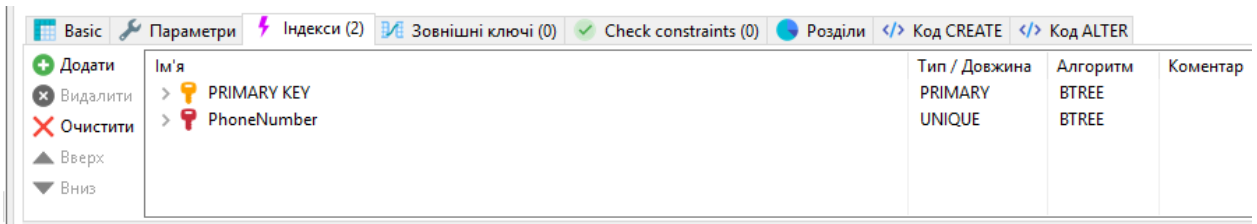


Рисунок 10. Налаштування індексів на таблиці в БД.

Особливу увагу слід приділяти композитним індексам, які створюються для кількох стовпців. Це корисно, якщо запити використовують фільтрацію за кількома умовами одночасно, наприклад, `WHERE phone_number = '12345' AND created_date > '2024-01-01'`. Проте важливо уникати надмірної кількості індексів, оскільки вони сповільнюють операції вставки та оновлення даних. Реалізація цих підходів дозволяє значно знизити час виконання запитів до БД, уникнути перевантаження сервера та забезпечити стабільну роботу застосунку навіть за умов зростання обсягів даних.

Інтеграція з хмарними сервісами, такими як Google Drive для генерації звітів, додає ще один рівень вимог. Система повинна вміти ефективно створювати ієрархію папок, перевіряти наявність файлів і, за потреби, завантажувати нові дані без надмірного використання API-запитів.

Таким чином, загальний час виконання залежить від багатьох факторів: мережових затримок, швидкості обробки зовнішніх API, продуктивності серверного обладнання та оптимізації внутрішніх алгоритмів. Для покращення роботи важливо застосовувати асинхронні методи виконання, ретельно відслідковувати та усувати вузькі місця, а також використовувати профілювання для виявлення ресурсомістких операцій. Це дозволить досягти

стабільної продуктивності навіть за умов зростання обсягів даних і складності аналізу.

3.3. Особливості розгортання серверного застосунку

Розгортання серверних застосунків може здійснюватися на кількох основних платформах, які різняться за своєю структурою, функціональністю та вартістю. Існує два основних підходи: використання хостинг-сервісів або розгортання локальних серверів. Хостинг-сервіси пропонують широкий спектр варіантів, включаючи віртуальний хостинг, VPS, хмарні рішення, виділені сервери та серверлесс архітектури. Вони забезпечують зручність налаштування, технічну підтримку, а також можливість масштабування відповідно до потреб проекту. З іншого боку, локальні сервери забезпечують повний контроль над інфраструктурою, що є особливо важливим для проектів з високими вимогами до безпеки або специфічними технічними потребами. Проте вони вимагають значних інвестицій у обладнання, підтримку і технічне обслуговування. Вибір між цими підходами залежить від специфіки завдань, обсягу трафіку, вимог до масштабованості та бюджету проекту.

Віртуальний хостинг є базовим варіантом для невеликих проектів. У межах такого сервісу один фізичний сервер обслуговує кілька веб-сайтів або застосунків, розподіляючи між ними ресурси. Хоча віртуальний хостинг є економічно вигідним і простим у налаштуванні, його обмеженість у ресурсах та недостатній контроль над середовищем робить його непридатним для завдань, які потребують обробки великих обсягів даних або використання спеціалізованих бібліотек, таких як OpenAI SDK.

VPS (віртуальні приватні сервери) є популярним вибором для середніх проектів. Такий хостинг забезпечує більшу ізольованість і можливість налаштування, оскільки кожен VPS працює як окремий віртуальний сервер із власними ресурсами. Це дозволяє встановлювати потрібні інструменти й бібліотеки для взаємодії з API, транскрипції аудіо та роботи з базами даних. VPS добре підходить для систем із середнім навантаженням і дозволяє

масштабувати ресурси відповідно до потреб проекту. Однак цей варіант потребує більше технічних знань для налаштування та обслуговування.

Хмарні рішення, такі як AWS, Google Cloud або Microsoft Azure, є найбільш гнучкими та потужними платформами. Вони забезпечують високий рівень масштабованості, продуктивності та інтеграції з іншими сервісами. Наприклад, використання хмарної інфраструктури дозволяє автоматизувати створення віртуальних серверів, налаштовувати балансувальники навантаження та інтегрувати сервери з хмарними базами даних. Крім того, хмарні платформи пропонують спеціалізовані рішення для обробки даних, наприклад, хмарні функції для обробки тексту чи аудіо. Це ідеальний вибір для масштабних проектів, які потребують високої продуктивності та інтеграції з іншими інструментами. Головним недоліком є вищі витрати, особливо за умов тривалого використання ресурсів.

Виділені сервери надають повний контроль над апаратним забезпеченням і ресурсами. Вони підходять для проектів, які потребують стабільного високого навантаження та використання специфічного програмного забезпечення. Однак цей варіант є дорогим і менш гнучким у масштабуванні порівняно з хмарними платформами. Також він вимагає значних технічних знань для підтримки.

Серверлесс архітектура є сучасним підходом, який дозволяє уникнути управління інфраструктурою. У такому випадку розробники зосереджуються лише на коді, тоді як платформа, наприклад AWS Lambda, автоматично масштабує ресурси відповідно до навантаження. Серверлесс рішення є ідеальними для подійно-орієнтованих систем, які працюють нерегулярно або мають пікові навантаження. Недоліком є обмеження в тривалості виконання функцій і залежність від платформи.

Розглянемо порівняння українського та іноземного продуктів хостингу. Ukraine.com.ua та DigitalOcean є представниками різних підходів до надання хостинг-послуг. Ukraine.com.ua орієнтований переважно на український ринок, пропонуючи локальні сервери, що забезпечують мінімальну затримку

для користувачів із цього регіону. Окрім цього, провайдер надає технічну підтримку українською мовою, що є важливим для підприємств, які працюють із локальною аудиторією. Вартість послуг Ukraine.com.ua є конкурентною, особливо для малого та середнього бізнесу, що прагне скоротити витрати, зберігаючи при цьому якість обслуговування. Основна спеціалізація включає традиційний веб-хостинг, VPS/VDS із SSD-дисками та виділені сервери, що адаптовані до потреб середньої складності проектів.

З іншого боку, DigitalOcean є глобальним провайдером хмарних рішень, орієнтованим на розробників та компанії, що працюють із масштабованими проектами. Ця платформа пропонує хмарні сервери, які можна налаштувати відповідно до індивідуальних потреб, зокрема через використання інтерфейсу програмування додатків (API). DigitalOcean забезпечує високу продуктивність і гнучкість завдяки своїй інфраструктурі, що включає центри обробки даних у багатьох країнах світу. Це дозволяє компаніям із глобальною аудиторією розміщувати сервери ближче до своїх клієнтів, мінімізуючи затримки. Хмарні бази даних, об'єктне зберігання та аналітичні інструменти доповнюють основний набір послуг DigitalOcean, забезпечуючи підтримку як для стартапів, так і для великих організацій.

Обидві платформи мають відмінності в технічному обслуговуванні. Ukraine.com.ua пропонує стандартну технічну підтримку, орієнтовану на вирішення типових завдань користувачів. Натомість DigitalOcean надає доступ до широкого спектра технічної документації, навчальних матеріалів та англомовної підтримки, яка більше орієнтована на досвідчених розробників.

Цінова політика також варіюється залежно від масштабу проекту. Ukraine.com.ua забезпечує вигідні умови для проектів із помірним навантаженням і пропонує фіксовані тарифи, що спрощує прогнозування витрат. У свою чергу, DigitalOcean працює за моделлю оплат за використання ресурсів, що є зручним для проектів із динамічним навантаженням, але може стати дорожчим за тривалого використання великих обсягів ресурсів.

Таблиця 3.2 — Порівняння Ukraine.com.ua та DigitalOcean

Характеристика	Ukraine.com.ua	DigitalOcean
Локалізація серверів	Україна	Глобально
Основна аудиторія	Локальні проекти, малий і середній бізнес	Глобальні проекти, стартапи, корпорації
Типи послуг	Веб-хостинг, VPS/VDS, виділені сервери	VPS, хмарні бази даних, об'єктне зберігання
Підтримка	Українська, локальна	Англійська, глобальна
Гнучкість масштабування	Обмежена	Висока
Модель оплати	Фіксовані тарифи	Оплата за використання
Технічна підтримка	Стандартна	Оплата за використання
Інтеграція з ШІ	Мінімальна	Широкі можливості

У таблиці 3.2 наведена порівняльна характеристика цих двох продуктів. Таким чином, вибір між Ukraine.com.ua та DigitalOcean залежить від потреб проекту. Ukraine.com.ua краще підходить для локальних бізнесів і проектів із чітко визначеним навантаженням, тоді як DigitalOcean є оптимальним вибором для масштабованих глобальних систем та технологічно складних проектів.

Для розгортання серверного застосунку обрано операційну систему Ubuntu Server 20.04, яка зарекомендувала себе як одна з найбільш надійних, продуктивних та гнучких систем серед UNIX-подібних платформ. Ubuntu Server забезпечує оптимальні умови для роботи серверного програмного забезпечення завдяки своїй стабільності, активній підтримці спільноти та розширеному функціоналу для управління сучасними інфраструктурами.

Ubuntu Server 20.04 має кілька суттєвих переваг перед іншими UNIX-подібними системами. Насамперед, вона є відкритим програмним забезпеченням із широкою спільнотою, що дозволяє користувачам отримувати своєчасну підтримку, в тому числі через форуми, документацію та безкоштовні ресурси. Це є значною перевагою перед комерційними UNIX-

системами, такими як AIX чи Solaris, які потребують значних ліцензійних витрат і мають обмежену спільноту користувачів.

Ще однією важливою перевагою Ubuntu Server є її довготривала підтримка (Long Term Support, LTS), яка забезпечує регулярні оновлення безпеки та стабільності протягом п'яти років після випуску. Це робить її ідеальною для розгортання серверних застосунків у виробничому середовищі, адже безпека та стабільність є критично важливими аспектами для таких систем.

Ubuntu також підтримує сучасні технології контейнеризації та оркестрації, зокрема Docker і Kubernetes, які дозволяють легко масштабувати застосунки та інтегрувати їх у хмарні інфраструктури. Крім того, система підтримує широкий спектр інструментів для автоматизації, таких як Ansible, Puppet і Chef, що спрощує управління серверами та їх конфігурацію. У контексті проєкту це забезпечує швидке налаштування середовища для запуску ключових компонентів застосунку, таких як обробка аудіозаписів, взаємодія з API сторонніх сервісів та інтеграція з мовними моделями OpenAI.

Порівняно з іншими популярними системами, такими як CentOS або Debian, Ubuntu Server має більш сучасну систему управління пакетами (apt), яка забезпечує швидкий доступ до актуальних версій програмного забезпечення. Це особливо важливо для підтримки таких бібліотек, як OpenAI SDK або інструментів для роботи з базами даних, що використовуються у поточному проєкті.

Отже, використання Ubuntu Server 20.04 є обґрунтованим вибором для розгортання серверного застосунку, оскільки вона поєднує в собі надійність, гнучкість, високу продуктивність та підтримку сучасних технологій, необхідних для стабільної роботи та розвитку програмних систем.

Для реалізації серверного застосунку використовувалася мова програмування Python 3.10, яка завдяки своїй сучасній архітектурі та багатому набору функціональних можливостей забезпечує ефективну розробку та інтеграцію серверних рішень. У цьому проєкті було залучено кілька

спеціалізованих бібліотек, що відповідають за різні аспекти функціональності застосунку.

Для взаємодії із зовнішніми сервісами використовувалася бібліотека `requests` (версія 2.32.3), яка надає простий і ефективний інтерфейс для роботи з HTTP-запитами. Це дозволяє завантажувати аудіофайли, ініціювати транскрибування та отримувати результати обробки даних від сторонніх сервісів. Аналіз транскрибованих текстів та оцінка якості роботи операторів здійснювалися за допомогою бібліотеки `openai` (версія 1.57.1), яка є офіційним SDK для інтеграції з мовними моделями OpenAI.

Для зберігання та використання конфіденційних даних, таких як API-ключі, паролі та змінні середовища, було обрано `python-dotenv` версії 1.0.1. Це дозволило забезпечити високий рівень безпеки, адже чутливі дані зберігаються окремо від основного коду. Водночас, для управління та обробки електронних таблиць Google Sheets застосовувалася бібліотека `gsread` (версія 6.1.4), яка інтегрується з сервісами Google через OAuth2. Вона спрощує створення та модифікацію таблиць, необхідних для генерації звітів.

Інтеграція з хмарними сервісами Google також забезпечувалася бібліотекою `oauth2client` (версія 4.1.3), яка дозволяє автентифікуватися через облікові записи Google і здійснювати доступ до файлів на Google Drive. Бібліотека `pydrive` (версія 1.3.1) розширювала ці можливості, забезпечуючи зручний доступ до керування файлами в Google Drive.

Для роботи з великими обсягами даних і здійснення аналітичних операцій застосовувалася бібліотека `pandas` (версія 2.2.3), яка надає потужні інструменти для обробки, маніпуляції та аналізу даних. Крім того, для інтеграції з API сервісів Google була залучена бібліотека `google-api-python-client` (версія 2.153.0), яка дозволяє використовувати функції хмарних сервісів Google, такі як створення, редагування та пошук файлів.

Для автоматизації задач і організації їх регулярного виконання була обрана бібліотека `schedule` (версія 1.2.2), що забезпечила зручний спосіб запуску функцій у заданий час. Інтеграція з базою даних MySQL

здійснювалася за допомогою `mysql-connector-python` (версія 9.1.0), що забезпечило ефективне підключення, виконання запитів і взаємодію з базою даних.

Для забезпечення автоматичного запуску застосунку щоденно була реалізована система розкладу за допомогою `cron`. Використання цього підходу є ефективним та надійним методом автоматизації завдань на сервері, оскільки `cron` спеціалізується на виконанні запланованих процесів в обраний час.

Процес налаштування включає кілька ключових кроків. Спершу, у файлі конфігурації `crontab` був налаштований розклад виконання завдання. У цьому випадку застосунок запускається кожного дня о 21:00, що відповідає команді `0 21 * * *`. Часовий пояс було задано через змінну `TZ='Europe/Kyiv'`, щоб забезпечити правильне виконання задачі з урахуванням локального часу. Окрім цього, було налаштовано перенаправлення стандартного виводу в `dev/null` і помилок у лог-файл, що дозволяє адміністратору легко контролювати можливі збої.

Скрипт, що виконується через `cron`, також був оптимізований для безпечного запуску. Він перевіряє наявність іншого екземпляру процесу за допомогою механізму блокування через файл (`LOCKFILE`). У разі виявлення запущеного процесу, попередній екземпляр завершується, і лише після цього скрипт створює новий блокувальний файл. Завдяки цьому уникається паралельне виконання кількох копій застосунку, що може призвести до конфліктів. Далі відбувається активація віртуального середовища Python та запуск основного скрипту застосунку. Це зображено на рисунку 11.

```

GNU nano 4.8
#!/bin/bash
LOCKFILE=/tmp/ai.lock

# Перевірка, чи вже запущено
if [ -e $LOCKFILE ]; then
    OLD_PID=$(cat $LOCKFILE)
    if kill -0 $OLD_PID 2>/dev/null; then
        echo "Previous instance (PID: $OLD_PID) is running. Terminating it."
        kill -9 $OLD_PID
    fi
fi

# Створення нового блокування
echo $$ > $LOCKFILE
trap "rm -f $LOCKFILE" EXIT

# Виконання Python скрипта
source /home/ai/venv/bin/activate
python /home/ai/app.py

```

[Read 19 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos	M-U Undo
^X Exit	^R Read File	^_ Replace	^U Paste Text	^T To Spell	^_ Go To Line	M-E Redo

Рисунок 11. Скрипт, який запускає застосунок

Використання cron замість бібліотек Python, таких як schedule, є обґрунтованим з кількох причин. По-перше, cron є частиною системного середовища, що забезпечує надійність виконання завдань навіть після перезавантаження системи, тоді як бібліотека Python потребує постійного виконання основного процесу програми. Це критично важливо для серверних рішень, де надійність і безперерійність є основними вимогами. По-друге, cron дозволяє адміністратору централізовано управляти всіма запланованими завданнями через єдиний файл конфігурації. Це спрощує моніторинг і підтримку.

Висновки

Штучний інтелект (ШІ) здійснив значний прорив у розвитку від теоретичних концепцій до прикладних рішень, які сьогодні інтегруються в різні галузі людської діяльності. Завдяки еволюції комп'ютерної техніки, математичних методів та алгоритмів машинного навчання, ШІ став ключовим інструментом для обробки великих обсягів даних, аналізу інформації та прийняття рішень із високою точністю. Ця магістерська робота демонструє один із аспектів застосування ШІ — автоматизацію оцінки якості обслуговування в закладах громадського харчування через аналіз комунікації клієнтів із операторами кол-центру.

Особливий акцент у роботі було зроблено на використанні мовних моделей, які здатні аналізувати текстові дані, розуміти тональність та зміст розмов. Використання таких моделей у поєднанні з методами обробки природної мови забезпечило можливість автоматизованого аналізу телефонних розмов за чітко визначеними критеріями. Інтеграція цих підходів дозволила створити систему, що оцінює роботу операторів за такими параметрами, як емоційна ввічливість, дотримання стандартів спілкування, швидкість реакції на запити та загальне задоволення клієнтів.

Під час реалізації проекту було розроблено функціональний прототип системи, що складається з кількох модулів, включаючи завантаження аудіофайлів із зовнішніх джерел, їх обробку через транскрипційні сервіси, аналіз отриманих текстів за допомогою API мовних моделей OpenAI, а також формування звітів для управлінського персоналу. Важливим аспектом стало врахування специфіки мовних та емоційних характеристик розмов, що підвищило точність оцінок.

Кодова база проекту продемонструвала використання сучасних технологій, таких як Python, бібліотек для інтеграції з API, інструментів для роботи з Google Drive та Google Sheets, а також бази даних для управління даними. Автоматизація ключових процесів, таких як завантаження записів,

розпізнавання тексту та формування аналітичних звітів, дозволила створити систему, що відповідає сучасним вимогам до подібних рішень.

Результати роботи підтвердили ефективність застосування ШІ для вирішення задач у сфері обслуговування, продемонструвавши значний потенціал для підвищення якості обслуговування клієнтів та оптимізації роботи персоналу. Створена система може бути адаптована для використання в інших галузях, де необхідний аналіз комунікацій та контроль якості. Таким чином, магістерська робота робить внесок у розвиток сучасних рішень на базі ШІ, підкреслюючи його універсальність і адаптивність для вирішення прикладних задач.

Література

1. Ворочек О. Г. Використання мовних моделей штучного інтелекту для генерації публікацій у соціальних мережах / О. Г. Ворочек, І. В. Соловей. // Харківський національний університет радіоелектроніки. – 2024. – С. 128–134.
2. Омельченко С. О. Використання штучного інтелекту в медицині / С. О. Омельченко, Н. Г. Аксак. // Харківський національний університет радіоелектроніки. – 2022. – С. 36–37.
3. Ramesh A. N. Artificial intelligence in medicine / Ramesh. // Annals of the Royal College of Surgeons of England. – 2004. – №5. – С. 334.
4. Радутний О. Е. Юридична освіта та сфера надання правових послуг в контексті штучного інтелекту / О. Е. Радутний // Інформація і право. – 2019. – С. 40–54.
5. KPMG Global [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://assets.kpmg.com/content/dam/kpmg/xx/pdf/2022/09/future-of-commercial-banking-web.pdf>.
6. Праздніков В. О. Моделі та методи машинного навчання для розпізнавання фейкового контенту / В. О. Праздніков, І. І. Сугоняк. // Державний університет «Житомирська політехніка». – 2023. – С. 131–136.
7. Malhotra T. Top Large Language Models (LLMs): A Comprehensive Ranking of AI Giants Across 13 Metrics Including Multitask Reasoning, Coding, Math, Latency, Zero-Shot and Few-Shot Learning, and Many More [Електронний ресурс] / Tanya Malhotra. – 2024. – Режим доступу до ресурсу: <https://www.marktechpost.com/2024/09/08/top-large-language-models-llms-a-comprehensive-ranking-of-ai-giants-across-13-metrics-including-multitask-reasoning-coding-math-latency-zero-shot-and-few-shot-learning-and-many-more/>.
8. Pichai S. Introducing Gemini: our largest and most capable AI model [Електронний ресурс] / S. Pichai, D. Hassabis. – 2023. – Режим доступу до ресурсу: <https://blog.google/technology/ai/google-gemini-ai/#sundar-note>.
9. World Economic Forum (2021). This alliance aims to accelerate the adoption of inclusive, trusted and transparent AI worldwide. [Електронний ресурс]. – 2021.

– Режим доступу до ресурсу: <https://www.weforum.org/agenda/2021/01/global-ai-action-alliance/>.

10. Круковська О. В. Інноваційні тенденції у логістиці: від автоматизації до штучного інтелекту / О. В. Круковська, О. Б. Кондрат, Н. М. Стрельченко. // Актуальні питання у сучасній науці. – 2024. – №6. – С. 94–105.

11. Мельник О. С. Використання штучного інтелекту для кібербезпеки / Олексій Сергійович Мельник. // Наука і техніка сьогодні. – 2024. – №7. – С. 871–882.

12. З штучним інтелектом: BMW навчилась контролювати виробництво новим методом [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://autogeek.com.ua/z-shtuchnym-intelektom-bmw-navchylas-kontrolyuvaty-vyrobnytvo-novym-metodom/>.

13. Мироненко О. Роль штучного інтелекту та машинного навчання в оптимізації логістичних процесів [Електронний ресурс] / О. Мироненко. – 2024. – Режим доступу до ресурсу: <https://cargofy.ua/uk/blog/rol-shtuchnogo-intelektu-ta-mashinnogo-navchannya-v-optimizaciji-logistichnih-procesiv>.

14. Wolfe A. McDonald's Tests AI-Powered Digital Menu Boards [Електронний ресурс] / Wolfe. – 2019. – Режим доступу до ресурсу: <https://hospitalitytech.com/mcdonalds-tests-ai-powered-digital-menu-boards>.

15. Domino's and Microsoft Cook Up AI-Driven Innovation Alliance for Smarter Pizza Orders and Seamless Operations [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.prnewswire.com/news-releases/dominos-and-microsoft-cook-up-ai-driven-innovation-alliance-for-smarter-pizza-orders-and-seamless-operations-301945321.html>.