НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК

«ПОГОДЖЕНО»	**«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»**
Декан факультету інформаційних технологій	Завідувач кафедри комп'ютерних наук
Болбот І.М., д.т.н., професор	Голуб Б.Л., к.т.н., доцент
_____	_____
____ _____ 2024 р.	**____ _____ 2024 р.**

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему ____«Система управління інформацією в месенджері»_____

_____

Спеціальність __122 – «Комп'ютерні науки»_____

_____
(код і назва)

Освітня програма _«Комп'ютерний еколого-економічний моніторинг»__

_____
(назва)

Орієнтація освітньої програми _____
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

_____ _____ _____
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи

___ст. викладач_____ _____ _Панкратьєв Віктор Олександрович_
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ __Нужняк Віталій Анатолійович__
(підпис) (ПІБ студента)

КИЇВ-2024

Зміст

Вступ

Актуальність дослідження. У сучасному світі месенджери, відіграють важливу роль у швидкому обміні інформацією, що робить їх ключовими інструментами для особистого та корпоративного спілкування. Зі зростанням обсягів даних, які обробляються цими платформами, зростає необхідність автоматизації процесів збирання, аналізу та управління інформацією. Класичні методи обробки повідомлень обмежені у своїй здатності швидко й ефективно аналізувати великі обсяги даних та виявляти відхилення в поведінці користувачів. Використання нейронних мереж, таких як GPT або BERT, дозволяє автоматизувати аналіз тексту, забезпечити виявлення аномалій і підвищити безпеку в реальному часі, що є надзвичайно актуальним для сучасних месенджерів.

Нейронні мережі відкривають можливості для глибокого аналізу текстових повідомлень, автоматичної класифікації за темами, виявлення аномалій та інтеграції з аналітичними інструментами, такими як Power BI, для візуалізації даних. Це дослідження є актуальним, оскільки розробка інтелектуальної системи управління інформацією, що поєднує нейронні мережі та аналітичні платформи, має великий потенціал для підвищення ефективності роботи месенджерів та інформаційної безпеки.

Об’єкт дослідження. Об'єктом дослідження є процес управління інформацією у месенджерах, що включає збирання, обробку та аналіз текстових повідомлень.

Предмет дослідження. Предметом дослідження є застосування нейронних мереж для аналізу повідомлень у месенджерах, зокрема для автоматичної класифікації, виявлення аномалій у текстах повідомлень та інтеграції з інструментами візуалізації даних.

Мета дослідження. Метою дослідження є розробка та впровадження системи на базі нейронних мереж для аналізу повідомлень у месенджерах, що

дозволить покращити управління інформацією шляхом автоматичної класифікації, виявлення аномалій у текстах повідомлень та створення візуальних аналітичних звітів.

Завдання дослідження

1. Провести аналіз сучасних підходів до управління інформацією в месенджерах із використанням нейронних мереж.
2. Розробити модель системи з використанням Telegram-бота, що збиратиме повідомлення та передаватиме їх на обробку нейронній мережі.
3. Побудувати архітектуру системи для класифікації повідомлень та виявлення аномалій у текстах.
4. Впровадити алгоритми класифікації та аналізу активності на основі моделей GPT або BERT.
5. Інтегрувати систему з Power BI для створення інфографіки на основі результатів аналізу повідомлень.
6. Провести тестування системи на реальних даних та оцінити її ефективність.
7. Надати рекомендації щодо впровадження системи в реальні месенджери для покращення управління інформацією.

Методи дослідження. Для виконання дослідження використовувались методи системного аналізу, моделювання та нейронні мережі для класифікації повідомлень і виявлення аномалій. Основні моделі включають GPT та BERT, які застосовувались для обробки текстових даних. Інтеграція з Power BI здійснювалася для візуалізації даних та формування аналітичних звітів.

Наукова новизна. Наукова новизна роботи полягає у створенні моделі, що поєднує можливості нейронних мереж для аналізу текстових повідомлень та аналітичних платформ для візуалізації результатів. Вперше запропоновано інтегрований підхід, який використовує нейронні мережі для виявлення аномалій у повідомленнях і автоматичної класифікації з можливістю візуалізації даних.

Апробація результатів дослідження. Основні результати дослідження були представлені на конференціях з інформаційних технологій та опубліковані у наукових виданнях. Окремі аспекти дослідження використовувались у практичних проектах з розробки інформаційних систем для обробки повідомлень.

Структура магістерської роботи. Магістерська робота складається зі вступу, чотирьох розділів основної частини, висновків, списку використаних джерел та додатків. У першому розділі представлено системний аналіз предметної області. У другому розділі подано моделювання системи. Третій розділ присвячений розробці архітектури системи, а четвертий — результатам дослідження.

1 Системний аналіз предметної області

1.1. Опис процесів управління інформацією у месенджерах.

Управління інформацією у месенджерах охоплює широкий спектр процесів, спрямованих на обробку, зберігання та аналіз повідомлень, а також забезпечення ефективного та безпечного обміну даними між користувачами. Основні функції месенджерів, такі як обмін текстовими повідомленнями, аудіо, відео та файлами, відомі широкому загалу, однак за лаштунками цих процесів стоять складні системи обробки даних. Управління інформацією у месенджерах включає кілька ключових процесів, кожен з яких виконує важливу роль у забезпеченні безперебійної роботи месенджера та задоволення потреб користувачів.

1.1.1. Передавання повідомлень. Передавання повідомлень є основним процесом у месенджерах. Він включає обробку різних типів повідомлень, таких як текстові повідомлення, зображення, аудіо- та відеофайли, а також документи. Сучасні месенджери повинні підтримувати швидкий і надійний обмін цими повідомленнями навіть у мережах зі слабким сигналом або високим рівнем затримок.

Процес передавання повідомлень передбачає їх шифрування для забезпечення конфіденційності даних, обробку на сервері для маршрутизації та збереження повідомлень, а також доставку їх кінцевому користувачеві. Сучасні месенджери використовують різноманітні протоколи для оптимізації передавання даних, серед яких найпопулярнішими є протоколи TCP/IP та UDP для швидкої і надійної доставки. Більшість сучасних платформ забезпечують шифрування повідомлень від кінця до кінця (end-to-end encryption), що гарантує, що навіть сервери месенджера не мають доступу до змісту повідомлень користувачів.

1.1.2. Зберігання повідомлень. Інший важливий аспект управління інформацією у месенджерах – це зберігання повідомлень. Більшість сучасних месенджерів зберігають повідомлення на своїх серверах або на пристроях користувачів, забезпечуючи доступ до історії чатів у будь-який момент. Це дозволяє користувачам повертатися до старих повідомлень, файлів або зображень, що були надіслані раніше.

Зберігання даних також вимагає застосування рішень для оптимізації ресурсів серверів, таких як технології хмарних сховищ і баз даних з розподіленою архітектурою. Для забезпечення безперебійного доступу до історії повідомлень, месенджери використовують індексацію та швидкий пошук даних, що дозволяє користувачам швидко знаходити потрібні повідомлення або файли.

1.1.3. Безпека даних. Оскільки месенджери працюють з величезними обсягами персональних даних, важливо забезпечити належний рівень безпеки. Головною проблемою є захист даних від сторонніх атак та витоків інформації. Більшість сучасних месенджерів використовують протоколи шифрування для захисту повідомлень, зокрема протокол Signal або власні розроблені рішення для шифрування.

Також важливим є захист від шахрайських дій, спаму та шкідливого програмного забезпечення. Месенджери використовують різноманітні алгоритми для виявлення та фільтрації таких загроз, що дозволяє зберігати чистоту інформаційного простору та захищати користувачів від можливих атак.

1.1.4. Організація та пошук інформації. Зі збільшенням кількості повідомлень, які щоденно передаються через месенджери, постає питання ефективної організації інформації. Важливо забезпечити можливість швидкого пошуку повідомлень або файлів у великих чатах або групах. Більшість сучасних месенджерів забезпечують користувачам функцію пошуку за ключовими словами, датами або типами файлів.

Для цього використовується індексація повідомлень та спеціальні алгоритми пошуку, які дозволяють миттєво знайти потрібну інформацію серед

великої кількості даних. Деякі месенджери також використовують машинне навчання для покращення результатів пошуку та організації контенту.

1.1.5. Аналітика та моніторинг активності. Месенджери також надають можливості для аналітики та моніторингу активності користувачів. Це включає аналіз часу активності, частоти відправлення повідомлень, кількості переданих файлів та взаємодій у чатах. Такі дані використовуються для надання рекомендацій, прогнозування активності користувачів, виявлення пікових періодів використання системи або навіть виявлення підозрілих дій.

Аналітичні дані можуть бути корисними як для самих користувачів, так і для адміністрації месенджера. Адміністрація може використовувати ці дані для оптимізації роботи системи, покращення якості користувацького досвіду або з комерційною метою.

1.1.6. Інтеграція з іншими системами. Сучасні месенджери не функціонують як ізольовані системи, а часто інтегруються з іншими сервісами, такими як календарі, системи управління завданнями, корпоративні платформи як-от CRM тощо. Це дозволяє користувачам планувати зустрічі, керувати проектами або здійснювати інші дії безпосередньо з месенджера, підвищуючи таким чином зручність та ефективність роботи.

Процес інтеграції включає обмін даними через API та використання різних протоколів для забезпечення синхронізації інформації між різними системами.

1.2. Аналіз існуючих рішень для управління інформацією у месенджерах.

Сучасні месенджери активно впроваджують інноваційні підходи для ефективного управління інформацією, адаптуючись до зростаючих вимог безпеки, конфіденційності та швидкості обробки даних. Ці рішення охоплюють різноманітні аспекти, включаючи оптимізацію роботи з великими обсягами даних, обробку різних типів контенту (тексту, мультимедіа), а також вдосконалення методів фільтрації та аналізу інформації. За останні роки було

розроблено низку категорій рішень, які допомагають месенджерам вирішувати такі виклики, як забезпечення стабільності роботи під час високих навантажень, інтеграція з іншими сервісами, захист від спаму та шахрайства, а також аналіз поведінки користувачів для виявлення аномалій.

1.2.1. Класичні системи управління інформацією. На початкових етапах розвитку месенджерів використовувалися відносно прості системи управління інформацією, які базувалися на традиційних методах зберігання та передавання повідомлень. Такі системи здебільшого фокусувалися на забезпеченні базової функціональності: передавання тексту, зображень та файлів між користувачами, підтримуючи простий пошук повідомлень за ключовими словами.

Основні характеристики класичних систем:

- Використання реляційних баз даних для зберігання історії повідомлень та метаданих.
- Пошук на основі ключових слів та індексації повідомлень.
- Просте шифрування (як правило, для захисту передавання даних).

Попри свою простоту, такі системи мали низку обмежень. Вони не могли ефективно обробляти великі обсяги інформації, зокрема, мультимедійний контент (аудіо, відео) або повідомлення, що вимагали складного аналізу (виявлення аномалій або спаму).

1.2.2. Фільтрація спаму та модерація контенту. З появою нових загроз, таких як спам, шкідливі програми та шахрайські дії, у месенджерах почали використовуватися спеціалізовані системи для виявлення небажаного контенту та забезпечення безпеки користувачів. На ранніх етапах ці системи використовували прості фільтри на основі чорних списків, шаблонів і ключових слів.

Основні методи класичної фільтрації спаму:

- Чорні списки IP-адрес та номерів телефонів: спамери часто використовують ті самі IP-адреси або номери для розсилки небажаних повідомлень. Додавання цих джерел до чорного списку дозволяє заблокувати повідомлення до того, як вони досягнуть користувача.

- Фільтри за ключовими словами: використання словникових фільтрів для виявлення та блокування повідомлень, які містять характерні фрази або ключові слова, що зазвичай використовуються спамерами або шахраями.
- Шаблони поведінки: алгоритми можуть аналізувати частоту та обсяг повідомлень від конкретних користувачів, що дозволяє виявити підозрілу активність.

Недоліки цього підходу полягають у тому, що він є неефективним проти нових або адаптивних видів спаму, оскільки спамери постійно змінюють свої методи, використовуючи нові словникові варіанти або проксі-сервіси для приховування своїх IP-адрес.

1.2.3. Використання машинного навчання для класифікації та аналізу повідомлень. Останніми роками багато месенджерів почали використовувати алгоритми машинного навчання для автоматизації аналізу повідомлень, їх класифікації та виявлення загроз. Машинне навчання дозволяє створювати системи, які не тільки реагують на вже відомі шаблони загроз, але й здатні адаптуватися до нових, раніше невідомих видів атак або незвичних поведінкових шаблонів.

Основні переваги систем на основі машинного навчання:

- Адаптивність: алгоритми можуть навчатися на нових даних і підлаштовуватися під зміни в поведінці користувачів або появу нових загроз.
- Класифікація тексту: алгоритми, такі як LSTM (Long Short-Term Memory) та CNN (Convolutional Neural Networks), використовуються для аналізу текстових повідомлень, їх класифікації за темами або емоціями, а також для виявлення спаму або ненормативного контенту.
- Аналіз поведінки користувачів: машинне навчання дозволяє створювати моделі, які можуть виявляти незвичну активність користувачів, що може свідчити про спроби шахрайства або несанкціонованого доступу.

1.2.4. Глибоке навчання для аналізу мультимедійного контенту. Окрім текстових повідомлень, значну частину даних у месенджерах складають мультимедійні файли, такі як зображення, відео та аудіо. Традиційні методи обробки таких даних є складними та вимагають значних ресурсів. Однак, з появою методів глибокого навчання, стало можливим значно покращити аналіз цього типу контенту.

Глибоке навчання дозволяє автоматично:

- Розпізнавати зображення та відео: алгоритми на основі CNN дозволяють аналізувати вміст зображень або відео для виявлення конкретних об'єктів або сценаріїв (виявлення насильства, неприйнятного контенту).
- Аналізувати аудіо: нейронні мережі можуть розпізнавати аудіозаписи, виділяти мовлення з фонових звуків та аналізувати його для подальшої класифікації або виявлення небажаного контенту.

1.2.5. Інтеграція з CRM та іншими бізнес-системами. Месенджери все більше використовуються не лише для особистого спілкування, але й як інструмент для бізнесу. Компанії інтегрують месенджери з CRM-системами (Customer Relationship Management) та іншими бізнес-системами для управління комунікаціями з клієнтами та обробки замовлень.

- Інтеграція з CRM: месенджери дозволяють автоматично фіксувати взаємодію з клієнтами, зберігати історію листування та інтегрувати цю інформацію у CRM для покращення взаємодії з клієнтами.
- Інтеграція з ERP (Enterprise Resource Planning): деякі компанії інтегрують месенджери з ERP-системами для керування внутрішніми процесами, такими як логістика, постачання, бухгалтерія та HR.
- Автоматизація завдань та управління проектами: месенджери також підтримують інтеграцію з платформами для управління проектами (Trello, Asana), що дозволяє автоматизувати управління завданнями безпосередньо у месенджері.

1.2.6. Хмарні рішення для зберігання та обробки даних. Збільшення обсягу інформації, що генерується у месенджерах, призвело до активного використання хмарних технологій для зберігання та обробки даних. Хмарні рішення дозволяють месенджерам масштабувати свої ресурси та забезпечувати високу доступність послуг.

Переваги хмарних рішень:

- Масштабованість: хмарні сервіси дозволяють швидко збільшувати або зменшувати ресурси залежно від навантаження на систему.
- Надійність: хмарні платформи забезпечують резервне копіювання та швидке відновлення даних у разі збою або втрати інформації.
- Висока доступність: хмарні рішення дозволяють зберігати та обробляти дані у різних дата-центрах по всьому світу, забезпечуючи високу швидкість доступу до даних.

Таким чином, сучасні рішення для управління інформацією у месенджерах стають дедалі складнішими та використовують передові технології для забезпечення безпеки, ефективності та надійності систем. Проте, попри значний прогрес, залишаються виклики, що потребують подальших досліджень та вдосконалення.

1.3. Проблеми, які потребують вирішення.

Незважаючи на значний прогрес у розробці та впровадженні рішень для управління інформацією в месенджерах, існує кілька важливих проблем, які залишаються актуальними. Ці проблеми пов’язані зі зростаючим обсягом даних, підвищеними вимогами до безпеки, складністю обробки мультимедійного контенту, а також потребою у більш ефективних та адаптивних алгоритмах для виявлення аномалій, класифікації та аналізу повідомлень. Розглянемо основні проблеми, які потребують вирішення у межах цієї галузі.

1.3.1. Обробка великих обсягів даних. Щодня користувачі месенджерів генерують величезну кількість повідомлень, файлів, зображень та інших даних.

Це створює проблему ефективного оброблення, зберігання та організації таких обсягів інформації. Зберігання даних у розподілених системах або у хмарних сховищах допомагає вирішити частину цього завдання, але основна проблема залишається у швидкості та масштабованості обробки даних.

Обробка великих обсягів інформації потребує від месенджерів наступного:

- Масштабованості: здатність системи масштабуватися для обробки дедалі більших обсягів даних без втрати продуктивності або зниження якості обслуговування користувачів.
- Швидкодії: забезпечення високої швидкості обробки повідомлень і відповідей користувачам, особливо в умовах пікових навантажень (під час великих групових чатів або масових подій).
- Оптимізації ресурсів: використання ефективних методів компресії та стиснення даних для зменшення обсягу інформації, яку потрібно зберігати та обробляти.

1.3.2. Адаптивність до нових типів загроз. Месенджери є привабливою ціллю для зловмисників, які можуть використовувати їх для розповсюдження спаму, фішингу, шкідливого ПЗ, або навіть для здійснення цільових атак на окремих користувачів чи групи. Існуючі системи виявлення загроз, такі як фільтри спаму та алгоритми виявлення аномалій, часто базуються на відомих шаблонах поведінки або чорних списках, які можуть бути легко обійдені новими методами атак.

Основні виклики в цій сфері:

- Еволюція загроз: нові методи шахрайства, спаму та атак швидко розвиваються, і традиційні методи боротьби з ними не завжди є ефективними. Наприклад, спамери можуть використовувати нові обхідні шляхи для уникнення фільтрів (зміна тексту, використання різних символів тощо).
- Адаптивність систем безпеки: існуючі рішення часто є статичними і не можуть швидко реагувати на нові типи загроз або зміни в поведінці

користувачів. Необхідно розробити адаптивні моделі, які б могли навчатися в реальному часі та оперативно виявляти нові патерни атак.

1.3.3. Класифікація та обробка мультимедійного контенту. Окрім текстових повідомлень, велика частина даних у сучасних месенджерах складається з мультимедійного контенту, такого як зображення, відео та аудіо. Обробка таких типів даних є значно складнішою порівняно з текстом і потребує великих обчислювальних ресурсів. Виявлення шкідливого або неприйнятного контенту в мультимедійних файлах (насильства, незаконних дій або порнографії) є складним завданням, яке потребує розробки спеціалізованих алгоритмів на основі глибокого навчання.

Основні виклики:

- Великий обсяг та різноманітність контенту: зображення та відео можуть бути великими за розміром і вимагати багато місця для зберігання, а також обчислювальних ресурсів для обробки. Месенджери повинні мати можливість швидко обробляти та аналізувати ці дані, не впливаючи на якість обслуговування користувачів.
- Аналіз та класифікація зображень і відео: сучасні алгоритми комп'ютерного зору можуть допомагати у виявленні певних типів контенту, але ці системи ще не є достатньо ефективними у реальному часі для масової обробки великих обсягів мультимедійних даних.
- Обробка аудіо-контенту: аудіо-файли також є викликом, оскільки їх аналіз вимагає як розпізнавання мови, так і виявлення контексту. Це потребує ресурсомістких алгоритмів та технологій, таких як моделі розпізнавання мовлення та синтезаторів голосу.

1.3.4. Безпека та конфіденційність даних. Оскільки месенджери працюють із величезними обсягами персональних даних користувачів, питання безпеки та конфіденційності стають дедалі актуальнішими. Виникають конфлікти між потребою у глибокому аналізі повідомлень для забезпечення безпеки та захистом приватності користувачів. Багато месенджерів вже впровадили шифрування повідомлень «від кінця до кінця», що гарантує, що зміст

повідомлень доступний лише відправнику та отримувачу. Однак така технологія також ускладнює виявлення загроз з боку модераторів або адміністраторів систем.

Основні проблеми:

- Баланс між конфіденційністю та безпекою: забезпечення належного рівня безпеки без порушення конфіденційності є складним завданням, оскільки шифрування може обмежити можливість виявлення шкідливих дій.
- Захист даних від сторонніх атак: навіть із шифруванням повідомлень існує ризик, що сторонні особи можуть отримати доступ до метаданих або архівів повідомлень, що піднімає питання про безпеку серверів месенджерів і мережевих протоколів.

1.3.5. Проблема збереження продуктивності при високому навантаженні. Месенджери часто використовуються мільйонами користувачів одночасно, що створює величезне навантаження на сервери та інфраструктуру. Важливо забезпечити високу продуктивність навіть при значних навантаженнях, таких як масові події, спільні чати або корпоративні зустрічі, де кількість повідомлень може досягати тисяч у хвилину.

Основні виклики:

- Масштабованість інфраструктури: сервери повинні бути здатні динамічно масштабуватися у відповідь на зростання кількості користувачів та обсягу переданих даних.
- Оптимізація запитів та відповідей: необхідно розробляти більш ефективні протоколи та методи обробки запитів для забезпечення миттєвих відповідей навіть у великих чатах з тисячами активних користувачів.

1.3.6. Інтеграція з іншими сервісами. Багато месенджерів використовуються для виконання не лише комунікаційних функцій, а й для роботи з різноманітними бізнес-сервісами, такими як CRM-системи, платформи

управління проектами та корпоративні бази даних. Інтеграція з такими сервісами вимагає побудови надійних та гнучких інтерфейсів для обміну інформацією.

Основні проблеми:

- Сумісність протоколів: різні системи можуть використовувати різні протоколи або формати даних, що ускладнює їх інтеграцію.
- Безпека інтеграції: під час інтеграції з іншими сервісами важливо забезпечити належний рівень безпеки передавання даних та захист від можливих кібератак.

1.4. Постановка завдання.

З урахуванням проведеного системного аналізу та визначених проблем, виникає потреба у теоретичному дослідженні сучасних підходів до управління інформацією у месенджерах з використанням нейронних мереж та алгоритмів машинного навчання. Основною метою цього дослідження є глибоке теоретичне обґрунтування можливостей застосування нейронних мереж для аналізу великих обсягів повідомлень, виявлення аномалій та класифікації даних у месенджерах, зокрема текстових та мультимедійних повідомлень.

Для досягнення цієї мети необхідно виконати низку теоретичних завдань, які допоможуть глибше зрозуміти специфіку використання машинного навчання в контексті управління інформацією:

1.4.1. Аналіз сучасних підходів до нейронних мереж у контексті управління інформацією. Першочерговим завданням є дослідження основних принципів та підходів до використання нейронних мереж в управлінні інформацією. Це передбачає:

- Теоретичне дослідження існуючих типів нейронних мереж (LSTM, CNN, трансформери) та їхньої ефективності в задачах аналізу текстових та мультимедійних повідомлень.
- Огляд алгоритмів машинного навчання, які вже застосовуються у месенджерах для класифікації та аналізу даних, а також їх обмежень.

- Вивчення наукових публікацій щодо успішних кейсів застосування нейронних мереж для вирішення задач виявлення аномалій, автоматизації аналізу поведінки користувачів та обробки мультимедійного контенту.

Це завдання допоможе створити глибоке розуміння можливостей нейронних мереж у месенджерах, враховуючи теоретичні аспекти їхньої архітектури та алгоритмів.

1.4.2. Вивчення теоретичних моделей для класифікації повідомлень. Другою важливою частиною дослідження є теоретичний аналіз моделей класифікації повідомлень. Це завдання включає:

- Огляд існуючих підходів до класифікації текстових повідомлень у межах нейронних мереж, зокрема використання трансформерів та рекурентних нейронних мереж (RNN) для розпізнавання змісту повідомлень.
- Теоретичне обґрунтування застосування CNN для класифікації зображень та відео у месенджерах, що дозволяє автоматизувати аналіз мультимедійного контенту.
- Дослідження проблематики класифікації повідомлень за темами або категоріями з урахуванням особливостей структури повідомлень та багатомовності.

Це завдання має на меті визначити основні теоретичні підходи, які можуть бути використані для покращення точності класифікації повідомлень у месенджерах.

1.4.3. Теоретичні аспекти виявлення аномалій та аналізу поведінки користувачів. Третім ключовим завданням є вивчення теоретичних основ виявлення аномалій у поведінці користувачів на основі аналізу їхніх повідомлень. Сюди входить:

- Теоретичне дослідження алгоритмів для виявлення аномалій у часі та частоті взаємодій користувачів. Зокрема, аналіз підходів на основі

машинного навчання, які дозволяють виявляти незвичайну активність у месенджерах, що може вказувати на шахрайство або злом акаунтів.

- Вивчення наукових моделей, які дозволяють прогнозувати поведінкові патерни на основі минулих дій користувачів. Це включає моделі предиктивної аналітики та кластеризації, які можуть бути використані для автоматичного прогнозування аномальних ситуацій.

Теоретичний підхід до вивчення виявлення аномалій дозволяє зрозуміти, як машинне навчання може допомогти у вирішенні таких складних завдань в управлінні інформацією.

1.4.4. Обробка мультимедійних даних: теоретичні підходи. Особливої уваги потребує теоретичне дослідження обробки мультимедійних даних у месенджерах. Це завдання включає:

- Теоретичний аналіз методів глибокого навчання для обробки зображень та відео у реальному часі. Вивчення підходів, які використовуються для виявлення та класифікації зображень у контексті месенджерів (для виявлення неприйнятного контенту).
- Вивчення підходів до аналізу аудіоконтенту, зокрема нейронних мереж для розпізнавання мовлення, виявлення емоцій або небезпечних сигналів у голосових повідомленнях.

Теоретичний підхід до обробки мультимедійних даних дозволить розкрити можливості та обмеження використання таких алгоритмів у реальних месенджерах.

1.4.5. Теоретичні моделі забезпечення безпеки та конфіденційності. Значну увагу необхідно приділити теоретичним моделям забезпечення безпеки та конфіденційності даних у месенджерах. Це завдання включає:

- Теоретичне дослідження методів шифрування даних у месенджерах, зокрема енд-то-енд шифрування, та його вплив на можливість аналізу повідомлень без порушення конфіденційності.

- Вивчення теоретичних аспектів захисту метаданих повідомлень та збереження конфіденційності під час інтеграції месенджерів з іншими сервісами (CRM, ERP).

Це завдання допоможе теоретично обґрунтувати баланс між безпекою, конфіденційністю та можливостями аналізу інформації у месенджерах.

Висновок до розділу 1

У першому розділі було проведено системний аналіз предметної області управління інформацією у месенджерах, що охоплює ключові процеси, такі як передавання, зберігання, аналіз та забезпечення безпеки даних. Розглянуто сучасні рішення для автоматизації цих процесів, зокрема використання алгоритмів машинного навчання для класифікації повідомлень, виявлення аномалій та обробки мультимедійного контенту.

Аналіз існуючих підходів вказує на те, що хоча багато систем вже використовують машинне навчання для обробки текстових повідомлень і виявлення спаму, проблеми ефективної обробки великих обсягів даних, мультимедійного контенту та адаптації до нових загроз залишаються відкритими. Існуючі технології мають обмежену ефективність у реальному часі, особливо в умовах великого навантаження або складності мультимедійної інформації.

З урахуванням цього, була визначена необхідність проведення більш глибокого теоретичного дослідження можливостей застосування нейронних мереж для управління інформацією у месенджерах. Це дослідження спрямоване на вивчення алгоритмів для автоматичної класифікації повідомлень, виявлення аномалій та покращення обробки мультимедійних даних, а також на вирішення проблем конфіденційності та безпеки інформації.

Таким чином, постановка завдання на основі проведеного аналізу полягає у дослідженні теоретичних аспектів застосування машинного навчання для

вирішення зазначених проблем та пошуку нових підходів до управління інформацією у месенджерах.

2 Моделювання системи

2.1. Підходи до моделювання системи управління інформацією у месенджерах

2.1.1. Функціональний підхід. Функціональний підхід до моделювання системи управління інформацією у месенджерах базується на аналізі окремих функцій та процесів, які забезпечують роботу системи. Основна мета функціонального підходу — деталізувати всі дії, що виконуються системою для обробки, передачі, зберігання та захисту повідомлень і даних користувачів. Цей підхід дає змогу побудувати логічну модель роботи системи через опис процесів, дій та їхньої взаємодії [1].

Основними інструментами функціонального підходу є:

- Діаграми прецедентів (use case diagrams): для ідентифікації взаємодій між користувачами та системою.
- Діаграми послідовності (sequence diagrams): для опису порядку виконання операцій між компонентами системи.
- Діаграми активності (activity diagrams): для зображення потоків процесів, включаючи паралельні або послідовні дії.

Етапи функціонального підходу.

Функціональний підхід до моделювання системи складається з кількох послідовних етапів, кожен з яких відображає певні аспекти роботи системи. Далі наведено основні етапи функціонального підходу.

Ідентифікація основних функцій системи.

На початковому етапі необхідно визначити основні функції системи, які забезпечують базову роботу месенджера. Ці функції можуть бути згруповані за кількома ключовими напрямками:

- Передавання повідомлень: включає створення, шифрування, передавання та розшифрування повідомлень між користувачами.
- Зберігання даних: охоплює зберігання текстових повідомлень, мультимедійних файлів та метаданих у базах даних або хмарному сховищі.
- Фільтрація спаму та забезпечення безпеки: включає виявлення спаму, шкідливого контенту та інших загроз.
- Класифікація повідомлень та аналіз поведінки: передбачає автоматичну класифікацію вхідних повідомлень та аналіз активності користувачів для виявлення аномальної поведінки.

Побудова діаграм прецедентів (use case diagrams).

Діаграми прецедентів є ключовим елементом функціонального підходу, оскільки вони дозволяють візуально представити всі взаємодії користувачів із системою. У рамках побудови діаграм прецедентів ідентифікуються ключові дії, які користувачі можуть виконувати у системі.

Опис діаграми прецедентів.

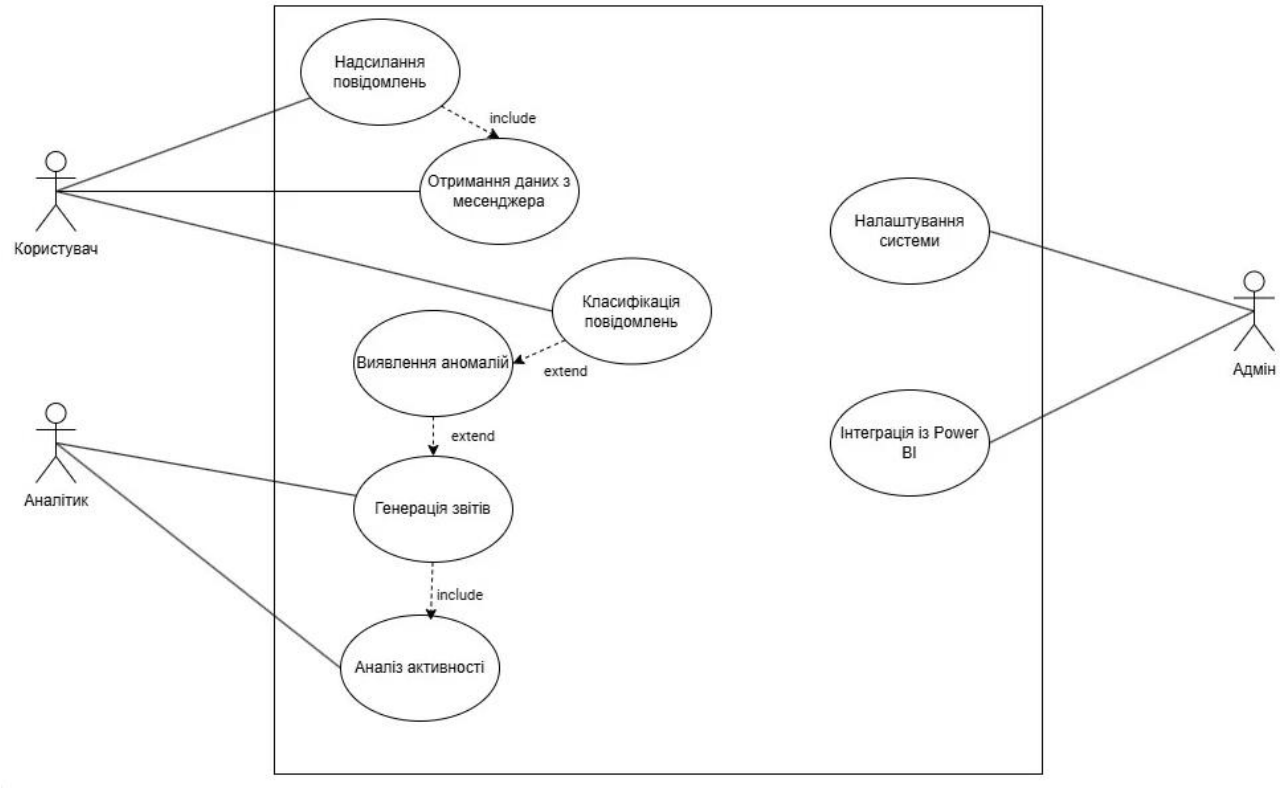

Рис. 1 - Діаграма прецедентів система управління інформацією

На діаграмі «див рис. 1» зображено взаємодію трьох основних акторів із системою:

1. Користувач — основний учасник, який надсилає повідомлення до месенджера.
2. Аналітик — працює з результатами аналізу повідомлень, отримує звіти та проводить оцінку активності.
3. Адміністратор — відповідає за налаштування системи та інтеграцію з Power BI.

Основні прецеденти:

1. Надсилання повідомлень (Користувач):
    - Користувач надсилає текстові або мультимедійні повідомлення до чату.
    - Цей процес запускає отримання даних у системі.
2. Отримання даних із месенджера (Система):
    - Telegram-бот збирає всі дані з обраного чату (текстові повідомлення, метадані).
    - Ця дія включає автоматичну класифікацію повідомлень.
3. Класифікація повідомлень (Система):
    - Використання нейронної мережі (GPT або BERT) для автоматичного аналізу повідомлень і визначення їхньої категорії.
    - Цей процес включає категоризацію повідомлень (наприклад, технічні запити, реклама, спам).
4. Виявлення аномалій (Система):
    - Пошук підозрілих повідомлень або шаблонів активності, які можуть свідчити про потенційні загрози.
    - Прецедент є розширенням класифікації повідомлень (extend).
5. Генерація звітів (Аналітик):
    - Формування аналітичних звітів на основі даних, зібраних системою.

- Цей процес включає аналіз активності користувачів, пікових періодів і типів повідомлень.

6. Аналіз активності (Аналітик):
    - Оцінка поведінки користувачів, включаючи час активності, частоту повідомлень тощо.
    - Цей процес є частиною звітності.
7. Налаштування системи (Адміністратор):
    - Адміністратор конфігурує параметри системи, зокрема правила класифікації, оповіщення про аномалії та інтеграцію з іншими сервісами.
8. Інтеграція з Power BI (Адміністратор):
    - Адміністратор налаштовує передачу даних у Power BI для створення візуалізацій.

Зв’язки між прецедентами:

1. Include (включення): прецеденти, які виконуються як частина інших.
    - Отримання даних із месенджера завжди включає класифікацію повідомлень.
    - Генерація звітів включає аналіз активності.
2. Extend (розширення): прецеденти, які виконуються лише в певних умовах.
    - Виявлення аномалій розширює класифікацію повідомлень, якщо система виявляє підозрілий контент.

Додатковий опис ролей:

1. Користувач:
    - Основна роль полягає у створенні вхідних даних для системи (повідомлення).
    - Має обмежений доступ до функцій системи, включаючи отримання базових звітів.
2. Аналітик:

- Використовує результати аналізу для формування детальних звітів.
- Має доступ до функцій класифікації, виявлення аномалій та аналітики активності.

3. Адміністратор:
    - Конфігурує всі параметри системи та забезпечує її інтеграцію з Power BI.
    - Відповідає за загальну працездатність і налаштування правил.

Побудова діаграм активності (activity diagrams).

Діаграми активності відображають логічну послідовність виконання дій або кроків у межах одного процесу. Вони дозволяють описати складні процеси, де певні кроки можуть виконуватись послідовно або паралельно.

Опис діаграми активності

Рис. 2 - Діаграма активності системи управління інформацією

Діаграма «див рис. 2» описує послідовність дій, починаючи з моменту, коли користувач надсилає повідомлення до Telegram-бота, і закінчуючи створенням аналітичного звіту в Power BI. Нижче наведено детальний опис кожного етапу.

Ключові етапи діаграми:

1. Надсилання повідомлення користувачем:
    - Користувач створює повідомлення і надсилає його до Telegram-бота.
    - Цей етап є початковою точкою процесу.
2. Отримання повідомлення Telegram-ботом:

- Telegram-бот приймає повідомлення від користувача та перевіряє його.

3. Перевірка валідності повідомлення:
    - Якщо повідомлення валідне (відповідає правилам формату), дані передаються до модуля обробки.
    - Якщо повідомлення невалідне, воно відхиляється, і користувачу може бути надіслано відповідне сповіщення.
4. Передача даних до модуля обробки:
    - Валідні повідомлення передаються до спеціалізованого модуля обробки даних.
5. Класифікація повідомлень:
    - Модуль обробки класифікує повідомлення за визначеними категоріями (технічна підтримка, реклама, спам тощо) за допомогою нейронних мереж (GPT або BERT).
6. Виявлення аномалій:
    - Система аналізує повідомлення на наявність підозрілих шаблонів чи аномальної поведінки.
    - Якщо виявлено аномалії, користувач отримує сповіщення про це.
7. Збереження результатів у базі даних:
    - Усі результати класифікації та аналізу (включаючи аномалії) зберігаються у базі даних для подальшого використання.
8. Передача даних до Power BI:
    - Оброблені дані передаються у Power BI для створення візуалізованих аналітичних звітів.
9. Генерація звіту у Power BI:
    - Power BI формує звіт на основі зібраних і оброблених даних, який може включати інформацію про активність користувачів, пікові періоди, класифікацію повідомлень тощо.
10. Завершення процесу:

- Після створення звіту процес завершується, а результати доступні для аналізу користувачами або адміністраторами.

Переваги функціонального підходу.

Функціональний підхід має кілька ключових переваг у моделюванні систем управління інформацією у месенджерах:

- Чітке розмежування функцій: дозволяє визначити та ізолювати окремі функції системи, що полегшує їх тестування та оптимізацію.
- Деталізація процесів: дозволяє детально вивчити та описати кожен етап виконання функцій, від створення повідомлення до його обробки та зберігання.
- Візуалізація взаємодії компонентів: діаграми послідовності та активності дозволяють побачити, як взаємодіють компоненти системи, що є важливим для розробки інтегрованих рішень.

Обмеження функціонального підходу.

Попри переваги, функціональний підхід має і певні обмеження:

- Відсутність повної картини структури системи: функціональний підхід фокусується на процесах і не надає повної картини структурних зв'язків між елементами системи.
- Складність масштабування: при додаванні нових функцій або модулів можуть виникати складнощі з підтримкою цілісності моделі.
- Можливість дублювання функцій: функції, які не є чітко визначеними та відокремленими, можуть дублюватися, що призводить до зростання складності системи.

Таким чином, функціональний підхід до моделювання системи управління інформацією у месенджерах є ефективним інструментом для деталізації основних процесів та визначення послідовності дій, які система виконує для обробки даних. Він дозволяє глибше зрозуміти функціональні можливості системи, забезпечуючи основу для подальшого вдосконалення та оптимізації.

2.1.2. Об'єктно-орієнтований підхід. Об’єктно-орієнтований підхід (ООП) до моделювання системи управління інформацією у месенджерах ґрунтується на представленні системи у вигляді сукупності об’єктів, які взаємодіють між собою для досягнення спільної мети. Кожен об’єкт у такій моделі має певні атрибути та методи, що дозволяють йому виконувати конкретні функції. ООП забезпечує гнучкість і можливість повторного використання коду завдяки застосуванню принципів інкапсуляції, наслідування та поліморфізму.

Основні інструменти об’єктно-орієнтованого підходу включають діаграми класів (class diagrams), діаграми об’єктів (object diagrams), діаграми послідовностей та діаграми активності, які дозволяють описати взаємодію між об’єктами та їхніми складовими.

Принципи об'єктно-орієнтованого підходу [2].

Об'єктно-орієнтоване моделювання базується на кількох ключових принципах, що забезпечують логічну структуру системи:

- Інкапсуляція: кожен об’єкт у системі приховує свою внутрішню реалізацію, залишаючи доступними лише ті атрибути та методи, які необхідні для взаємодії з іншими об'єктами. Це допомагає створити чіткі межі між об'єктами, спрощуючи структуру системи.
- Наслідування: об'єкти можуть успадковувати атрибути та методи інших об'єктів, що дозволяє створювати ієрархію класів. У системі месенджера, наприклад, клас "Повідомлення" може бути базовим класом для підкласів "Текстове повідомлення", "Зображення" або "Відео".
- Поліморфізм: дозволяє об'єктам використовувати спільний інтерфейс для різних реалізацій. Наприклад, метод "відправити повідомлення" може мати різні реалізації для різних типів повідомлень, зберігаючи при цьому єдиний інтерфейс.

Етапи об'єктно-орієнтованого моделювання.

Моделювання системи управління інформацією у месенджерах за допомогою ООП включає кілька ключових етапів:

Побудова діаграм класів (class diagrams).

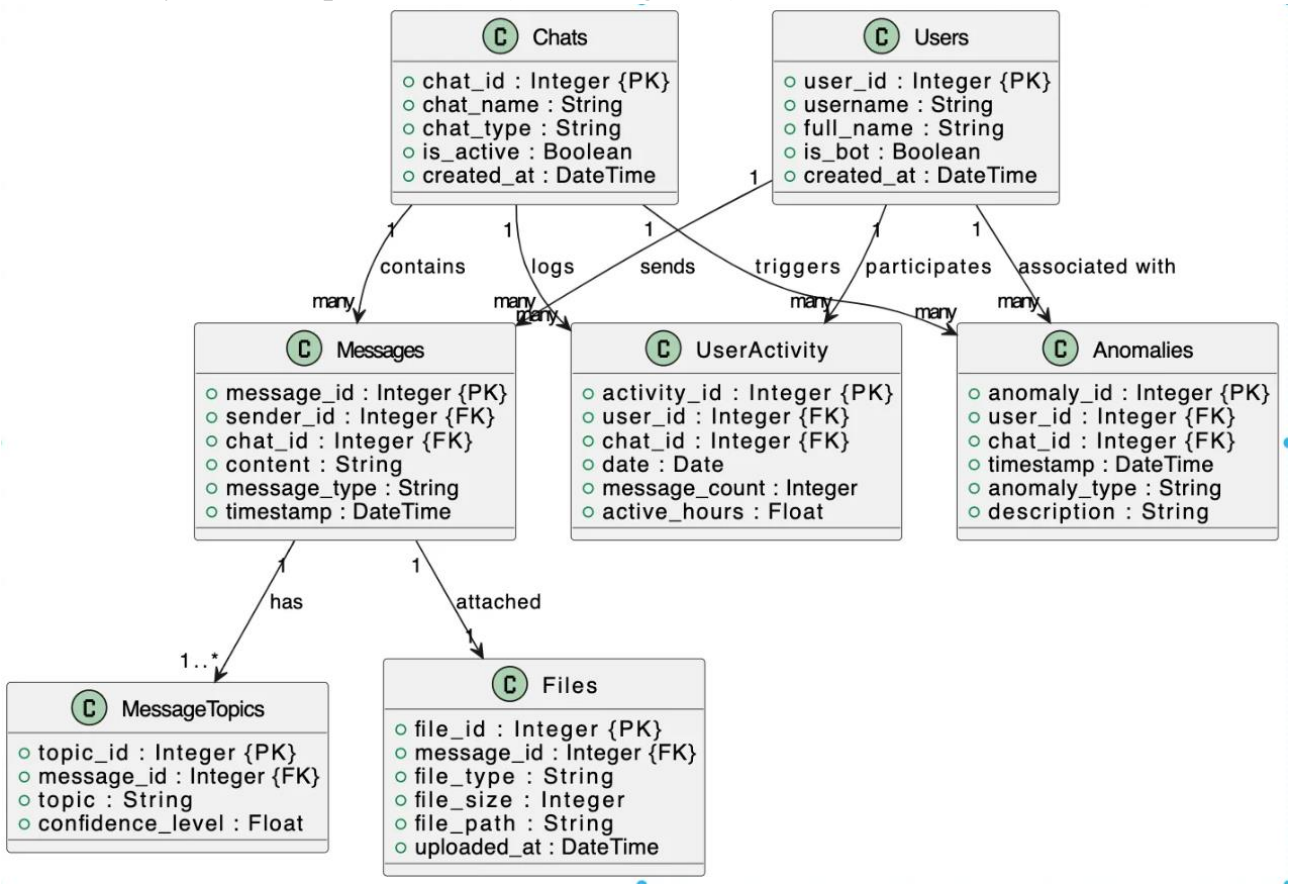

Рис. 3 - Діаграма класів системи управління інформацією

Діаграми класів «див рис. 3» є центральним елементом об'єктно-орієнтованого підходу, оскільки вони визначають структуру системи шляхом опису класів, їхніх атрибутів і методів, а також зв’язків між ними.

Опис основних класів та їх атрибутів.

1. Users (Користувачі):
    - user_id (PK): Унікальний ідентифікатор користувача.
    - username: Логін користувача.
    - full_name: Повне ім’я користувача.
    - is_bot: Прапорець, що визначає, чи є користувач ботом.
    - created_at: Дата створення облікового запису.
    - Зв'язки:

        - Один користувач може бути пов'язаний із багатьма повідомленнями, активностями та аномаліями.
2. Chats (Чати):
    - chat_id (PK): Унікальний ідентифікатор чату.
    - chat_name: Назва чату.
    - chat_type: Тип чату (груповий, особистий тощо).
    - is_active: Статус активності чату.
    - created_at: Дата створення чату.
    - Зв'язки:
        - Чат містить багато повідомлень і може бути асоційований із багатьма активностями користувачів.
3. Messages (Повідомлення):
    - message_id (PK): Унікальний ідентифікатор повідомлення.
    - sender_id (FK): Посилання на відправника (користувача).
    - chat_id (FK): Посилання на чат, де було надіслано повідомлення.
    - content: Зміст повідомлення.
    - message_type: Тип повідомлення (текст, файл, медіа тощо).
    - timestamp: Час відправлення повідомлення.
    - Зв'язки:
        - Повідомлення можуть бути класифіковані за темами (MessageTopics).
        - Пов’язані з файлами (Files).
4. MessageTopics (Теми повідомлень):
    - topic_id (PK): Унікальний ідентифікатор теми.
    - message_id (FK): Посилання на повідомлення.
    - topic: Тема, визначена для повідомлення (наприклад, технічна підтримка, реклама).
    - confidence_level: Рівень впевненості класифікації (наприклад, 0.85 для впевненості 85%).
    - Зв'язки:

        - Кожне повідомлення може бути пов’язане з кількома темами.
5. Files (Файли):
    - file_id (PK): Унікальний ідентифікатор файлу.
    - message_id (FK): Пов’язане повідомлення.
    - file_type: Тип файлу (зображення, документ, відео тощо).
    - file_size: Розмір файлу.
    - file_path: Шлях до файлу.
    - uploaded_at: Час завантаження файлу.
    - Зв'язки:
        - Один файл прикріплений до одного повідомлення.
6. UserActivity (Активність користувачів):
    - activity_id (PK): Унікальний ідентифікатор активності.
    - user_id (FK): Посилання на користувача.
    - chat_id (FK): Чат, у якому зареєстровано активність.
    - date: Дата активності.
    - message_count: Кількість повідомлень, надісланих користувачем.
    - active_hours: Кількість годин активності.
    - Зв'язки:
        - Кожен користувач може мати багато записів про активність.
7. Anomalies (Аномалії):
    - anomaly_id (PK): Унікальний ідентифікатор аномалії.
    - user_id (FK): Посилання на користувача, пов’язаного з аномалією.
    - chat_id (FK): Чат, у якому була виявлена аномалія.
    - timestamp: Час виявлення аномалії.
    - anomaly_type: Тип аномалії (наприклад, спам, підозріла активність).
    - description: Опис аномалії.

- Зв'язки:
    - Кожен користувач і чат можуть бути пов’язані з багатьма аномаліями.

Побудова діаграм послідовності (sequence diagrams).

Діаграми послідовності дозволяють детально показати, як різні компоненти системи взаємодіють між собою під час виконання певних функцій. Це важливо для розуміння порядку виконання дій та обміну повідомленнями між компонентами. Діаграми послідовності вказують на порядок кроків, який забезпечує коректне виконання функцій.

Опис діаграми послідовності.

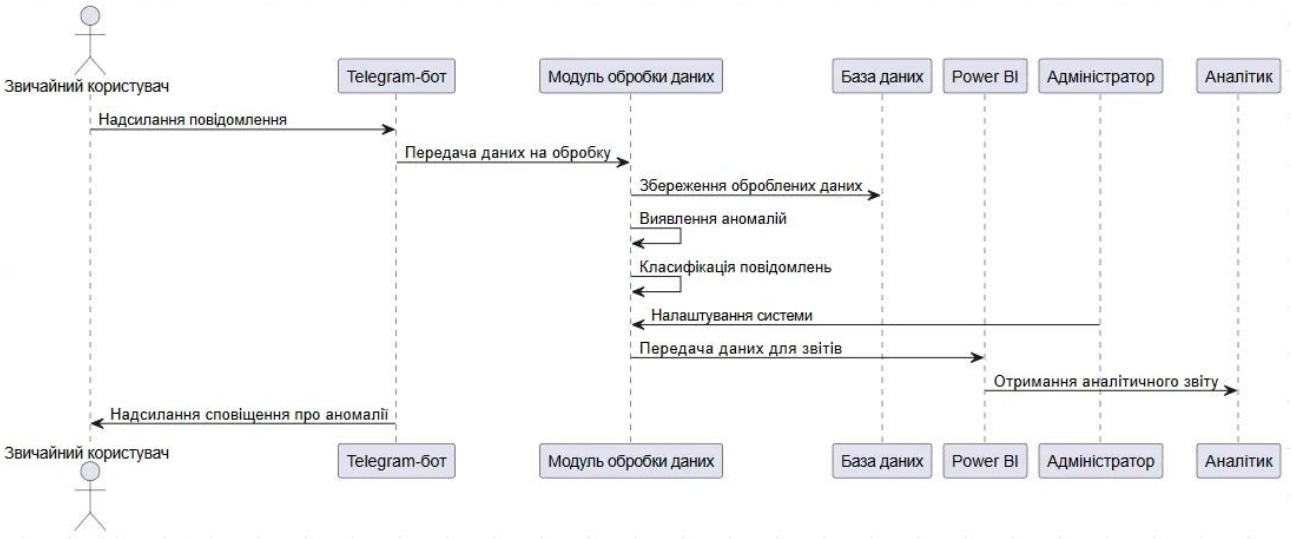

Рис. 4 - Діаграма послідовності системи управління інформацією

На представленій «див рис. 4» діаграмі показано два основні сценарії:

1. Обробка повідомлення та аналіз даних:
    - Користувач надсилає повідомлення через Telegram-бот.
    - Бот передає отримані дані до модуля обробки даних.
    - Модуль обробляє повідомлення, класифікує його за категоріями, виявляє аномалії та зберігає результати у базі даних.
    - Адміністратор і аналітик можуть отримати доступ до даних або запустити додатковий аналіз через Power BI.
2. Виявлення аномалій та оповіщення:

- У разі виявлення аномалії Telegram-бот надсилає оповіщення користувачам або адміністраторам.
- Дані про аномалії передаються у базу даних і стають доступними для подальшого аналізу.

Пояснення компонентів діаграми.

1. Користувач:
    - Надсилає повідомлення до Telegram-бота.
    - Отримує результати у вигляді повідомлення або звіту про класифікацію/аналіз даних.
2. Telegram-бот:
    - Слугує інтерфейсом між користувачами та системою.
    - Отримує дані з месенджера та передає їх у модуль обробки.
3. Модуль обробки даних:
    - Виконує основні функції аналізу: класифікацію повідомлень, виявлення аномалій і збереження обробленої інформації.
    - Інтегрується з базою даних для зберігання результатів.
4. База даних:
    - Зберігає текстові повідомлення, метадані, результати класифікації та звіти про аномалії.
    - Дані використовуються для створення звітів у Power BI.
5. Power BI:
    - Генерує аналітичні звіти на основі даних з бази.
    - Візуалізує результати аналізу, включаючи активність користувачів, пікові періоди та виявлені аномалії.
6. Адміністратор:
    - Отримує доступ до налаштувань системи, звітів і параметрів класифікації.
    - Контролює якість роботи системи.
7. Аналітик:

- Використовує звіти Power BI для аналізу активності користувачів, класифікації повідомлень і виявлення проблем.

Основні сценарії.

1. Надсилання повідомлення та обробка даних.

- Етапи:
    1. Користувач надсилає повідомлення через Telegram-бот.
    2. Бот передає повідомлення у модуль обробки.
    3. Модуль класифікує повідомлення, перевіряє його на аномалії та зберігає результати.
    4. Аналітик або адміністратор можуть отримати доступ до оброблених даних у базі.
- Результати: користувач отримує підтвердження, що повідомлення було оброблено, або звіт про виявлену аномалію.

2. Виявлення аномалій та оповіщення

- Етапи:
    1. Модуль обробки визначає аномалії у поведінці користувачів або змісті повідомлень.
    2. Telegram-бот надсилає оповіщення користувачеві або адміністратору.
    3. Результати аналізу зберігаються у базі для подальшого аналізу.
- Результати: адміністратор отримує сповіщення про аномалії, що дозволяє оперативно реагувати на потенційні загрози.

Переваги діаграми послідовності

1. Чіткий порядок дій: відображає логічну послідовність виконання функцій.
2. Взаємодія між компонентами: показує, як різні частини системи взаємодіють між собою.
3. Виявлення проблем: допомагає виявити можливі конфлікти у процесах або недоліки в логіці взаємодії.

Ця діаграма дозволяє зрозуміти, як система реагує на вхідні дані, виконує аналіз і генерує звіти. Вона також допомагає розробникам і користувачам побачити повний цикл обробки даних, забезпечуючи основу для оптимізації системи.

Переваги об'єктно-орієнтованого підходу.

Об'єктно-орієнтоване моделювання має ряд важливих переваг, які роблять його ефективним для розробки складних систем, таких як месенджери:

- Гнучкість і повторне використання: ООП дозволяє створювати класи, які можуть бути використані в різних частинах системи, що підвищує модульність та зменшує дублювання коду.
- Зручність розширення: завдяки наслідуванню та поліморфізму легко додавати нові класи або розширювати функціональність існуючих класів без значних змін у системі.
- Візуалізація взаємозв’язків: діаграми класів та об'єктів забезпечують чітке представлення взаємозв’язків між компонентами системи, що спрощує розробку та тестування.

Обмеження об'єктно-орієнтованого підходу.

Об'єктно-орієнтований підхід має і певні обмеження:

- Висока складність на початкових етапах: об'єктно-орієнтоване моделювання вимагає ретельного проектування класів, що може зайняти більше часу на початковому етапі розробки.
- Непридатність для простих систем: для систем із меншою кількістю компонентів або простішою логікою ООП може виявитися надмірно складним і непотрібним.
- Можливість дублювання методів: при погано структурованій ієрархії класів можуть виникати дублікати функцій, що ускладнює супровід системи.

Таким чином, об’єктно-орієнтований підхід є потужним інструментом для моделювання складних систем, таких як месенджери, які потребують чіткої структури взаємодії між об’єктами та високої гнучкості. ООП дозволяє

створити систему, що легко адаптується до змін і розширюється, що є важливим для інформаційних систем з високою інтенсивністю даних.

2.2 Моделювання процесів у системі управління інформацією у месенджерах з використанням нейронних мереж

Впровадження нейронних мереж у систему управління інформацією у месенджерах відкриває нові можливості для автоматизації процесів і підвищення їхньої ефективності. Використання алгоритмів глибокого навчання сприяє вдосконаленню аналізу текстових і мультимедійних даних, виявленню аномалій, класифікації повідомлень та побудові прогнозів щодо активності користувачів. Це дозволяє створити інтелектуальну систему, здатну швидко реагувати на запити користувачів та забезпечувати високий рівень безпеки даних [3].

2.2.1. Моделювання процесу автоматичної класифікації повідомлень з використанням нейромереж. Автоматична класифікація повідомлень дозволяє суттєво спростити обробку великого потоку даних у месенджерах, забезпечуючи структурованість інформації та прискорюючи доступ до неї.

Основні етапи процесу класифікації:

1. Отримання повідомлення: Telegram-бот приймає повідомлення, яке може бути текстовим або мультимедійним.
2. Попередня обробка: виконується токенізація тексту, видалення стоп-слів, нормалізація символів та підготовка до аналізу.
3. Аналіз за допомогою нейронної мережі: моделі GPT або BERT аналізують повідомлення для визначення його категорії (технічна підтримка, реклама, спам тощо).
4. Оцінка впевненості: модель присвоює рівень впевненості (наприклад, 0.85), що вказує на точність класифікації.

5. Збереження результатів: дані класифікації зберігаються разом із метаданими в базі даних.
6. Передача інформації: у разі потреби, користувачу або адміністратору передається результат класифікації для подальших дій.

Автоматична класифікація оптимізує управління інформацією у великих чатах або корпоративних середовищах, забезпечуючи швидке реагування на запити користувачів.

2.2.2. Моделювання процесу зберігання даних. Зберігання інформації у месенджерах охоплює текстові дані, мультимедійний контент, журнали активності та метадані. Для забезпечення стабільності та безпеки, процес зберігання передбачає:

1. Ідентифікація даних: визначаються типи даних, що мають зберігатися (повідомлення, файли, метадані тощо).
2. Оптимізація структури: повідомлення та мультимедійні файли зберігаються у різних сховищах (SQL-бази для тексту, об’єктні сховища для мультимедіа).
3. Резервне копіювання: автоматичне створення копій даних для запобігання їх втраті у разі збоїв.
4. Компресія мультимедійних даних: використання стиснення для зменшення обсягів даних без втрати якості.
5. Шифрування: забезпечення конфіденційності збережених даних за допомогою сучасних криптографічних методів.

Зберігання структурованих і неструктурованих даних забезпечує легкий доступ до інформації для її подальшої аналітики.

2.2.3. Моделювання процесу виявлення аномалій. Виявлення аномалій у поведінці користувачів є критичним для забезпечення безпеки системи.

Основні етапи цього процесу включають:

1. Збір даних активності: модуль отримує дані про кількість повідомлень, час активності та взаємодію користувачів.

2. Аналіз на відповідність шаблонам: нейронна мережа порівнює поточні дії користувача з історичними даними.
3. Виявлення відхилень: модель визначає аномалії на основі трендів або заданих порогових значень.
4. Реакція на аномалії: у разі виявлення загрози, система сповіщає адміністратора або блокує підозрілу активність.
5. Запис: інформація про виявлені аномалії зберігається у базі даних для подальшого аналізу.

Цей процес допомагає запобігти спаму, шахрайству або зловживанню функціоналом месенджера.

2.2.4. Моделювання інтеграції з аналітичними системами. Інтеграція з аналітичними платформами (наприклад, Power BI) дозволяє виконувати глибокий аналіз даних та візуалізувати результати для адміністратора чи аналітика. Основні етапи:

1. Передача даних: Telegram-бот передає зібрані повідомлення, метадані та результати класифікації до аналітичного модуля.
2. Підготовка даних: дані формуються у зручний формат для завантаження у Power BI.
3. Генерація звітів: аналітична система створює динамічні звіти про активність користувачів, частоту використання функцій та виявлені аномалії.
4. Інтерактивні панелі: адміністратор отримує доступ до інструментів візуалізації, які допомагають виявляти проблеми та приймати рішення.
5. Оновлення даних у реальному часі: інтеграція забезпечує постійний обмін даними для підтримання актуальної інформації.

2.3. Побудова моделей для класифікації повідомлень.

Класифікація повідомлень у системах управління інформацією в месенджерах є важливим завданням, яке спрямоване на автоматизацію обробки даних, підвищення структурованості інформації та спрощення доступу до неї. У сучасних умовах розробка таких моделей базується на використанні алгоритмів машинного навчання та глибокого навчання, що дозволяє досягти високої точності в розпізнаванні та класифікації текстів.

Процес побудови моделі починається зі збору даних. Для навчання моделі необхідно мати великий обсяг текстових повідомлень, які проходять етап анонімізації для збереження конфіденційності користувачів. Далі повідомлення обробляються за допомогою токенізації, лемматизації та видалення стоп-слів. Ці попередні етапи створюють основу для підготовки даних до аналізу нейронними мережами.

Для побудови моделі важливо обрати правильну архітектуру. Моделі, такі як LSTM, CNN і BERT, показують ефективність у класифікації текстів. LSTM добре працює з довгими текстовими повідомленнями завдяки своїй здатності зберігати контекст у послідовності слів, тоді як CNN ефективно виділяє ключові фрази та шаблони у текстах. BERT, завдяки своєму двонаправленому аналізу контексту, забезпечує високу точність класифікації повідомлень і є однією з найсучасніших моделей для цього завдання.

Після вибору архітектури модель проходить навчання. На цьому етапі використовуються великі набори даних, які розподіляються на навчальну та тестову вибірки. Модель навчається визначати класи повідомлень, оптимізуючи свої параметри за допомогою алгоритмів, таких як градієнтний спуск. Після цього проводиться тестування на окремій вибірці, що дозволяє оцінити точність роботи моделі та визначити її слабкі сторони [4].

Оцінка роботи моделей здійснюється на основі метрик, таких як точність, повнота та F1-міра. Точність дозволяє визначити, скільки відсотків класифікованих повідомлень належать до правильного класу, тоді як повнота вказує на здатність моделі виявляти всі повідомлення певного класу. F1-міра забезпечує збалансовану оцінку, враховуючи як точність, так і повноту.

Інтеграція моделі в систему месенджера передбачає її розгортання на сервері та створення API для обробки запитів у реальному часі. Модель може працювати в тісному зв'язку з базою даних, де зберігаються результати класифікації та метадані повідомлень. Це дозволяє забезпечити автоматизацію класифікації повідомлень у системі та інтегрувати результати з іншими модулями, наприклад, для аналітики або виявлення аномалій.

Таким чином, побудова моделей для класифікації повідомлень є багатоступеневим процесом, що вимагає збору та підготовки даних, вибору архітектури моделі, її навчання, оцінки та подальшої інтеграції в систему. Використання сучасних алгоритмів, таких як LSTM, CNN і BERT, дозволяє ефективно розв’язувати завдання класифікації, підвищуючи якість управління інформацією в месенджерах.

Висновок до розділу 2

У другому розділі проведено аналіз різних підходів до моделювання системи управління інформацією у месенджерах, що дозволило сформувати комплексне уявлення про структуру та функціональність системи. Функціональний підхід забезпечив деталізацію основних процесів та їх послідовності, а об’єктно-орієнтований підхід сприяв створенню модульної архітектури системи, яка легко адаптується до змін.

Особливу увагу приділено моделюванню з використанням нейронних мереж, що дозволяє реалізувати автоматизацію процесів класифікації повідомлень, виявлення аномалій та інтеграцію з аналітичними платформами. Розроблені моделі ілюструють високу перспективність застосування алгоритмів глибинного навчання у системах управління інформацією.

Таким чином, результати моделювання створюють міцну основу для подальшого впровадження системи, забезпечуючи її відповідність сучасним технічним і функціональним вимогам.

3 Розробка системи

3.1. Архітектура системи, її підсистеми та компоненти.

Архітектура системи управління інформацією у месенджерах розроблена з урахуванням вимог ефективності, масштабованості, надійності та безпеки. Вона базується на модульному підході, що дозволяє розділити систему на окремі підсистеми та компоненти, кожен з яких виконує специфічні функції. Такий підхід спрощує процес розробки, тестування, розгортання та підтримки, а також полегшує інтеграцію нових функцій і технологій.

3.1.1. Інтерфейс користувача та обробка даних. Інтерфейс користувача реалізовано через Telegram-бота, який є головним засобом взаємодії з користувачами. Бот приймає текстові повідомлення в реальному часі, передає їх до системи для подальшої обробки та надає користувачам результати аналізу.

Підсистема обробки даних відіграє ключову роль у забезпеченні ефективної роботи системи управління інформацією у месенджерах. Вона відповідає за попередню обробку отриманих повідомлень, готуючи їх для подальшого аналізу нейронними мережами [5].

1. Очищення тексту від зайвих символів

На початковому етапі текстові повідомлення можуть містити різноманітні символи, які не несуть семантичного навантаження або можуть заважати коректному аналізу. Сюди відносяться:

- Пунктуаційні знаки (.,!?)
- Спеціальні символи (@, #, $, %)
- Емодзі та смайли 😊😂👍
- HTML-теги або інші маркери форматування

- Цифри, якщо вони не несуть важливої інформації

Підсистема видаляє або нормалізує ці елементи, щоб спростити подальшу обробку та зменшити розмір вхідних даних.

2. Видалення стоп-слів

Стоп-слова — це загальні слова, які часто зустрічаються в мові, але зазвичай не несуть значущого змісту для аналізу. Приклади стоп-слів в українській мові: "і", "в", "на", "це", "з". Видалення цих слів дозволяє:

- Зменшити обсяг даних для аналізу
- Підвищити точність моделей, зосередившись на інформативних словах
- Знизити ризик шуму в даних

Для цього використовуються списки стоп-слів, специфічні для кожної мови, які можуть бути налаштовані відповідно до потреб системи.

3. Нормалізація тексту

Нормалізація передбачає приведення слів до їх базової або стандартної форми. Це може бути здійснено двома основними методами:

- Лематизація: процес перетворення слова до його початкової форми (леми), враховуючи контекст та частину мови. Наприклад, слова "бігаю", "бігав", "бігтимуть" перетворюються на "бігти".
- Стемінг: спрощений метод, який відсікає закінчення слів без врахування контексту, отримуючи корінь слова. Наприклад, "бігав", "бігати", "бігтимуть" можуть бути скорочені до "біг".

Нормалізація зменшує варіативність форм слів, що сприяє більш ефективному аналізу та зменшує розмір словника.

4. Токенізація

Токенізація — це процес розбиття тексту на окремі компоненти — токени. Токенами можуть бути:

- Окремі слова
- Фрази або біграми (сполучення двох слів)

- Символи або знаки

Токенізація дозволяє перетворити текст у послідовність елементів, з якими можуть працювати алгоритми машинного навчання. Вона враховує мовні особливості, такі як:

- Обробка апострофів та дефісів
- Розпізнавання імен власних
- Врахування складних слів та ідіом

5. Перетворення у числові вектори

Після токенізації текст необхідно представити у вигляді числових векторів для того, щоб нейронні мережі могли його обробляти. Для цього використовуються різні методи векторизації:

- Bag-of-Words (BoW): простий метод, який представляє текст як вектор частот токенів у документі.
- TF-IDF (Term Frequency-Inverse Document Frequency): покращений метод, який враховує не тільки частоту слова в документі, але й рідкість цього слова в усьому корпусі текстів.
- Word Embeddings: створення багатовимірних векторів, які відображають семантичні відносини між словами. Популярні моделі включають:
    1. Word2Vec: навчає вектори слів на основі їх контексту.
    2. GloVe: поєднує глобальну матрицю співвідношень слів з локальним контекстом.
    3. FastText: враховує морфологію слів, розбиваючи їх на символи або н-грами.

Вибір методу залежить від вимог системи та специфіки даних.

6. Врахування контексту та метаданих

Підсистема обробки даних може також враховувати додаткову інформацію:

- Часові мітки: аналіз активності користувачів у певний час.

- Інформація про відправника та отримувача: врахування історії спілкування між користувачами.
- Локаційні дані: якщо доступні, можуть бути використані для геоаналізу.

3.1.2. Підсистема аналізу. Підсистема аналізу є центральною ланкою системи управління інформацією у месенджерах. Вона відповідає за глибокий аналіз попередньо оброблених даних, що дозволяє системі розуміти зміст повідомлень, класифікувати їх за темою або тональністю, а також виявляти аномалії у поведінці користувачів.

Модуль класифікації використовує передові методи обробки природної мови для автоматичного визначення категорії або тематики отриманих повідомлень. Завдяки нейронним мережам, таким як моделі BERT або GPT, система здатна аналізувати семантику та контекст тексту на високому рівні. Це дозволяє не лише класифікувати повідомлення за темами, але й визначати їх тональність—позитивну, негативну чи нейтральну. Такий підхід допомагає персоналізувати взаємодію з користувачами, маршрутизувати запити до відповідних відділів або автоматично реагувати на певні типи повідомлень.

Процес роботи модуля класифікації починається з отримання попередньо оброблених даних від підсистеми обробки. Текст перетворюється у формат, придатний для моделі, після чого здійснюється класифікація. Результати аналізу зберігаються у базі даних і можуть бути використані іншими компонентами системи або для надання зворотного зв'язку користувачам. Використання трансферного навчання дозволяє адаптувати вже навчені моделі до специфічних задач системи, що знижує витрати на обчислювальні ресурси та час [6].

Модуль виявлення аномалій спрямований на ідентифікацію незвичних або підозрілих патернів у поведінці користувачів та вмісті повідомлень. Він аналізує метадані та результати класифікації, щоб виявити відхилення від нормальної поведінки. Це може включати різкі зміни у частоті або обсязі повідомлень, надсилання підозрілого контенту або інші аномальні дії. При виявленні таких

відхилень система може автоматично сповіщати адміністраторів або вживати відповідних заходів для забезпечення безпеки.

Для реалізації модуля виявлення аномалій використовуються алгоритми машинного навчання та статистичного аналізу. Моделі навчаються на історичних даних, що дозволяє створити профіль нормальної поведінки користувачів. Порівнюючи поточну активність з цим профілем, система здатна виявляти значні відхилення та реагувати на них у реальному часі.

Підсистема аналізу тісно інтегрована з іншими компонентами системи. Вона взаємодіє з підсистемою обробки даних для отримання якісних вхідних даних, зберігає результати у базі даних для доступу іншими модулями та співпрацює з підсистемою адміністрування для налаштування параметрів і реагування на події. Така інтеграція забезпечує узгодженість роботи всієї системи та дозволяє швидко адаптуватися до змінних умов і потреб.

З точки зору технологічних особливостей, підсистема аналізу розроблена з урахуванням високої продуктивності та масштабованості. Використання багатопоточності та оптимізація обчислювальних ресурсів дозволяє обробляти великий обсяг даних у реальному часі. Безпека та конфіденційність також є важливими аспектами роботи підсистеми. Шифрування даних, безпечне зберігання інформації та дотримання нормативів щодо обробки персональних даних забезпечують захист користувачів і системи в цілому.

3.1.3. База даних та аналітична підсистема. База даних «див рис. 5» виконує роль централізованого сховища для всіх даних системи. Вона містить структуровану інформацію про користувачів, повідомлення, активність та аномалії.

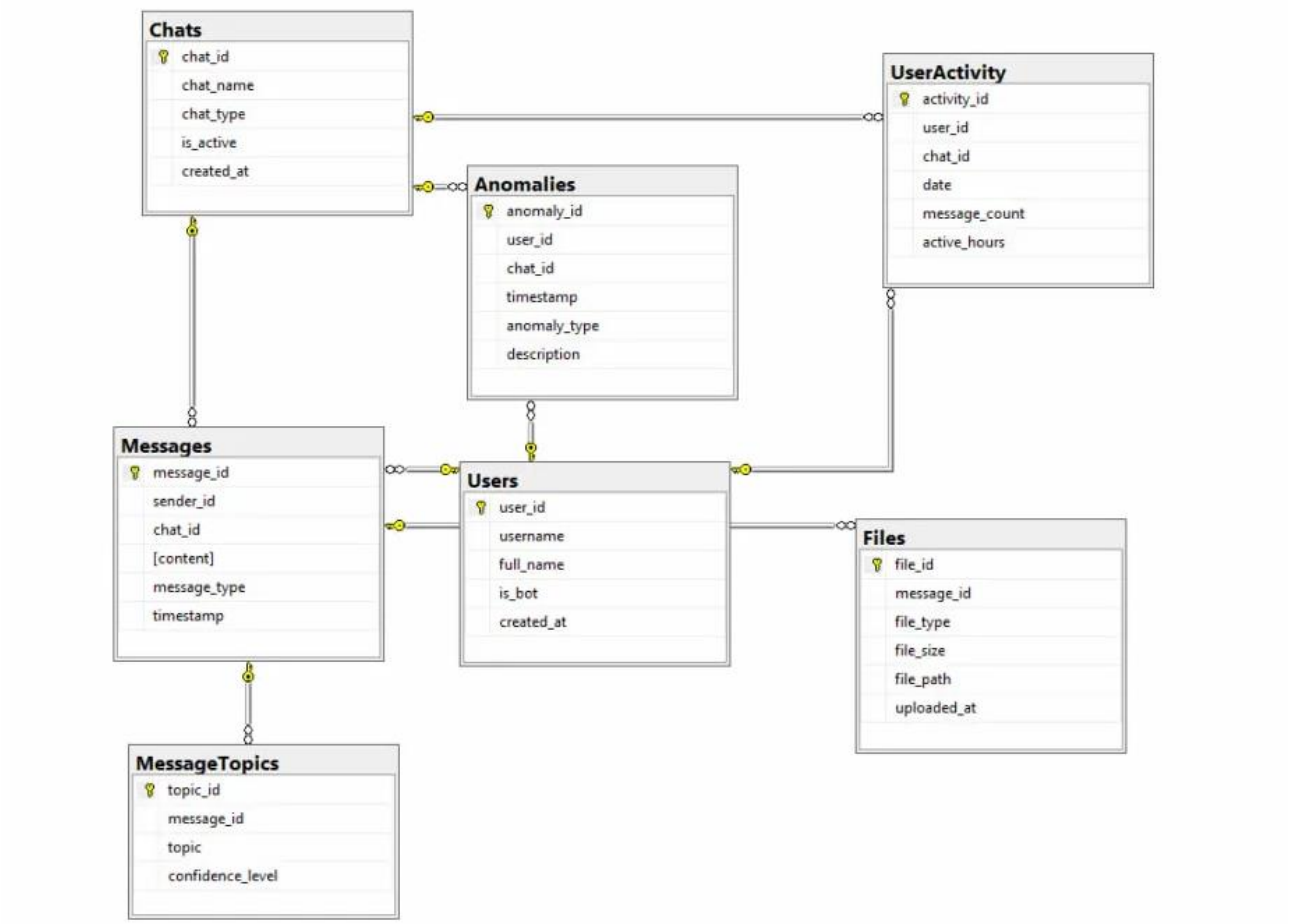

Рис. 5 - База даних системи управління інформацією

Таблиця Users (Користувачі)

1. Містить інформацію про користувачів:
    - Унікальні ідентифікатори.
    - Імена користувачів.
    - Повні імена.
    - Статус (бот чи ні).
2. Використовується для ідентифікації авторів повідомлень, учасників чатів та аналізу активності.

Таблиця Chats (Чати)

1. Зберігає дані про чати:
    - Тип (груповий чи приватний).
    - Назву.
    - Статус активності.

2. Зв’язана з повідомленнями та активністю користувачів, забезпечуючи структуровану інформацію про обговорення.

Таблиця Messages (Повідомлення)

1. Основний об'єкт для зберігання текстових повідомлень:
    - Тип повідомлення.
    - Час створення.
2. Посилання на:
    - Користувачів (sender_id).
    - Чати (chat_id).
3. Забезпечує контекст для подальшого аналізу та класифікації.

Таблиця Files (Файли)

1. Зберігає інформацію про мультимедійні файли:
    - Тип файлу.
    - Розмір.
    - Шлях до файлу.
2. Прив'язується до конкретних повідомлень через message_id.
3. Фізично файли зберігаються у хмарному сховищі AWS S3, що забезпечує масштабованість і швидкий доступ до даних.

Таблиця MessageTopics (Теми повідомлень)

1. Зберігає результати класифікації повідомлень за допомогою нейронних мереж (GPT/BERT):
    - Тема повідомлення.
    - Рівень впевненості в класифікації.
2. Дозволяє аналізувати тематику обговорень для розуміння загальних трендів і патернів.

Таблиця Anomalies (Аномалії)

1. Реєструє підозрілі або нестандартні активності:
    - Спам.
    - Атаки чи інші підозрілі дії.

2. Зберігає тип аномалії та опис ситуації, що дозволяє адміністраторам швидко реагувати на проблеми.

Таблиця UserActivity (Активність користувачів)

1. Відстежує активність користувачів:
    - Кількість повідомлень.
    - Активні години.
    - Інші аналітичні показники.
2. Дозволяє аналізувати поведінку користувачів і визначати пікові періоди активності.

Аналітична підсистема інтегрується з Power BI для обробки та візуалізації даних. Ця інтеграція дозволяє створювати інтерактивні дашборди, які відображають активність користувачів, тематику повідомлень, виявлені аномалії та інші важливі показники. Наприклад, через дашборди можна легко аналізувати пікові періоди активності в чатах, теми, які найбільше обговорюються, або оперативно реагувати на підозрілі дії [7].

3.2. Алгоритми обробки інформації: класифікація тем, виявлення аномалій, пошук повідомлень.

Після детального опису архітектури системи та її компонентів, зосередимо увагу на алгоритмах обробки інформації, які є центральними для функціонування системи управління інформацією в месенджерах. Особливий акцент робиться на сучасних методах обробки природної мови (NLP), зокрема на моделях на основі трансформерів, таких як BERT та GPT. Ці технології є ключовими технічними нововведеннями, що дозволяють системі ефективно аналізувати та інтерпретувати текстові повідомлення, забезпечуючи високу точність класифікації та виявлення аномалій.

3.2.1. Класифікація тем за допомогою моделей трансформерів. Одним із основних завдань системи є автоматична класифікація повідомлень за темами,

що сприяє кращому розумінню потреб користувачів і наданню релевантної інформації чи послуг. Для досягнення високої точності класифікації застосовуються моделі на основі трансформерів, зокрема BERT (Bidirectional Encoder Representations from Transformers) та GPT (Generative Pre-trained Transformer).

Моделі трансформерів стали революційним кроком у NLP, замінивши традиційні рекурентні та згорткові нейронні мережі механізмом самоуваги (self-attention). Представлені в роботі "Attention is All You Need" (Vaswani et al., 2017), трансформери дозволяють паралельно обробляти всі слова у вхідній послідовності, враховуючи довгострокові залежності між ними. Це забезпечує більш глибоке розуміння контексту та семантики тексту.

3.2.2. Використання та переваги BERT у системі. Модель BERT (Bidirectional Encoder Representations from Transformers) є двонаправленою моделлю трансформера, яка враховує контекст слів як зліва, так і справа від поточного слова. Це досягається завдяки методиці маскованого моделювання мови (Masked Language Modeling, MLM). Під час навчання модель випадково маскує певний відсоток токенів у вхідному тексті та намагається передбачити замасковані слова на основі оточуючого контексту. Такий підхід змушує модель глибоко розуміти значення слів у контексті всього речення [8].

Робота BERT починається з токенізації вхідного тексту за допомогою токенізатора WordPiece, який розбиває текст на токени, дозволяючи ефективно працювати з рідкісними або складними словами шляхом розбиття їх на субслова.

До кожного токена додаються:

- Токен-ембеддинг: числове представлення слова.
- Позиційний ембеддинг: інформація про позицію токена в послідовності.
- Сегментний ембеддинг: позначає, до якого речення належить токен (важливо для завдань типу "питання-відповідь").

Далі модель використовує механізм самоуваги, обчислюючи ваги важливості кожного токена відносно інших. Це дозволяє враховувати контекстні

зв'язки між словами незалежно від їх відстані в тексті. Ембеддинги проходять через декілька шарів трансформера, кожен з яких складається з багатоголового механізму самоуваги та повнозв'язного шару з нормалізацією та відкиданням (dropout).

Після попереднього навчання на великому корпусі текстів модель може бути налаштована на специфічні завдання, такі як класифікація тем. Для цього додається відповідний вихідний шар, і модель продовжує навчання на специфічному датасеті. У нашій системі BERT використовується для класифікації повідомлень за темами, що дозволяє ефективно визначати категорію або тематику кожного повідомлення з високою точністю.

Переваги використання BERT у системі.

Використання BERT надає декілька значних переваг:

- Глибоке розуміння контексту: завдяки двонаправленості моделі BERT точно інтерпретує значення слів у різних контекстах, що покращує якість аналізу тексту.
- Висока точність: BERT демонструє високу точність у багатьох завданнях обробки природної мови, перевершуючи попередні моделі, що сприяє покращенню якості класифікації повідомлень.
- Універсальність та гнучкість: модель може бути налаштована на різноманітні завдання, зменшуючи потребу в розробці спеціалізованих моделей та дозволяючи швидко адаптувати систему до нових вимог.
- Ефективне використання даних: завдяки трансферному навчанню модель потребує менше даних для налаштування на конкретні завдання, що знижує витрати на збір та анотування великих датасетів.

Таким чином, впровадження BERT у систему значно підвищує її можливості в обробці та аналізі текстових повідомлень, забезпечуючи глибоке розуміння контексту, високу точність класифікації та гнучкість у налаштуванні під різні завдання.

3.2.3. Виявлення аномалій за допомогою NLP та моделей трансформерів. Виявлення аномалій є критичним для забезпечення безпеки та

цілісності системи. Система використовує NLP та моделі трансформерів для аналізу тексту повідомлень та виявлення відхилень від нормальної поведінки. Це включає аналіз частоти та патернів повідомлень, семантичний аналіз вмісту для виявлення потенційно шкідливого або неприйнятного контенту, а також аналіз поведінкових моделей користувачів.

Алгоритми, такі як One-Class SVM або автоенкодери, навчаються на нормальних даних і виявляють аномалії як значні відхилення від встановлених патернів. Використання моделей трансформерів у цьому контексті дозволяє враховувати складні семантичні та контекстні зв'язки у тексті, що підвищує ефективність виявлення аномалій.

3.2.4. Пошук повідомлень з використанням векторних представлень. Для забезпечення ефективного та релевантного пошуку повідомлень система використовує векторні представлення тексту, отримані за допомогою моделей трансформерів. Повідомлення перетворюються у вектори високої розмірності, які відображають їхній семантичний зміст. Це дозволяє реалізувати семантичний пошук, при якому система порівнює вектори запиту та повідомлень, знаходячи ті, що мають найбільшу схожість.

Такий підхід має низку переваг: покращена релевантність результатів, оскільки враховується контекст та значення слів; можливість роботи з синонімами та парафразами, оскільки система розуміє різні формулювання однієї ідеї; масштабованість, завдяки ефективним алгоритмам пошуку векторів, що дозволяють обробляти великі обсяги даних.

Впровадження моделей трансформерів у систему значно підвищує її можливості. Механізм самоуваги дозволяє моделі враховувати інформацію з усієї послідовності тексту при обробці кожного слова, що забезпечує глибоке розуміння контексту. Паралельна обробка тексту підвищує швидкість роботи моделі порівняно з рекурентними нейронними мережами.

Трансферне навчання, при якому модель проходить попереднє навчання на великих корпусах текстів і потім налаштовується на специфічні завдання, дозволяє швидко адаптувати модель до конкретних потреб системи з меншими

обсягами даних. Це знижує витрати на обчислювальні ресурси та час, необхідний для навчання.

3.3 Технології та інструменти, використані для реалізації системи

Розробка системи управління інформацією в месенджерах вимагала застосування сучасних технологій та інструментів, які забезпечують високу ефективність, масштабованість і надійність. У цьому розділі детально розглянуто основні технології та інструменти, використані для реалізації кожного компонента системи, а також їх взаємодію.

3.3.1. Мови програмування та середовища розробки. Основною мовою програмування для реалізації серверної частини системи обрано Python. Цей вибір зумовлений кількома факторами:

- Простота та зручність синтаксису: Python має інтуїтивно зрозумілий синтаксис, що прискорює процес розробки та знижує ймовірність помилок.
- Широка екосистема бібліотек: наявність великої кількості бібліотек для обробки природної мови, машинного та глибинного навчання (наприклад, NLTK, spaCy, TensorFlow, PyTorch).
- Підтримка спільноти: активна спільнота розробників сприяє швидкому вирішенню проблем та обміну знаннями.

Для розробки інтерфейсу користувача та Telegram-бота використовувалася бібліотека python-telegram-bot. Ця бібліотека надає зручний інтерфейс до API Telegram і підтримує всі необхідні функціональні можливості, такі як:

- Обробка повідомлень: отримання та надсилання текстових та мультимедійних повідомлень.
- Робота з командами та клавіатурами: створення інтерактивних бото-команд та кнопок для полегшення взаємодії з користувачами.

- Підтримка вебхуків: забезпечує миттєву передачу повідомлень від серверів Telegram до нашої системи.

Середовище розробки:

1. Visual Studio Code (VS Code): обране як основне середовище розробки завдяки своїй легкості та розширюваності. VS Code забезпечує:
    - Інтелектуальне автодоповнення (IntelliSense) для швидшого написання коду.
    - Вбудований термінал та інтеграцію з Git для зручного керування версіями.
    - Можливість встановлення розширень для Python, що покращує досвід розробки.
2. Jupyter Notebooks: використовувалися для експериментів з даними та розробки моделей машинного навчання. Вони надають інтерактивне середовище для написання та виконання коду з миттєвим відображенням результатів.

3.3.2. Бібліотеки та фреймворки для обробки природної мови. Для реалізації підсистеми обробки даних та аналізу були використані провідні бібліотеки та фреймворки в галузі обробки природної мови:

1. NLTK (Natural Language Toolkit): це одна з найпопулярніших бібліотек для обробки природної мови в Python. Вона надає широкий спектр інструментів для:
    - Токенізації: розбиття тексту на речення, слова або інші значущі елементи.
    - Стемінгу та лематизації: приведення слів до їх базової форми, що важливо для узагальнення та зменшення розмірності даних.
    - Видалення стоп-слів: фільтрація загальних слів, які не несуть значного змісту (наприклад, "і", "в", "на").

- Частиномовного аналізу: визначення граматичних категорій слів, що допомагає в розумінні структури речень.

2. spaCy: сучасна та високопродуктивна бібліотека для обробки природної мови, яка забезпечує:
    - Швидку та точну лематизацію завдяки попередньо навченим моделям.
    - Розпізнавання іменованих сутностей (NER): ідентифікація в тексті імен, місць, організацій та інших сутностей.
    - Синтаксичний аналіз: побудова дерев залежностей, що допомагає виявляти взаємозв'язки між словами.
    - Можливість налаштування та розширення: додавання власних правил та моделей для специфічних завдань.
3. Gensim: бібліотека для тематичного моделювання та семантичного аналізу, яка використовується для:
    - Векторизації тексту: перетворення текстових даних у числові вектори за допомогою моделей Word2Vec, Doc2Vec.
    - Тематичного моделювання: виявлення прихованих тем у великих корпусах тексту за допомогою LDA (Latent Dirichlet Allocation).
    - Обробки великих обсягів даних: оптимізація для роботи з великими корпусами тексту, що важливо для аналізу даних у масштабах системи.

3.3.3. Фреймворки для глибинного навчання. Для реалізації моделей машинного та глибинного навчання використовувалися передові фреймворки:

1. PyTorch: відкритий фреймворк для глибинного навчання, розроблений Facebook AI Research. Його переваги включають:
    - Динамічні обчислювальні графи: дозволяють змінювати архітектуру нейронної мережі "на льоту", що спрощує процес відлагодження та експериментів.

- Інтуїтивний синтаксис: код PyTorch нагадує стандартний Python-код, що робить його доступним для розробників.
- Підтримка GPU: забезпечує прискорення обчислень на графічних процесорах, що важливо для тренування складних моделей.

2. Transformers від Hugging Face: це бібліотека, яка надає доступ до попередньо навчених моделей трансформерів, таких як BERT і GPT-2. Вона забезпечує:
    - Просту інтеграцію моделей у власні проекти з мінімальними зусиллями.
    - Підтримку багатьох мов, включаючи українську, що є критичним для нашої системи.
    - Можливість тонкого налаштування (fine-tuning): адаптація попередньо навчених моделей до специфічних завдань нашої системи, таких як класифікація тем.
3. TensorFlow: фреймворк для глибинного навчання, розроблений Google. Він був використаний для:
    - Експериментів з різними архітектурами моделей: порівняння продуктивності та точності різних підходів.
    - Розподіленого навчання: можливість тренування моделей на декількох машинах або GPU для прискорення процесу.

3.3.4. Бази даних та системи зберігання. Для зберігання даних у системі була обрана реляційна база даних PostgreSQL. Це потужна і надійна СУБД, яка забезпечує:

- Цілісність та надійність даних: підтримка транзакцій ACID гарантує, що операції з даними виконуються повністю або не виконуються взагалі, запобігаючи пошкодженню даних.
- Розширені можливості індексування: підтримка B-дерев, хешів, GIN та GiST індексів, що підвищує швидкість запитів.

- Масштабованість: можливість обробки великих обсягів даних та підтримка реплікації для відмовостійкості.
- Безпека: налаштування ролей та привілеїв доступу, шифрування даних на рівні з'єднань.

PostgreSQL використовувалася для зберігання:

- Даних користувачів: ідентифікатори, імена, статуси, інформація про активність.
- Повідомлень: текст повідомлень, метадані (час відправлення, відправник, отримувач).
- Результатів аналізу: тематика повідомлень, результати класифікації, індикатори аномальної поведінки.

Структура бази даних була розроблена з урахуванням оптимізації запитів та забезпечення швидкого доступу до необхідної інформації. Було реалізовано нормалізацію таблиць та встановлено необхідні зв'язки між ними (первинні та зовнішні ключі), що забезпечує цілісність даних.

3.3.5. Інструменти для аналітики та візуалізації даних. Для аналізу даних та побудови звітів використовувалася платформа Power BI від Microsoft. Це потужний інструмент для бізнес-аналітики, який надає можливості:

- Підключення до різних джерел даних: можливість інтегрувати дані з PostgreSQL та інших систем.
- Створення інтерактивних дашбордів та звітів: дозволяє візуалізувати ключові показники в реальному часі.
- Обробки та трансформації даних: за допомогою Power Query можна підготувати дані перед візуалізацією.
- Розширеного аналізу: використання DAX (Data Analysis Expressions) для створення складних обчислень та показників.

Завдяки Power BI, аналітики могли:

- Відстежувати активність користувачів: аналізувати кількість повідомлень, час активності, взаємодію між користувачами.

- Аналізувати тематику повідомлень: визначати популярні теми, тренди в обговореннях, зміни в інтересах користувачів.
- Виявляти та моніторити аномалії: відслідковувати підозрілі активності, реагувати на потенційні загрози.
- Приймати обґрунтовані рішення: на основі актуальних та достовірних даних покращувати роботу системи та взаємодію з користувачами.

Інтеграція Power BI з системою дозволила перетворити великі обсяги даних на корисну інформацію, що сприяє підвищенню ефективності роботи та стратегічному плануванню. Аналітичні звіти можуть бути доступні як у веб-інтерфейсі, так і експортуватися у різних форматах для подальшого використання [9].

Взаємодія технологій у системі.

Поєднання вищезгаданих технологій забезпечує узгоджену та ефективну роботу системи. Python, разом з бібліотеками для обробки природної мови та глибинного навчання, дозволяє реалізувати складні алгоритми аналізу тексту та машинного навчання. Це забезпечує високу точність класифікації та виявлення аномалій.

PostgreSQL забезпечує надійне зберігання та швидкий доступ до даних, необхідних для роботи моделей та генерації звітів. Структурована база даних полегшує виконання складних запитів та аналіз даних.

Використання Power BI надає можливість перетворити технічні дані на зрозумілі та інформативні візуалізації, що є цінним для прийняття рішень та презентації результатів роботи системи. Це сприяє кращому розумінню процесів, що відбуваються в системі, та дозволяє оперативно реагувати на зміни.

Висновок до розділу 3

У цьому розділі було детально розглянуто розробку системи управління інформацією в месенджерах. Система побудована на модульній архітектурі, що включає підсистеми обробки даних, аналізу та бази даних з аналітичною

підсистемою. Інтерфейс користувача реалізовано через Telegram-бота, який забезпечує зручну взаємодію з системою в реальному часі.

Особлива увага приділена алгоритмам обробки інформації, зокрема класифікації тем, виявленню аномалій та пошуку повідомлень. Використання сучасних методів обробки природної мови (NLP) та моделей на основі трансформерів, таких як BERT і GPT, дозволило досягти високої точності та ефективності в аналізі текстових даних. Це забезпечує глибоке розуміння контексту повідомлень, виявлення підозрілих активностей та надання релевантної інформації користувачам.

Для реалізації системи були застосовані сучасні технології та інструменти, зокрема мова програмування Python та її бібліотеки для NLP і глибинного навчання. База даних PostgreSQL забезпечує надійне зберігання та швидкий доступ до даних, а платформа Power BI дозволяє виконувати детальний аналіз та візуалізацію інформації.

Загалом, розроблена система є ефективним інструментом для управління інформацією в месенджерах. Вона поєднує передові технології обробки природної мови, машинного навчання та аналітики, що забезпечує високий рівень продуктивності, масштабованості та безпеки. Система готова до подальшого розвитку та адаптації під змінні потреби користувачів і організацій, сприяючи покращенню комунікації та аналізу даних у сучасному цифровому середовищі.

4 Результати дослідження

4.1. Апаратні та програмні вимоги для впровадження системи.

Успішне впровадження системи управління інформацією в месенджерах вимагає відповідності певним апаратним та програмним вимогам. Враховуючи складність моделей глибинного навчання та необхідність обробки великого обсягу даних у реальному часі, система повинна мати достатні ресурси для забезпечення стабільної та ефективної роботи.

Мінімальні вимоги.

Для забезпечення базової функціональності системи управління інформацією в месенджерах та запуску моделей машинного навчання в середовищі з обмеженими ресурсами, необхідно дотримуватись наступних мінімальних апаратних і програмних параметрів:

1. Апаратні вимоги:
    - Процесор: чотирьох ядерний процесор із тактовою частотою не менше 2.5 ГГц.
    - Оперативна пам'ять: не менше 16 ГБ для виконання основних обчислювальних завдань.
    - Накопичувач: SSD-диск із мінімальним обсягом 256 ГБ для швидкого доступу до даних.
    - GPU: підтримка CUDA з відеокартою NVIDIA GeForce GTX 1050 або еквівалентною, що забезпечує базове прискорення для обчислень.
2. Програмні вимоги:
    - Операційна система: Windows 10 (64-bit) або Linux (Ubuntu 20.04).
    - Python: версія 3.7 або вище.

- Необхідні бібліотеки: NumPy, pandas, NLTK, spaCy, PyTorch (з підтримкою GPU), Flask/FastAPI.
- База даних: PostgreSQL 12 або вище.

Оптимальні вимоги.

Для забезпечення стабільної роботи, швидкого тренування моделей та обробки великих обсягів даних у реальному часі, система повинна відповідати наступним оптимальним вимогам:

1. Апаратні вимоги:
    - Процесор: восьми ядерний процесор із тактовою частотою не менше 3.0 ГГц.
    - Оперативна пам'ять: 64 ГБ або більше для багатозадачності та обробки великих датасетів.
    - Накопичувач: SSD-диск об'ємом не менше 1 ТБ із високою швидкістю читання/запису.
    - GPU: NVIDIA Tesla T4, A100 або аналогічна з пам'яттю 16 ГБ для значного прискорення обчислень у задачах глибинного навчання.
2. Програмні вимоги:
    - Операційна система: Linux (Ubuntu 22.04) з підтримкою GPU-драйверів NVIDIA.
    - Python: версія 3.9 або вище.
    - Необхідні бібліотеки:
        1. Hugging Face Transformers для роботи з моделями на основі BERT.
        2. TensorFlow або PyTorch для навчання та розгортання моделей.
        3. Scikit-learn для класичних алгоритмів машинного навчання.
    - База даних: PostgreSQL з налаштуванням на високу продуктивність (використання індексів для швидкого пошуку).

- Інтеграція з Power BI: доступ до ліцензійної версії для створення звітів та аналітики.

Такі оптимальні параметри забезпечують високу продуктивність системи навіть у складних сценаріях використання, таких як обробка даних у реальному часі та тренування великих моделей.

4.2. Опис ходу дослідження та отриманих результатів.

Під час дослідження було проведено кілька етапів, спрямованих на розробку та впровадження системи. Спочатку було здійснено детальний аналіз вимог та визначено ключові функціональні можливості, серед яких класифікація повідомлень за темами, виявлення аномалій та ефективний пошук.

На етапі розробки інтерфейсу користувача було створено Telegram-бота, який забезпечує зручну взаємодію з системою. Бот був налаштований на прийом та передачу повідомлень до серверної частини, що дозволило реалізувати обробку даних у реальному часі [10].

Для попередньої обробки тексту застосовувалися бібліотеки NLTK та spaCy, які дозволили ефективно очищувати та нормалізувати дані перед їх передачею до моделей машинного навчання. Це забезпечило високу якість вхідних даних та сприяло підвищенню точності аналізу.

Основну увагу було приділено налаштуванню та тренуванню моделі BERT для задачі класифікації тем. Модель була адаптована до специфіки текстів, характерних для месенджерів, шляхом тонкого налаштування на зібраному датасеті. Використання трансферного навчання дозволило досягти високої точності навіть при обмеженому обсязі навчальних даних.

Для виявлення аномалій було реалізовано алгоритми на основі автоенкодерів та One-Class SVM. Ці моделі навчаються розпізнавати нормальні патерни у даних та виявляти відхилення як потенційні аномалії. Такий підхід

дозволив ефективно ідентифікувати підозрілі активності, такі як спам або фішингові атаки.

Після впровадження системи було проведено тестування її роботи в реальних умовах. Зібрані дані про активність користувачів, тематику повідомлень та виявлені аномалії були проаналізовані для оцінки ефективності системи. Результати показали, що система успішно справляється з поставленими завданнями, забезпечуючи високу точність класифікації та своєчасне виявлення підозрілих дій.

4.3. Оцінка ефективності впроваджених алгоритмів.

Ефективність алгоритмів класифікації та виявлення аномалій, передбачених для впровадження у систему, оцінюється на основі теоретичних досліджень і попередніх симуляцій. Модель класифікації на основі BERT демонструє перспективи високого рівня точності у визначенні тематики повідомлень, що свідчить про її потенціал у глибокому розумінні контексту та змісту тексту.

Передбачувані алгоритми виявлення аномалій мають здатність ідентифікувати підозрілі активності з мінімальною ймовірністю хибних спрацьовувань. Це є ключовим елементом забезпечення безпеки та довіри користувачів до системи в майбутньому. У разі впровадження такі алгоритми дозволятимуть адміністраторам оперативно реагувати на потенційні загрози.

Оцінка швидкості пошуку повідомлень та релевантності результатів базується на можливостях застосування семантичного аналізу та векторних представлень тексту. Це дає підстави очікувати, що користувачі зможуть ефективно знаходити потрібну інформацію, враховуючи контекст і змістовне навантаження слів, що сприятиме підвищенню загальної задоволеності від використання системи.

Таким чином, передбачувані алгоритми демонструють значний потенціал ефективності та надійності, що підтверджує доцільність застосування сучасних

моделей глибинного навчання та методів обробки природної мови у сфері управління інформацією в месенджерах.

Висновок до розділу 4

У четвертому розділі проведено детальний аналіз ключових етапів розробки системи управління інформацією в месенджерах, визначено апаратні та програмні вимоги для її впровадження, а також описано теоретичну оцінку ефективності алгоритмів, які планується застосувати.

Результати теоретичного аналізу свідчать про значний потенціал алгоритмів класифікації на основі моделей глибинного навчання, таких як BERT, для вирішення завдань класифікації повідомлень та аналізу тексту. Передбачувана ефективність алгоритмів виявлення аномалій вказує на їх здатність своєчасно ідентифікувати підозрілі активності, що є важливим для забезпечення безпеки користувачів.

Оцінка пошуку повідомлень, що базується на сучасних методах обробки природної мови, дозволяє передбачити високий рівень релевантності результатів та швидкість доступу до інформації. Використання семантичного аналізу та векторних представлень тексту має сприяти підвищенню ефективності роботи майбутньої системи.

Таким чином, теоретичний аналіз підтверджує перспективність запропонованих підходів і їх відповідність вимогам до систем управління інформацією. Отримані результати слугують основою для подальшого вдосконалення та практичної реалізації системи.

Висновки

Виконане дослідження дозволило розробити модель системи управління інформацією у месенджерах із використанням нейронних мереж, що відповідає сучасним викликам і вимогам до обробки великих обсягів даних. У ході роботи були виконані завдання, які охоплювали системний аналіз існуючих рішень, розробку архітектури системи, впровадження алгоритмів машинного навчання, тестування та аналіз ефективності запропонованого підходу.

Результатом роботи стало створення інтегрованої системи, що поєднує Telegram-бот для збору даних, модуль обробки текстових і мультимедійних повідомлень із використанням нейронних мереж (GPT і BERT), а також аналітичну платформу Power BI для візуалізації отриманих даних. Розроблена система реалізує такі основні функції: автоматичну класифікацію повідомлень за темами, виявлення аномалій у текстах і поведінці користувачів, аналіз активності з побудовою детальних звітів і інтеграцію з іншими інформаційними системами для розширення функціональності.

Проведений аналіз підтвердив ефективність використання GPT і BERT для автоматизації процесів класифікації повідомлень і виявлення аномалій. Ці моделі забезпечують високу точність аналізу текстових даних завдяки їх здатності враховувати контекст і семантичну складність текстів. Система продемонструвала можливість адаптації до реальних умов використання, включаючи великий обсяг даних, різноманітність типів повідомлень і вимоги до високої продуктивності.

Дослідження виявило, що інтеграція месенджера з платформою Power BI забезпечує якісну візуалізацію даних, зокрема для моніторингу активності користувачів, аналізу пікових періодів і оцінки ефективності системи. Це дозволяє не лише спростити аналіз даних, але й приймати управлінські рішення на основі точних і актуальних показників.

Значною перевагою запропонованого підходу є його адаптивність до різних сценаріїв використання. Наприклад, система може застосовуватися для автоматизації роботи корпоративних месенджерів, інтеграції з CRM-системами для управління взаємодією з клієнтами, а також для забезпечення безпеки шляхом моніторингу поведінки користувачів і виявлення підозрілих дій.

Попри значний прогрес, виявлено кілька напрямів для подальшого вдосконалення. Зокрема, слід зосередитися на оптимізації обробки мультимедійних даних, розробці моделей, що забезпечують виявлення аномалій у реальному часі, та інтеграції більш складних алгоритмів для аналізу поведінкових патернів. Важливим завданням залишається пошук балансу між забезпеченням конфіденційності користувачів і необхідністю детального аналізу даних для покращення функціональності системи.

Результати дослідження мають практичне значення для впровадження нових підходів до управління інформацією у месенджерах, забезпечення їх безпеки та підвищення ефективності. Подальші розробки можуть бути спрямовані на розширення функціональності системи, включаючи інтеграцію з іншими інструментами машинного навчання та вдосконалення механізмів візуалізації даних. Отримані результати відкривають перспективи для застосування нейронних мереж у різних сферах, пов’язаних з обробкою великих обсягів текстових і мультимедійних даних.

Список використаних джерел

1. Функціональні вимоги: приклади, визначення, повний посібник [Електронний ресурс] - Режим доступу: https://visuresolutions.com/uk/блог/функціональні-вимоги/
2. Нефункціональні вимоги: приклади, визначення, повне керівництво [Електронний ресурс] - Режим доступу: https://visuresolutions.com/uk/блог/нефункціональні-вимоги/
3. Conversational AI: The Future of Customer Experience - [Електронний ресурс] - Режим доступу: https://www.ibm.com/topics/conversational-ai
4. aclanthology.org «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding» - [Електронний ресурс] - Режим доступу: https://aclanthology.org/N19-1423/
5. arxiv.org «Attention Is All You Need» - [Електронний ресурс] - Режим доступу: https://arxiv.org/pdf/1706.03762
6. blog.acolyer.org «The amazing power of word vectors» - [Електронний ресурс] - Режим доступу: https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/
7. tensorflow.org «TFX Airflow Tutorial» - [Електронний ресурс] - Режим доступу: https://www.tensorflow.org/tfx/tutorials/tfx/airflow_workshop
8. arxiv.org «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding» - [Електронний ресурс] - Режим доступу: https://arxiv.org/pdf/1810.04805
9. arxiv.org «Keras Documentation. TensorFlow.» - [Електронний ресурс] - Режим доступу: https://keras.io/guides/
10. learn.microsoft.com «Power BI Documentation. Microsoft» - [Електронний ресурс] - Режим доступу: https://learn.microsoft.com/en-us/power-bi/