

**ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**  
**Навчально-науковий інститут економіки, управління, права та**  
**інформаційних технологій**  
**Кафедра інформаційних систем та технологій**

# **КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня вищої освіти магістр

на тему: «**Модель віртуального асистента на основі телеграм-боту**  
**та GPT-4o-mini**»

Виконав: здобувач вищої освіти  
за освітньою програмою  
Інформаційні управляючі системи та  
технології  
спеціальності 126 Інформаційні  
системи та технології  
ступеня вищої освіти магістр  
групи 126ІСТ\_мд\_2023  
Перевозкін Артем Сергійович  
Керівник: Слюсарь Ігор Іванович  
Рецензент: Муравльов Володимир  
В'ячеславович

**Полтава – 2024 року**

## ВСТУП

Актуальність теми кваліфікаційної роботи обумовлена стрімким розвитком технологій штучного інтелекту та їх застосуванням у різних сферах, зокрема в автоматизації взаємодії з користувачами через месенджери, одним з яких є Telegram. Його функціонал дозволяє створювати гнучкі та потужні рішення для автоматизації бізнес-процесів за допомогою ботів. Завдяки відкритій API та можливості інтеграції з різними зовнішніми сервісами, даний месенджер став оптимальною платформою для розробки інтелектуальних чат-ботів, здатних працювати з великим обсягом даних та надавати користувачам різноманітні послуги в автоматичному режимі.

Водночас, розвиток технологій глибокого навчання та обробки природної мови, зокрема на базі архітектури Generative Pre-trained Transformer (GPT), надає нові можливості для створення високотехнологічних віртуальних асистентів. Прикладом такої моделі є GPT-4o mini, що здатна обробляти та генерувати текстові відповіді в режимі реального часу. Інтеграція цієї моделі з телеграм-ботом відкриває нові горизонти для створення інтелектуальних систем, які можуть виконувати складні завдання, відповідаючи на запити користувачів, аналізуючи їх потреби та допомагаючи в прийнятті рішень та ін.

*Зв'язок роботи з науковими програмами, темами.* Робота відповідає дослідженням в межах науково-дослідної ініціативної тематики «Організаційно-методологічні аспекти впровадження інформаційно-комунікаційних систем і технологій в управлінні діяльністю сучасних організацій та підприємств за умов переходу до цифрової економіки» (ДРН 0123U105060, 2023-2028 рр.), що реалізується на кафедрі інформаційних систем та технологій, тематиці досліджень навчально-дослідної лабораторії інтелектуальних систем, комп'ютерних мереж та інтернет речей кафедри інформаційних систем та технологій Полтавського державного аграрного університету.

*Метою* кваліфікаційної роботи є підвищення ефективності автоматизованої взаємодії з користувачами за рахунок інтелектуальних функцій боту.

Завданнями кваліфікаційної роботи є:

- визначення особливостей використання Telegram API;
- інтеграція GPT-4o-mini та месенджера;
- аналіз взаємодії GPT-4o-mini з ботом месенджера;
- обґрунтування вибору серверних рішень для хостингу моделі;
- формування рекомендацій щодо інтеграції месенджера з генеративним

штучним інтелектом.

*Об'єктом дослідження* є процес функціонування чат-боту.

*Предметом дослідження* є інтеграція месенджера з генеративним штучним інтелектом.

*Методами* дослідження в рамках визначення інструментарію для сегментації тексту, і техніко-економічного обґрунтування прийнятих рішень використовувався аналітичний метод досліджень, а для розробки моделі віртуального асистента – моделювання.

*Інформаційна база* кваліфікаційної роботи сформована з ресурсів, що містять інформацію про Telegram API, функціонал ботів месенджерів, нейронні мережі GPT.

*Елементи наукової новизни* роботи полягають в розробці моделі віртуального асистента на основі телеграм-боту та генеративного штучного інтелекту.

*Практична значущість* роботи полягає в розробці рекомендацій щодо інтеграції месенджера з генеративним штучним інтелектом – можуть бути використані для подальших досліджень за даною тематикою та при проектуванні хмарних сервісів.

*Апробація результатів* відбувалася в рамках VI Міжнародної студентської наукової конференції «Тренди та перспективи розвитку мультидисциплінарних досліджень» (жовтень 2024 р., м. Кременчук) та VII

Міжнародної студентської наукової конференції «Розвиток суспільства та науки в умовах цифрової трансформації» (листопад 2024 р., м. Тернопіль).

*Структура кваліфікаційної роботи* логічно пов'язана з завданнями досліджень і містить вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг пояснювальної записки кваліфікаційної роботи складає 66 сторінок формату А4. Вона містить 9 рисунків 5 таблиць.

## РОЗДІЛ 1

### АНАЛІЗ ОСОБЛИВОСТЕЙ ПЛАТФОРМИ TELEGRAM ДЛЯ СТВОРЕННЯ ВІРТУАЛЬНИХ АСИСТЕНТІВ

#### 1.1 Загальна характеристика Telegram як платформи для створення ботів

Telegram – це сучасний месенджер, який поєднує високу швидкість роботи, стабільність, безпеку та багатий набір функцій для користувачів і розробників. Однією з головних особливостей платформи є підтримка ботів, які стали універсальним інструментом для автоматизації багатьох процесів у межах месенджера. Завдяки простоті створення та інтеграції ботів Telegram є зручною платформою як для бізнесу, так і для індивідуальних розробників.

Telegram є однією з найбільш популярних платформ обміну повідомленнями у світі. Станом на сьогодні, месенджер налічує понад 800 мільйонів активних користувачів на місяць, і це число постійно зростає. Його популярність пояснюється унікальним поєднанням функціональності, зручності, високої безпеки і гнучких можливостей для розробників, які активно використовують цю платформу для створення ботів і автоматизації бізнес-процесів [1].

Telegram має користувачів у більш ніж 150 країнах, включаючи регіони з високим рівнем цифровізації, такі як Європа, Північна Америка, та Азія. У таких країнах, як Індія та Індонезія Telegram входить до трійки найбільш завантажуваних додатків для спілкування. У Західній Європі він часто використовується як альтернатива WhatsApp завдяки своїм розширеним можливостям у сфері захисту даних.

Особливою популярністю Telegram користується серед молоді та технологічно обізнаних користувачів, які цінують можливості кастомізації месенджера, такі як налаштування тем, ботів та інтеграцій із зовнішніми сервісами.

Telegram зазнав значного збільшення кількості користувачів упродовж останніх років. Наприклад, у 2021 р. платформа досягла позначки в 500 мільйонів активних користувачів після масових міграцій через побоювання щодо конфіденційності в інших месенджерах. Лише за перший квартал 2023 р. Telegram залучив додаткові 100 мільйонів користувачів.

Основними факторами зростання популярності Telegram є його здатність задовольняти потреби користувачів у конфіденційності та прозорості. Відсутність таргетованої реклами, наявність чіткої політики обробки даних та забезпечення високого рівня безпеки стали важливими аспектами, що приваблюють людей, які цінують свою приватність. Зміни в умовах обслуговування WhatsApp у 2021 р. стали додатковим стимулом для багатьох користувачів перейти на Telegram, який, на відміну від конкурентів, не вимагає таких суворих умов і пропонує більшу свободу у використанні.

Telegram також приваблює користувачів завдяки своїм можливостям для організації комунікації великих спільнот. Наприклад, можливість створювати групи до 200000 учасників та канали з необмеженою кількістю підписників дає користувачам величезні можливості для обміну інформацією та налагодження контактів. Це робить платформу незамінною як для малих груп, так і для великих організацій, медіа-компаній, а також для різноманітних освітніх установ.

Telegram активно використовується в бізнесі, освіті та медіа. Канали Telegram стали потужним інструментом для просування товарів і послуг, адже вони дозволяють безпосередньо взаємодіяти з кінцевими споживачами, поширюючи інформацію про нові продукти та послуги. У сфері освіти Telegram активно застосовується для організації дистанційного навчання, управління студентськими спільнотами, а також для публікації навчальних матеріалів, що робить його важливим інструментом у освітньому процесі. ЗМІ використовують Telegram-канали для оперативного поширення новин, що дозволяє їм швидко реагувати на зміни у світі та інформувати аудиторію без затримок.

Особливо важливим для Telegram є його можливість інтеграції з іншими сервісами через API, що робить платформу привабливою для розробників. Telegram надає розширені можливості для створення чат-ботів, що автоматизують різні бізнес-процеси, організують комунікацію з клієнтами або допомагають у виконанні внутрішніх завдань компаній.

На відміну від таких конкурентів, як WhatsApp та Viber, Telegram має ряд суттєвих переваг. Зокрема, можливість зберігати необмежену кількість повідомлень у хмарному сховищі дає змогу користувачам мати доступ до історії чатів без обмежень. Крім того, платформа підтримує передачу великих файлів (до 2 ГБ), що є важливою перевагою при обміні документами чи мультимедійними матеріалами. Telegram також дозволяє використовувати канали та групи для трансляції інформації широкій аудиторії, що є важливим для організацій та брендів, які хочуть досягти максимальної охопленості. Завдяки можливості створювати складні автоматизовані рішення через боти, Telegram забезпечує високу ефективність у вирішенні задач, пов'язаних з автоматизацією бізнес-процесів, а також взаємодією з клієнтами.

Отже, Telegram зберігає лідерство серед комунікаційних платформ завдяки своїй універсальності, зручності та високому рівню безпеки. Це дозволяє йому залишатися популярним інструментом як для особистого спілкування, так і для бізнесу, освіти та медіа.

Telegram-боти представляють собою потужний інструмент для автоматизації завдань, обробки даних, інтеграції з зовнішніми системами та удосконалення взаємодії з користувачем. Завдяки своїй легкості впровадження, гнучким налаштуванням і доступності, боти в Telegram охоплюють широке коло застосувань – від простих чатів з автоматизованими відповідями до складних систем управління бізнесом.

Telegram-боти можуть виконувати такі наступні функції.

Обробка текстових запитів. Боти здатні відповідати на текстові запити користувачів, використовуючи заздалегідь визначені алгоритми або штучний інтелект для аналізу контексту. Наприклад, вони можуть відповідати на

питання про час роботи організації або надавати персоналізовану інформацію. Telegram-боти дозволяють інтегрувати сторонні API, бази даних або системи управління завданнями. Наприклад:

- підключення до CRM-систем для управління клієнтами;
- інтеграція з платіжними платформами для обробки онлайн-оплат;
- використання зовнішніх API для отримання даних (погода, курси валют, новини).

Автоматизація рутинних завдань. Боти можуть використовуватися для автоматизації регулярних процесів, таких як:

- надсилання нагадувань;
- формування щоденних звітів;
- збір і аналіз даних, наприклад, через опитування.

Мультимедіа і розширені функції. Telegram-боти підтримують мультимедіа (зображення, відео, аудіо), що дозволяє створювати інтуїтивно зрозумілі інтерфейси для користувачів [2]. Вони також можуть використовувати такі функції, як:

- інтерактивні клавіатури;
- картки з кнопками для голосування чи переходу до інших дій;
- надсилання геолокації.

Впровадження штучного інтелекту. Використання моделей машинного навчання, таких як GPT, дозволяє ботам працювати як віртуальні асистенти, аналізуючи запити і надаючи відповідь, максимально схожу на людську.

Telegram-боти знаходять розповсюджене використання в різних галузях завдяки своїй функціональності, зручності та ефективності. У бізнесі та комерції боти активно використовуються для оптимізації бізнес-процесів. Вони дозволяють клієнтам переглядати товари, оформляти замовлення, проводити оплату, а також автоматизувати відповіді на типові питання. Крім того, за допомогою ботів можна автоматизувати генерацію звітів для співробітників і керівництва, що значно спрощує управління.

В освіті Telegram-боти служать інструментом для розсилки навчальних матеріалів, організації тестувань і вікторин, планування навчальних занять і нагадування про них. Завдяки цьому вони стають важливим елементом сучасного навчального процесу.

У громадських та соціальних ініціативах боти допомагають організувати голосування, обробляти заявки на участь у заходах і поширювати інформацію про події. В індустрії розваг Telegram-боти використовуються для створення інтерактивних ігор, чат-ботів для спілкування або проведення соціальних опитувань, що робить їх незамінними в створенні цікавого контенту.

Серед переваг Telegram-ботів слід зазначити простоту їх створення завдяки зручному API, що дозволяє розробляти навіть складні боти з мінімальними витратами часу. Telegram не вимагає оплати за використання своїх ресурсів для роботи ботів, що робить цю платформу ще більш привабливою. Крім того, боти здатні обслуговувати десятки тисяч користувачів одночасно, що забезпечує їхню ефективність навіть у великих організаціях.

Telegram-боти не лише автоматизують завдання, а й допомагають економити час і ресурси. Вони забезпечують цілодобовий доступ до послуг, дозволяють ефективно обробляти великі обсяги даних і створюють комфортний користувацький досвід.

## **1.2 Особливості Telegram API для інтеграції ботів**

Telegram API є одним із найзручніших і найпопулярніших інструментів для розробки ботів, завдяки простоті використання, детальній документації та широким можливостям для інтеграції. Цей API дозволяє розробникам створювати автоматизовані системи, які можуть бути використані користувачами для взаємодії в межах Telegram. Його головна особливість

зводиться до того, що він є абсолютно безкоштовним і не вимагає встановлення додаткових програм чи модулів. API доступний через стандартні HTTP-запити, що робить його сумісним з будь-якою мовою програмування та платформою, яка підтримує роботу з веб-запитами.

Однією з головних рис Telegram API є підтримка двох способів обміну даними між сервером і платформою: опитування (long polling) та вебхуки (webhooks). Опитування дозволяє серверу бота періодично надсилати запити до Telegram-серверів, щоб отримати нові повідомлення чи оновлення [3]. Це простий метод, який не потребує складної конфігурації, але може мати затримки у відповіді. Натомість вебхуки забезпечують миттєву передачу даних з Telegram-серверів на сервер бота, що значно підвищує швидкість обробки запитів. Однак для цього необхідно налаштувати сервер із відкритим HTTPS-доступом.

Telegram API підтримує широкий спектр функцій для роботи з ботами. Основною функцією є обробка повідомлень користувачів. Боти можуть приймати текстові повідомлення, мультимедіа (зображення, відео, аудіо, документи), геолокацію, голосові повідомлення та інші типи даних. Крім того, API надає можливість створення інтерактивних елементів, таких як клавіатури, кнопки та інлайн-меню. Це дозволяє реалізовувати складні сценарії взаємодії з користувачами, наприклад, опитування, голосування або процеси вибору в декілька етапів.

Додатково API підтримує роботу з каналами, групами та супергрупами. Боти можуть автоматично публікувати повідомлення в каналах, модерувати чати або аналізувати активність учасників груп. Така функціональність є особливо корисною для адміністраторів великих спільнот, які прагнуть автоматизувати повсякденні задачі.

Ще однією важливою особливістю Telegram API є можливість підключення до інших платформ та зовнішніми базами даних. Завдяки цьому боти можуть використовувати сторонні API для отримання додаткової інформації чи виконання спеціалізованих завдань. Для таких інтеграцій

Telegram API надає спеціальні команди для надсилання HTTP-запитів і обробки результатів.

API також пропонує засоби безпеки, що важливо при роботі з конфіденційними даними. Зокрема, кожен бот має унікальний токен доступу, який генерується в процесі створення. Цей токен слід зберігати в таємниці, адже саме він забезпечує доступ до функцій бота.

Ще однією корисною функцією є підтримка мультимовності. Крім того, Telegram API дозволяє отримувати інформацію про мову інтерфейсу користувача, що спрощує адаптацію ботів до потреб різних регіонів.

Однією з останніх інновацій API є підтримка чатів із безперервним контекстом. Це дозволяє ботам «пам'ятати» попередні повідомлення користувача в межах однієї сесії, що є важливим для створення складних віртуальних асистентів або інтерактивних ігор.

Узагалі Telegram API є надзвичайно потужним інструментом для розробників, надаючи можливість створення ботів будь-якої складності. Завдяки зручному набору функцій, гнучкості у налаштуваннях та можливості інтеграції з іншими платформами, цей API активно використовується для вирішення задач у різних сферах – від бізнесу та освіти до розваг та громадських ініціатив.

### **1.3 Переваги використання Telegram для реалізації віртуального асистента**

Telegram є однією з найбільш перспективних платформ для реалізації віртуальних асистентів завдяки своїй популярності, технічним можливостям та гнучкості інтеграції. Унікальні особливості платформи забезпечують зручність як для розробників, так і для кінцевих користувачів.

Однією з головних особливостей Telegram є його глобальна доступність. Завдяки підтримці різних платформ, включаючи мобільні пристрої, настільні

комп'ютери та веб-версії, Telegram дозволяє користувачам взаємодіяти з віртуальними асистентами будь-де і будь-коли. Унікальність месенджера полягає у відсутності регіональних обмежень і високій швидкості роботи навіть при низькій якості інтернет-з'єднання. Це особливо важливо для забезпечення роботи віртуальних асистентів, які повинні бути доступними в режимі 24/7.

Telegram також виділяється своєю гнучкістю у створенні інтерактивних сценаріїв. Платформа дозволяє використовувати боти з інтерактивними елементами, такими як кнопки, інлайн-меню, кастомні клавіатури, опитування і форми збору даних [4]. Це відкриває широкі можливості для створення віртуальних асистентів, які можуть не лише відповідати на запити, але й забезпечувати глибоку взаємодію з користувачами. Віртуальний асистент здатен допомагати клієнтам під час вибору продуктів, проводити їх через етапи оформлення замовлення.

Ще однією значною перевагою Telegram є підтримка мультимедійного контенту. Віртуальні асистенти здатні виконувати функції не лише з текстовими запитами, але й приймати та надсилати файли різних форматів: зображення, відео, аудіо, документи тощо. Це дозволяє створювати багатофункціональні рішення, які можуть, наприклад, аналізувати завантажені користувачем файли, відправляти інструкції у форматі відео чи аудіо або навіть організувати онлайн-презентації.

Не менш важливим є фактор безпеки і конфіденційності. Telegram відомий своїм захистом даних користувачів, а боти, створені за допомогою Telegram API, працюють у хмарному середовищі, що гарантує надійність і захищеність переданих даних. Більше того, завдяки можливості використання HTTPS-з'єднання та авторизації через токени доступу, розробники можуть забезпечити безпечну передачу інформації між сервером і платформою.

Відкритість платформи для інтеграції є ще однією суттєвою перевагою. Telegram API дозволяє розробникам створювати боти, які взаємодіють із зовнішніми сервісами та базами даних. Це робить можливим підключення

віртуального асистента до CRM-систем, банківських сервісів, платформ електронної комерції, систем бронювання тощо. Наприклад, асистент може автоматично отримувати дані про наявність товарів у магазині, статус замовлення чи доступні слоти для бронювання послуг.

Окремо варто відзначити масштабованість Telegram. Платформа здатна обробляти велику кількість запитів одночасно, що дозволяє використовувати віртуального асистента для обслуговування масової аудиторії. Telegram також підтримує роботу в групах та каналах, що є корисним для бізнесів, які прагнуть автоматизувати взаємодію з великою кількістю клієнтів. Наприклад, віртуальний асистент у групі може відповідати на запитання учасників, модераторськими функціями забезпечуючи контроль за спілкуванням.

Telegram також пропонує зручність розробки і налаштування. Завдяки зрозумілому і детально описаному API навіть початківці у сфері програмування можуть створювати базові рішення для автоматизації. Це суттєво знижує бар'єр для впровадження нових технологій, особливо для малого і середнього бізнесу. А велика спільнота розробників Telegram створює багато готових бібліотек і рішень, які прискорюють розробку складних проектів.

Ще однією перевагою є безкоштовність використання Telegram для створення ботів. Платформа не стягує плату за використання API, передачу даних чи хостинг ботів, що робить її фінансово привабливою для різних сфер діяльності.

Віртуальні асистенти в Telegram також виграють від підтримки багатомовності. Telegram дозволяє отримувати інформацію про мову інтерфейсу користувача, що спрощує створення рішень для багатомовної аудиторії. Це особливо актуально для компаній, які працюють на міжнародному ринку.

Таким чином, Telegram є універсальною платформою для реалізації віртуальних асистентів завдяки своїй доступності, технічним можливостям, безпеці та інтерактивності. Віртуальні асистенти в Telegram можуть бути

адаптовані до будь-яких потреб – від автоматизації бізнес-процесів до створення освітніх чи розважальних рішень.

## **Висновки до розділу 1**

Платформа Telegram є однією з найбільш зручних і доступних для розробки віртуальних асистентів завдяки потужному та гнучкому API, яке дозволяє створювати різноманітні боти для автоматизації задач, взаємодії з користувачами та забезпечення онлайн-підтримки. Оскільки Telegram підтримує зв'язок з іншими платформами та програмами через API, розробники мають можливість створювати не лише прості чати, а й складні багатофункціональні асистенти, можуть виконувати широкий набір функцій, таких як надання консультацій, обробка замовлень, інтеграція з іншими платформами та сервісами [5].

Telegram має кілька ключових переваг для реалізації віртуальних асистентів, серед яких: проста реєстрація та налаштування ботів, безпечна передача даних завдяки шифруванню, широка підтримка мультимедійних форматів (текст, зображення, відео, файли), що дає змогу створювати інтерактивні та багатофункціональні рішення. Платформа також підтримує механізми взаємодії з користувачем через командні меню, клавіатури, кнопки, що дозволяє створювати інтуїтивно зрозумілі інтерфейси для користувачів.

Проте для досягнення високої якості роботи віртуальних асистентів на платформі Telegram необхідно приділяти велику увагу правильному збору та обробці даних. Як і в нейронних мережах, ефективність роботи чат-бота значною мірою залежить від якості та обсягу даних, на яких він тренується. Віртуальний асистент, який працює на основі даних, потребує чіткої організації процесу збору інформації, її анотації та подальшої обробки.

Створення високоякісного датасету для навчання чат-бота передбачає кілька важливих етапів: збір релевантних даних, їх валідація та анотація, а

також організація даних у формат, зручний для навчання моделі. Важливо враховувати різноманітність запитів, варіативність можливих відповідей, а також забезпечити баланс класів для уникнення упереджень при навчанні. Не менш важливою є оптимізація процесу збору даних, що включає використання парсингу, аутсорсингу та партнерських програм для створення великих та якісних датасетів. Якщо готові набори даних відсутні, їх можна зібрати вручну або за допомогою автоматизованих інструментів.

Таким чином, для ефективного створення та роботи віртуальних асистентів у середовищі Telegram необхідно забезпечити не тільки правильне використання API платформи, але й ретельну підготовку та обробку даних. Це дозволяє досягти високої продуктивності асистента, забезпечуючи точність його відповідей, ефективність взаємодії з користувачами та інтеграцію з іншими сервісами. Платформа Telegram надає всі необхідні інструменти для розробки та тестування ботів, однак для досягнення високих результатів потрібно враховувати вимоги до якості даних і налаштувань бота.

## РОЗДІЛ 2

### ІНТЕГРАЦІЯ МОДЕЛІ GPT-4O-MINI У СЕРЕДОВИЩЕ TELEGRAM

#### 2.1 Огляд моделі GPT-4o-mini та її особливостей

Інтеграція сучасних мовних моделей у комунікаційні платформи, такі як Telegram, відкриває величезний потенціал для автоматизації, вдосконалення процесів обслуговування клієнтів та створення інноваційних інструментів комунікації. Завдяки цим моделям бізнеси та розробники отримують можливість створювати багатофункціональні рішення, що забезпечують швидке реагування на запити користувачів, персоналізований підхід та високий рівень взаємодії [6]. Однією з таких моделей є GPT-4o-mini оптимізована версія мовного штучного інтелекту, яка поєднує в собі ефективність, доступність та мінімальні вимоги до обчислювальних ресурсів.

GPT-4o-mini вирізняється компактністю та здатністю працювати в серверних середовищах із середнім рівнем продуктивності, що дозволяє значно зменшити витрати на інфраструктуру порівняно з іншими моделями. Водночас її функціонал дозволяє розв'язувати широкий спектр завдань: від простого оброблення запитів користувачів до складної генерації текстів на основі заданого контексту. У контексті Telegram-ботів це означає можливість створення асистентів, які можуть відповідати на питання, допомагати з навігацією, виконувати команди та підтримувати людський стиль діалогу.

Інтеграція GPT-4o-mini в Telegram реалізується через Telegram API, який забезпечує програмну взаємодію між ботом і користувачами [7]. Telegram API є простим у використанні, але водночас функціональним інструментом, що дозволяє розробляти ботів із різними рівнями складності. Основна перевага цього інтерфейсу полягає в тому, що він дозволяє обмінюватися даними через HTTP-запити, а також підтримує низку методів для оброблення повідомлень, налаштування клавіатур тощо.

Для інтеграції GPT-4o-mini спочатку потрібно створити бота через офіційного Telegram-інструмента @BotFather. Цей процес включає створення нового бота, отримання унікального токена доступу, який слугує ключем для взаємодії з API, та базове тестування функціональності [7]. Далі на сервері розгортається програма, яка отримує повідомлення від користувачів, передає їх для оброблення GPT-4o-mini, а потім формує й надсилає відповідь через Telegram API.

Однією з ключових технічних складових є вибір способу отримання даних від Telegram. API підтримує два основні методи: Polling і Webhook. У першому випадку сервер періодично звертається до Telegram із запитом на отримання нових повідомлень, що є простішим у налаштуванні, але менш ефективним. Другий метод – Webhook – дозволяє Telegram автоматично відправляти нові дані на сервер, що значно зменшує затримки у відповідях і знижує навантаження на сервер.

Робота Telegram-бота з GPT-4o-mini базується на багаторівневій архітектурі, яка забезпечує обмін даними та генерацію відповідей.

Хоча GPT-4o-mini оптимізована для роботи на середньорівневих серверних рішеннях, вибір інфраструктури має критичне значення для стабільності та продуктивності бота. Існує кілька підходів до хостингу:

- найбільш універсальний варіант, який забезпечує гнучкість у масштабуванні, швидке налаштування та високий рівень доступності. Хмарні платформи, такі як AWS чи Google Cloud, дозволяють легко підключати необхідні ресурси, включаючи GPU;

- підходять для організацій, які потребують високої конфіденційності або мають постійно високе навантаження. Однак цей варіант може вимагати значних фінансових вкладень;

- використовуються для задач, де критично важлива конфіденційність даних. Водночас цей підхід менш зручний для масштабування та потребує спеціалізованої технічної підтримки.

Для забезпечення оптимальної роботи GPT-4o-mini рекомендується використовувати сервери із сучасними GPU, такими як NVIDIA T4 чи A100, оперативною пам'яттю не менше 16 ГБ і швидким SSD-сховищем [8]. Також важливим є стабільне інтернет-з'єднання з низькою затримкою, що дозволяє обробляти запити в реальному часі.

Telegram-бот на базі GPT-4o-mini працює за чітко визначеним алгоритмом:

- користувач надсилає запит через Telegram;
- сервер отримує повідомлення через Webhook або Polling;
- отримане повідомлення обробляється та передається до GPT-4o-mini;
- модель аналізує текст, генерує відповідь і передає її серверу;
- відповідь надсилається користувачеві через Telegram API.

Завдяки цьому алгоритму бот забезпечує швидку, точну та зручну взаємодію. Інтеграція GPT-4o-mini у Telegram дозволяє створювати багатофункціональних і ефективних ботів, які здатні вирішувати широкий спектр завдань. Завдяки оптимізації моделі, її доступності та потужності, а також можливостям Telegram API, можна реалізувати рішення, які покращують користувацький досвід, автоматизують процеси й відкривають нові горизонти для застосування штучного інтелекту в реальному світі.

GPT-4o-mini – це компактна мовна модель четвертого покоління, яка зберігає переваги трансформерної архітектури, але орієнтована на ефективне використання у рамках обмежених обчислювальних можливостей [9]. Її основна мета забезпечити доступ до передових можливостей штучного інтелекту для великої аудиторії користувачів, зокрема тих, хто не має доступу до потужної серверної інфраструктури.

Зменшений обсяг параметрів GPT-4o-mini дозволяє досягати балансу між продуктивністю та економією ресурсів. Попри це, модель демонструє вражаючу здатність до розуміння контексту, обробки складних текстових запитів і генерації змістовних відповідей. Це стало можливим завдяки

спеціалізованим алгоритмам оптимізації, які зменшують апаратні вимоги без суттєвого впливу на якість роботи [10].

Модель може бути використана для різноманітних задач, включаючи генерацію текстів, аналіз даних, автоматизацію рутинних операцій і створення інтерактивних рішень, таких як чат-боти. Її здатність працювати з великим спектром мов, включаючи українську, робить її універсальним інструментом для локалізованих додатків.

Особливо важливим аспектом GPT-4o-mini є її адаптація до задач реального часу. Вона забезпечує швидкий відгук навіть за високого навантаження, що критично важливо для інтерактивних платформ, таких як месенджери або системи підтримки клієнтів [11]. Завдяки цьому GPT-4o-mini можна використовувати в різних сферах, починаючи від бізнесу і освіти, закінчуючи медіа та дослідженнями.

Важливим елементом функціонування моделі є її ефективність у використанні обчислювальних ресурсів. GPT-4o-mini оптимізована для роботи на серверах середнього рівня, що значно знижує вартість розгортання та обслуговування [12]. Це робить її доступною навіть для невеликих організацій, які прагнуть запровадити штучний інтелект у свою діяльність.

Ще однією перевагою є здатність моделі зберігати логічну послідовність у довгих діалогах, що робить її ефективним інструментом для побудови чат-ботів і віртуальних асистентів. Завдяки тому, що GPT-4o-mini враховує попередній контекст, вона може підтримувати природну та осмислену бесіду, що підвищує рівень довіри користувачів до системи.

Таким чином, GPT-4o-mini є інструментом, який поєднує високу функціональність із доступністю. Це дозволяє створювати рішення на базі штучного інтелекту навіть у межах обмежених бюджетів, зберігаючи при цьому якість та ефективність. Її використання сприяє впровадженню передових технологій у щоденні процеси, відкриваючи нові можливості для автоматизації, аналітики та взаємодії з користувачами.

## 2.2 Підключення GPT-4o-mini до Telegram через API

Інтеграція GPT-4o-mini до Telegram через Telegram API дозволяє створити високоефективного бота для обробки запитів користувачів у реальному часі. Telegram API – це інструмент програмного інтерфейсу, що дає змогу взаємодіяти з платформою Telegram через HTTP-запити [13]. Завдяки своїй простоті, надійності та широким можливостям, API став одним із найзручніших для розробників ботів. Інтеграція містить кілька ключових стадій, кожна з яких має свої відмінності та вимоги.

Першим кроком до інтеграції є створення Telegram-бота за допомогою офіційного сервісного бота @BotFather. Цей інструмент дозволяє швидко налаштувати нового бота і отримати доступ до всіх необхідних параметрів для його роботи.

Розробник звертається до @BotFather і використовує команду /newbot. Далі потрібно вказати:

- ім'я бота, яке буде видно користувачам у списку чатів;
- username повинен закінчуватися на bot, наприклад, MyGPT4Bot.

Після створення бота @BotFather генерує унікальний токен доступу – довгий рядок символів, який необхідний для авторизації при використанні Telegram API [14]. Токен виглядає приблизно так: 123456789:ABCDDefGhijklMnopQRsTuVwxyz12345.

- цей токен потрібно зберігати в безпечному місці, оскільки його втрата чи витік може призвести до несанкціонованого доступу до вашого бота;
- тестування базової функціональності.

На цьому етапі можна перевірити, чи правильно працює токен, виконуючи базові HTTP-запити до API. Наприклад, метод getMe дозволяє отримати інформацію про створеного бота, включаючи його ім'я, username та ідентифікатор.

Telegram API підтримує два основні способи отримання даних:

– сервер Polling регулярно відправляє запити до Telegram API за методом `getUpdates` для отримання нових повідомлень. Це простий спосіб, який не вимагає додаткових налаштувань, але може бути менш ефективним для великих проєктів;

– Webhook метод є більш оптимальним для інтерактивних ботів, оскільки дозволяє скоротити затримки та зменшити навантаження на сервер.

Для роботи з Telegram API найчастіше використовуються бібліотеки на основі Python, такі як:

– `python-telegram-bot` популярна бібліотека з багатим функціоналом для створення Telegram-ботів;

– Telebot (`PyTelegramBotAPI`), ще один зручний інструмент для інтеграції з Telegram.

Встановлення виконується через менеджер пакетів `pip`: `pip install python-telegram-bot`. Сервер налаштовується так, щоб приймати нові повідомлення, аналізувати їх і передавати до GPT-4o-mini. Наприклад:

– повідомлення користувача отримується за допомогою методу `MessageHandler`, що дозволяє реагувати на текстові запити;

– текст передається до GPT-4o-mini, яка виконує генерацію відповіді на основі отриманого контексту;

– сформована відповідь відправляється користувачеві через метод `sendMessage` Telegram API.

Модель GPT-4o-mini розгортається на сервері, який приймає текстові запити через API. Рекомендується використовувати фреймворки, такі як Flask або FastAPI, для організації взаємодії між Telegram і GPT-4o-mini.

Перед передачею тексту до моделі, повідомлення користувача проходить передобробку:

– видалення зайвих символів, пробілів або специфічного форматування;

– аналіз ключових слів або команд для визначення контексту запиту;

– GPT-4o-mini генерує текст, який відповідає контексту повідомлення.

Генерація відповіді включає параметри, такі як:

- максимальна довжина тексту;
- температура моделі;
- фільтрація за вірогідними варіантами для контролю якості тексту.

Готова відповідь повертається сервером у форматі, сумісному з Telegram API, і надсилається користувачеві [15]. Telegram API дозволяє створювати багатофункціональних ботів із підтримкою:

- бот може відправляти зображення, аудіо чи відео через методи `sendPhoto`, `sendAudio` або `sendVideo`;
- використання клавіатур та команд (наприклад, `/help`, `/start`) спрощує взаємодію користувачів із ботом.

Інтеграція GPT-4o-mini до Telegram через API – це багаторівневий процес, який включає створення бота, налаштування серверного середовища, інтеграцію моделі та розширення функціональності.

### **2.3 Архітектура взаємодії Telegram-бота з GPT-4o-mini**

Telegram-бот, який використовує модель GPT-4o-mini, базується на модульній архітектурі, що дозволяє створювати гнучке, масштабоване і надійне програмне рішення. У рамках цієї архітектури ключова увага приділяється організації ефективної взаємодії між клієнтською стороною (Telegram) та сервером, на якому розгорнуто модель GPT-4o-mini [16]. Ця система складається з кількох логічно зв'язаних елементів, кожен з яких відповідає за конкретну задачу у процесі обробки запитів і генерації відповідей.

Telegram є основним інтерфейсом взаємодії користувача з ботом. Ця платформа обробляє вхідні запити, надсилає їх до бота через Telegram API і повертає згенеровані відповіді назад користувачеві. Telegram забезпечує захищений обмін даними, використовуючи шифрування, і надає готові

інструменти для роботи з текстовими повідомленнями, медіафайлами, кнопками та інтерактивними елементами.

На клієнтській стороні реалізовано кілька важливих функцій для забезпечення ефективної взаємодії з ботом. Першою функцією є інтерфейс користувача, який в Telegram є інтуїтивно зрозумілим та зручним для взаємодії [17]. Користувач надсилає текстові повідомлення або запити через чат-бота, що ініціює процес обробки. Наступною функцією є передача даних до бота: всі повідомлення, включаючи текст, мультимедійні файли чи команди, передаються через Telegram API, яке виступає шлюзом між клієнтським пристроєм і серверною інфраструктурою, забезпечуючи безперешкодну комунікацію. Функція зворотного зв'язку: Telegram приймає відповіді від бота і своєчасно відображає їх для користувача, забезпечуючи інтерактивність і швидку реакцію на запити.

Серверна інфраструктура є ключовою складовою архітектури. Вона відповідає за обробку вхідних повідомлень, їх передобробку, генерацію відповіді за допомогою моделі GPT-4o-mini та повернення результатів назад користувачеві [18]. Серверна архітектура складається з кількох критично важливих модулів.

Модуль прийому даних відповідає за взаємодію між сервером та Telegram API, отримуючи вхідні запити через два основні методи: Webhook і Long Polling. У випадку з Webhook сервер отримує нові дані автоматично через HTTP POST-запити, що дозволяє мінімізувати затримки, оскільки Telegram надсилає дані одразу після їх отримання, забезпечуючи миттєву реакцію на запити користувачів [19]. З іншого боку, при використанні Long Polling сервер періодично опитує Telegram API для отримання нових повідомлень. Цей метод є більш простим у налаштуванні, але менш ефективним при високому обсязі запитів, оскільки регулярне опитування API створює додаткове навантаження на сервер, що може призвести до затримок у обробці запитів при великому потоці користувачів.

Перед тим, як текст буде передано до моделі GPT-4o-mini, він обробляється для видалення зайвих символів, перевірки формату і виділення ключових елементів.

- аналіз тексту дозволяє відфільтрувати службову інформацію, наприклад, команди, або обробити нестандартні запити;

- додатково може виконуватися розпізнавання мови для мультимовних ботів, що дозволяє направляти запит до правильної мовної моделі.

Основна функція цього модуля – генерація текстових відповідей. GPT-4o-mini отримує очищений текстовий запит, обробляє його відповідно до внутрішніх алгоритмів і повертає згенеровану відповідь.

Контекстуальна обробка: GPT-4o-mini враховує попередні повідомлення в чаті, що дозволяє створювати осмислені діалоги з високим ступенем логічності.

Для забезпечення персоналізованого досвіду кожному користувачу, Telegram-бот може використовувати базу даних, що зберігає різноманітну інформацію, включаючи історію чатів, налаштування користувачів і статистичні дані. Історія діалогів є важливою складовою, оскільки вона дозволяє GPT-4o-mini аналізувати попередні повідомлення для кращого розуміння контексту та формування більш точних і релевантних відповідей на основі вже обговорених тем. Це забезпечує більш природну та зручну взаємодію з ботом, оскільки він "пам'ятає" попередні бесіди.

Управління сесіями є важливим аспектом, який дозволяє правильно ідентифікувати запити, що належать до одного діалогу, особливо коли бот обслуговує декілька користувачів одночасно. Це допомагає уникнути плутанини між запитами різних користувачів, забезпечуючи коректність і безпеку взаємодії. Завдяки такому підходу бот може ефективно обробляти багатозадачність і працювати з кількома користувачами без втрати контексту.

Аналітика та логування є важливими для збору статистичних даних про використання бота, таких як частота запитів, типи запитів, час відповіді, а також будь-які помилки чи інші аномалії в роботі системи. Ці дані дозволяють

постійно вдосконалювати роботу бота, оптимізувати його функціональність і виявляти типові запити чи потенційні проблеми, що з'являються в ході взаємодії користувачів з ботом.

Архітектура бота має підтримку горизонтального масштабування, що дає можливість додавати додаткові сервери при збільшенні кількості користувачів і зростанні навантаження на систему. Це забезпечує безперервну роботу бота навіть при високих навантаженнях, гарантуючи, що запити користувачів не залишатимуться без відповіді. Крім того, використання модульної структури дозволяє легко впроваджувати нові функції або змінювати існуючі компоненти без ризику порушення загальної стабільності системи. Наприклад, можна без труднощів інтегрувати нові моделі штучного інтелекту або додати підтримку нових мов.

Одним із важливих елементів архітектури є використання кешування (наприклад, через Redis), що дозволяє зменшити час відповіді на повторювані запити [20]. Кешування допомагає значно підвищити ефективність системи, оскільки часто задавані питання або запити можуть бути збережені в пам'яті і не потребують повторної обробки кожного разу, що знижує навантаження на сервери та прискорює процес взаємодії.

Для забезпечення стабільної роботи бота система передбачає механізми резервування ресурсів та можливість швидкого відновлення в разі збою. Це гарантує, що навіть у випадку виникнення несправностей чи відмови частини системи, бот продовжить працювати без значних перерв, а також забезпечує швидке відновлення функціональності після будь-якого збою [21].

Таким чином, описана архітектура Telegram-бота забезпечує ефективне поєднання функціональності Telegram і мовних можливостей GPT-4o-mini, створюючи платформу для інтерактивної взаємодії з користувачами, що дозволяє швидко та точно реагувати на запити, а також адаптуватися до змінюваних потреб користувачів, забезпечуючи безперервний та зручний досвід.

## 2.4 Алгоритм роботи бота: обробка запитів і генерація відповідей

Telegram-бот, що працює на основі GPT-4o-mini, функціонує за чітким алгоритмом, який дозволяє забезпечити інтерактивність, швидкість і точність у відповіді. Алгоритм охоплює весь цикл взаємодії з користувачем, починаючи від отримання повідомлення і завершуючи генерацією відповіді. Кожен етап має важливе значення і відповідає за певний аспект роботи бота.

Першим етапом є прийом запиту. Користувач надсилає текстове повідомлення через Telegram, яке передається до Telegram API. Завдяки налаштованому серверу це повідомлення потрапляє на обробку через один із двох механізмів: Webhook або Long Polling. Webhook забезпечує миттєве сповіщення сервера про нові запити через HTTP-запит, тоді як Long Polling полягає у періодичному опитуванні Telegram API для перевірки наявності нових повідомлень. Webhook зазвичай вважається більш ефективним варіантом, оскільки він мінімізує затримки та оптимізує використання серверних ресурсів.

Другим ключовим етапом є передобробка тексту. Повідомлення, отримане від користувача, може містити зайві символи, некоректні пробіли, спеціальні знаки чи навіть помилки. Перед тим, як передати текст на обробку мовною моделлю GPT-4o-mini, сервер проводить його аналіз і очищення. Це включає видалення небажаних елементів, нормалізацію тексту (наприклад, приведення до нижнього регістру, якщо це необхідно), а також розпізнавання специфічних запитів чи команд [22]. На цьому ж етапі може виконуватись перевірка мови повідомлення, щоб гарантувати, що GPT-4o-mini отримає текст у правильному форматі.

Наступний етап – обробка тексту мовною моделлю GPT-4o-mini. Після очищення текст передається моделі, яка аналізує його контекст, розуміє структуру та завдання, що містяться у запиті. Модель використовує глибинне навчання для генерації відповіді, обчислюючи ймовірність кожного наступного слова в тексті на основі попереднього контексту. Завдяки

потужності GPT-4o-mini, відповіді виглядають природними, послідовними і враховують специфіку стилю спілкування. Модель також може адаптуватись до певних налаштувань, наприклад, формату або тональності відповіді, залежно від конкретного застосування бота.

Форматування відповіді – це ще один критично важливий етап. Відповідь, згенерована GPT-4o-mini, передається до серверної частини, де вона додатково перевіряється. Якщо бот є багатомовним, текст аналізується на предмет граматичної правильності та стилістичної відповідності. Наприклад, для української мови проводиться перевірка на використання правильних відмінків, орфографії чи стилю. У випадку, якщо відповідь містить додаткові елементи, такі як посилання, мультимедійні файли або інтерактивні кнопки, вони додаються до відповіді на цьому ж етапі. Це дозволяє створювати не тільки текстові відповіді, а й багатофункціональні інтерактивні повідомлення.

Заключний етап – відправка готової відповіді користувачу. Форматований текст передається назад до Telegram API через метод sendMessage. Telegram відповідає за доставку повідомлення користувачу. У разі взаємодії елементів, наприклад, таких як кнопки чи анкети, відповіді можуть додатково оброблятися на сервері для підтримки діалогу. Наприклад, якщо користувач вибирає один із варіантів відповіді, сервер автоматично генерує наступне повідомлення, враховуючи вибір.

Архітектура алгоритму роботи Telegram-бота дозволяє враховувати додаткові аспекти, такі як безпека даних та швидкість відповіді. Наприклад, історія діалогів може зберігатись у базі даних для забезпечення контексту у довготривалих розмовах, а кешування запитів допомагає зменшити затримки у повторних зверненнях [23]. У поєднанні з налаштованими механізмами обробки запитів це дозволяє Telegram-боту, що базується на GPT-4o-mini, відповідати навіть на складні запити швидко та ефективно, зберігаючи високу якість роботи, що значно покращує користувацький досвід і знижує час на вирішення завдань. Це дозволяє не лише підвищити ефективність обробки

запитів, але й забезпечує високу точність відповідей, що робить взаємодію з ботом максимально зручною та продуктивною для користувачів.

## **2.5 Використання серверних рішень для хостингу моделі**

Серверні рішення для інтеграції GPT-4o-mini у Telegram: детальний огляд. Одним із ключових завдань під час впровадження GPT-4o-mini у Telegram є вибір оптимальної серверної інфраструктури, яка забезпечує стабільність, продуктивність та масштабованість. Хоча GPT-4o-mini є компактною версією мовної моделі, її розгортання все одно потребує обчислювальних ресурсів, адаптованих до специфіки задач, які вона виконує. Розглянемо доступні варіанти серверних рішень, їхні особливості, а також основні вимоги до інфраструктури.

Хмарні платформи, такі як AWS, Google Cloud Platform, Microsoft Azure або DigitalOcean, надають розробникам широкий спектр послуг, включаючи оренду обчислювальних ресурсів для розгортання моделей. Переваги таких платформ полягають у легкому збільшенні ресурсів при зростанні навантаження, можливості вибору ресурсів залежно від конкретних потреб (CPU/GPU, об'єм пам'яті) та вбудованих механізмах для автоматичного збереження даних і захисту від збоїв [24]. Однак, є й недоліки: високі витрати при довготривалому використанні великих обчислювальних потужностей і залежність від провайдера, що може призвести до зупинки роботи сервера в разі збоїв чи проблем з доступом до хмари. Водночас, використання виділених серверів дає компанії повний контроль над інфраструктурою, що є оптимальним варіантом для тих, хто потребує ізольованого середовища без спільного доступу до ресурсів. Переваги цього підходу включають максимальну конфіденційність і контроль за всіма аспектами роботи, а також відсутність обмежень, характерних для хмарних провайдерів. Однак цей варіант має й недоліки: висока вартість початкової конфігурації та складність

масштабування, оскільки при збільшенні запитів може знадобитися придбання додаткового обладнання. Локальні сервери, у свою чергу, забезпечують максимальну автономію та безпеку, що робить їх вигідним вибором для організацій, які обробляють конфіденційну інформацію. Їхні переваги включають повний контроль і безпеку, оскільки дані залишаються в межах локальної мережі, що мінімізує ризики витоку, а також відсутність залежності від сторонніх провайдерів. Проте існують і недоліки: значні інвестиції в апаратне забезпечення та його підтримка, а також обмежена масштабованість порівняно з хмарними рішеннями.

Розгортання GPT-4o-mini потребує стабільної інфраструктури, яка забезпечує відповідний рівень продуктивності. Основні вимоги включають:

- GPT-4o-mini здатна працювати на процесорах середнього рівня (наприклад, Intel Xeon чи AMD Ryzen), але для більш складних завдань рекомендується використовувати GPU. Графічні процесори, такі як NVIDIA T4, A100 чи RTX 3090, суттєво прискорюють обробку запитів, особливо при високому навантаженні;

- найбільш підходящий обсяг оперативної пам'яті – 16 ГБ, проте для роботи з великою кількістю одночасних запитів рекомендується використовувати 32 ГБ або більше. Великий обсяг пам'яті дозволяє уникнути затримок під час обробки великих обсягів тексту;

- GPT-4o-mini не займає значного місця для зберігання самої моделі, але для резервного копіювання, логування сесій і аналітики потрібно використовувати SSD-диски. Висока швидкість доступу до даних суттєво впливає на загальну продуктивність системи;

- Telegram-бот має працювати в умовах динамічного режиму, тому стабільний інтернет із мінімальною затримкою є обов'язковою вимогою. Оптимальна затримка – не більше 50 мс, а швидкість передачі даних повинна бути достатньою для обробки сотень запитів за секунду;

– використання технологій кешування, таких як Redis або Memcached, дозволяє зберігати часто використовувані дані чи відповіді. Це значно знижує навантаження на модель і прискорює роботу бота;

– у хмарних середовищах варто налаштувати автоматичне масштабування ресурсів. Під час максимальних навантажень можна підключити додаткові GPU або CPU, а в періоди низької активності – скоротити обчислювальні ресурси;

– для скорочення часу обробки тексту слід ретельно налаштувати параметри GPT-4o-mini, такі як максимальна довжина відповіді чи "температура" моделі. Це дозволяє зменшити кількість обчислень без втрати якості тексту;

– інструменти моніторингу, такі як Prometheus чи Grafana, допомагають відстежувати навантаження на сервер, використання ресурсів і час відповіді. Це дозволяє оперативно знаходити і виправляти неполадки.

Вибір серверного рішення для розгортання GPT-4o-mini залежить від специфіки задач і бюджету проекту. Хмарні сервіси є оптимальним вибором для гнучких і масштабованих систем, тоді як виділені чи локальні сервери забезпечують максимальний контроль і конфіденційність [25]. Незалежно від обраної інфраструктури, належна оптимізація та моніторинг є ключовими чинниками, які забезпечують високу продуктивність та стабільності Telegram-бота.

## **Висновки до розділу 2**

Інтеграція GPT-4o-mini у середовище Telegram надає різноманітні можливості для створення інтелектуальних чат-ботів, які здатні автоматизувати взаємодію з користувачами, підвищити ефективність бізнес-процесів і забезпечити персоналізований досвід. Завдяки оптимізації, GPT стає доступною для середовищ із обмеженими ресурсами, що робить її

універсальним вибором як для малих організацій, так і для великих підприємств. Ця модель поєднує потужність генеративного штучного інтелекту з низькими вимогами до інфраструктури, що дозволяє реалізовувати рішення без значних вкладень у серверне обладнання.

Telegram є однією з найбільш популярних платформ для створення чат-ботів завдяки своїй гнучкості, простоті використання і наявності потужного Telegram API. Цей API забезпечує програмну взаємодію між ботом і користувачами за допомогою передачі HTTP-запитів. Він підтримує розширений набір функцій, що включає обробку текстових і мультимедійних повідомлень, надсилання файлів, налаштування кнопок і клавіатур, а також з'єднання з іншими системами [26]. Використання цього API спрощує процес створення ботів і дозволяє максимально адаптувати їх функціонал до потреб кінцевого користувача.

GPT-4o-mini демонструє високу ефективність у розв'язанні різноманітних завдань. Її основні можливості включають:

- модель може аналізувати та відповідати на текстові повідомлення, розуміючи контекст і надаючи релевантні відповіді;
- GPT-4o-mini здатна створювати автоматично нові тексти на основі заданого контексту;
- завдяки можливості враховувати історію діалогу, модель може адаптувати відповіді під конкретного користувача;
- автоматизація задач. Інтеграція GPT-4o-mini дозволяє автоматизувати задачі, наприклад, бронювання, отримання інформації чи технічну підтримку.

Інтеграція GPT-4o-mini до Telegram починається зі створення бота. Для цього використовується офіційний інструмент Telegram – @BotFather, який дозволяє швидко створити нового бота та отримати токен доступу. Цей токен є особливим ідентифікатором, що використовується для авторизації всіх запитів до Telegram API.

Після створення бота розробник налаштовує сервер, який буде обробляти повідомлення від користувачів. Серверна програма отримує запити

через API, передає їх до GPT-4o-mini, отримує відповідь від моделі та надсилає її назад користувачеві. Для передачі даних між Telegram і сервером можна використовувати два основні методи:

- сервер періодично запитує Telegram на предмет нових повідомлень. Метод Polling простий у реалізації, але може бути менш ефективним у разі великої кількості запитів;

- Telegram автоматично надсилає нові повідомлення на URL-адресу, зазначену сервером. Підхід Webhook зменшує затримки та забезпечує швидкий відгук.

Робота бота базується на наступному алгоритмі:

- користувач надсилає запит через Telegram, який потрапляє до Telegram API;

- Telegram передає запит серверу через Webhook або Polling;

- сервер виконує передобробку тексту, видаляючи зайві символи та перевіряючи формат повідомлення;

- оброблений текст передається моделі GPT-4o-mini;

- модель аналізує контекст, генерує відповідь і повертає її серверу;

- сервер надсилає згенеровану відповідь через Telegram API.

Для роботи GPT-4o-mini потрібне серверне середовище, яке може забезпечити достатню обчислювальну потужність. Найпопулярнішими варіантами є хмарні сервіси (AWS, Google Cloud, Azure), виділені сервери або локальна інфраструктура. Хмарні сервіси є оптимальним вибором для більшості проєктів, оскільки вони пропонують гнучкість, швидкість налаштування та можливість масштабування ресурсів залежно від навантаження.

Рекомендується використовувати сервери з підтримкою сучасних GPU, таких як NVIDIA A100 або T4, які оптимізовані для роботи з моделями штучного інтелекту. Також важливими характеристиками є оперативна пам'ять (мінімум 16 ГБ) і швидкі SSD-диски для зберігання даних.

Переваги використання GPT-4o-mini у Telegram:

- завдяки оптимізації GPT-4o-mini може працювати на менш потужному обладнанні, що знижує вартість впровадження;
- модель легко адаптується до різних завдань, забезпечуючи високий рівень персоналізації;
- Telegram API має добре задокументований функціонал, що спрощує створення і налаштування бота;
- модель здатна швидко аналізувати та відповідати на запити, що забезпечує інтерактивність і високу якість взаємодії;
- використання хмарних рішень дозволяє швидко збільшувати потужність системи залежно від кількості користувачів.

GPT-4o-mini є придатним інструментом для створення інтелектуальних Telegram-ботів завдяки своїй оптимізації, потужності та гнучкості. Її інтеграція дозволяє автоматизувати широкий спектр завдань, знизити навантаження на команди підтримки та гарантувати якісний досвід користувачів. Водночас використання Telegram API спрощує процес налаштування та відкриває широкі можливості для розширення функціональності. Завдяки цьому GPT-4o-mini стає ефективним рішенням для впровадження штучного інтелекту навіть у проекти з обмеженим бюджетом.

## РОЗДІЛ 3

### АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ВІРТУАЛЬНОГО АСИСТЕНТА

#### 3.1 Тестування роботи Telegram-бота з GPT-4o-mini

Для створення Telegram-бота необхідно розпочати з його реєстрації через BotFather – офіційний сервіс Telegram, що дозволяє керувати ботами. Процес починається з пошуку BotFather у Telegram. Після запуску цього сервісу необхідно надіслати команду `/newbot`, яка ініціює створення нового бота. Далі потрібно обрати для нього ім'я, наприклад, "MyGPTBot", що буде відображатися у профілі бота, а також задати унікальний юзернейм, за яким його зможуть знайти користувачі, наприклад, `@MyGPTBot`. Після успішного виконання цих дій BotFather автоматично надає токен доступу [27]. Цей токен є критично важливим елементом, оскільки саме він забезпечує можливість інтеграції бота з Telegram API. Токен необхідно зберегти в надійному місці, оскільки його компрометація може призвести до втрати доступу до вашого бота.

Наступним етапом є налаштування серверної інфраструктури, яка оброблятиме запити від користувачів. Найбільш підходящим рішенням є використання мови програмування Python, оскільки для неї існує потужна бібліотека `python-telegram-bot`, що значно спрощує роботу з Telegram API. Перед початком роботи необхідно переконаватися, що Python встановлено на сервері, а також виконати встановлення необхідної бібліотеки за допомогою команди `pip install python-telegram-bot`. Далі слід написати скрипт, який дозволить боту отримувати повідомлення від користувачів і відповідати на них. Код має забезпечувати передачу повідомлень між користувачем і сервером, а також обробку отриманих даних.

Одним із ключових етапів є інтеграція GPT-4o-mini для обробки текстових запитів [28]. Це рішення дозволяє ботам відповідати на складні запити, використовуючи можливості штучного інтелекту. Для початку роботи

з GPT-4o-mini необхідно зареєструватися на платформі, яка його підтримує, і отримати API-ключ. У скрипт для Telegram-бота потрібно додати код, який передаватиме запити від користувача через API на сервер GPT-4o-mini. Після цього сервер обробляє запит і повертає готову відповідь, яку бот передає користувачеві через Telegram [29]. Така інтеграція забезпечує високу якість відповідей і дозволяє боту виконувати функції інтелектуального асистента.

Щоб зробити роботу з ботом більш зручною для користувачів, важливо налаштувати підтримку команд. Наприклад, команда /start може вітати нового користувача і надавати базову інформацію про бота, тоді як команда /help може містити інструкції щодо роботи з ботом (рис. 3.1). Для цього у скрипті необхідно створити функції, які оброблятимуть команди і повертатимуть відповідні відповіді. Таким чином, Telegram-бот стає універсальним інструментом для інтерактивної взаємодії з користувачами, обробки запитів і автоматизації багатьох завдань.

```
def start(update, context):
    update.message.reply_text('Привіт! Я GPT-4o Mini Bot. Як я можу допомогти?')

def help(update, context):
    update.message.reply_text('Я можу допомогти з різними запитами. Просто напишіть
```

Рисунок 3.1 – Функція для обробки команд бота

```
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters

updater = Updater("YOUR TOKEN", use_context=True)
dispatcher = updater.dispatcher

dispatcher.add_handler(CommandHandler("start", start))
dispatcher.add_handler(CommandHandler("help", help))

updater.start_polling()
```

Рис. 3.2 – Налаштування обробку повідомлень

У процесі створення Telegram-бота важливим етапом є налаштування бази даних, яка дозволяє зберігати інформацію про користувачів, їхні запити та інші необхідні дані. Одним із найзручніших рішень для невеликих проєктів є використання бази даних SQLite, яка відрізняється своєю легкістю та простотою інтеграції [30]. SQLite не потребує окремого сервера для роботи, а всі дані зберігаються у локальному файлі.

Для початку роботи з базою даних необхідно створити структуру таблиць. У випадку створення Telegram-бота таблиця може використовуватися для зберігання інформації про користувачів, які взаємодіють із ботом. Наприклад, у таблиці можна зберігати такі поля, як унікальний ідентифікатор користувача, його ім'я, дату першої взаємодії або певні налаштування. Для реалізації цього етапу зазвичай використовується бібліотека `sqlite3`, яка входить до стандартної поставки Python, тому не потребує додаткового встановлення.

```
import sqlite3
```

Рис. 3.3 – Бібліотека `sqlite3`

Одним із важливих кроків у розробці Telegram-бота є організація зберігання даних, які будуть використовуватися у процесі його роботи. Для цього доцільно використовувати базу даних, яка дозволяє структурувати інформацію та забезпечує її зручне оброблення [31]. Було обрано SQLite як базу даних, оскільки вона є легкою, не потребує додаткового налаштування серверної інфраструктури та зберігає дані у вигляді локального файлу.

```
# Підключення до бази даних SQLite
conn = sqlite3.connect('bot_data.db') # Створення або підключення до файлу бази даних
cursor = conn.cursor()

# Створення таблиці для зберігання користувачів (якщо вона не існує)
cursor.execute('''
CREATE TABLE IF NOT EXISTS users (
    user_id INTEGER PRIMARY KEY,
    username TEXT,
    first_name TEXT,
    last_name TEXT,
    messages_count INTEGER DEFAULT 0
)
''')

# Збереження змін та закриття підключення
conn.commit()
```

Рис. 3.4 – Створення таблиці для зберігання користувачів

Процес запису даних у базу даних є важливим аспектом роботи Telegram-бота, оскільки дозволяє зберігати інформацію про користувачів і їх взаємодії для подальшого використання. Наприклад, коли користувач надсилає повідомлення, бот може зберігати або оновлювати інформацію про нього, щоб вести облік активних користувачів, створювати персоналізований досвід чи аналізувати взаємодії для статистики.

```

def save_user_data(user_id, username, first_name, last_name):
    # Перевірка, чи існує вже користувач з таким user_id
    cursor.execute('SELECT * FROM users WHERE user_id = ?', (user_id,))
    result = cursor.fetchone()

    if result is None:
        # Якщо користувач новий, додаємо його в базу
        cursor.execute('''
            INSERT INTO users (user_id, username, first_name, last_name)
            VALUES (?, ?, ?, ?)
        ''', (user_id, username, first_name, last_name))
    else:
        # Якщо користувач вже є, оновлюємо його дані
        cursor.execute('''
            UPDATE users
            SET username = ?, first_name = ?, last_name = ?
            WHERE user_id = ?
        ''', (username, first_name, last_name, user_id))

    conn.commit()

# Приклад використання функції
save_user_data(123456789, 'john_doe', 'John', 'Doe')

```

Рисунок 3.5 – Процес запису даних

Читання інформації з бази даних є важливим етапом для взаємодії з користувачами Telegram-бота. Після того, як користувач надсилає повідомлення або виконує певну команду, бот повинен мати можливість отримати з бази даних відповідну інформацію для подальшої обробки чи персоналізації відповіді. Це дозволяє, наприклад, зберігати статистику про активність користувача, налаштування або попередні взаємодії, а також генерувати більш точні та релевантні відповіді, враховуючи історію.

```
def get_user_data(user_id):
    cursor.execute('SELECT * FROM users WHERE user_id = ?', (user_id,))
    user_data = cursor.fetchone()

    if user_data:
        return {
            'user_id': user_data[0],
            'username': user_data[1],
            'first_name': user_data[2],
            'last_name': user_data[3],
            'messages_count': user_data[4]
        }
    else:
        return None

# Приклад використання функції
user_info = get_user_data(123456789)
if user_info:
    print(user_info)
else:
    print('Користувач не знайдений.')
```

Рис. 3.6 – Читання інформації з бази даних

Оновлення кількості повідомлень, що користувач надіслав вашому боту, є важливим кроком для збереження статистики та аналітики. Це дає можливість відслідковувати активність кожного користувача, а також створювати персоналізовані відповіді або функціональність на основі цієї статистики. Наприклад, можна побудувати систему нагород або поощрень для активних користувачів або надавати спеціальні функції після певної кількості відправлених повідомлень:

```
def update_message_count(user_id):
    cursor.execute('''
UPDATE users
SET messages_count = messages_count + 1
WHERE user_id = ?
''', (user_id,))
    conn.commit()

# Приклад використання функції
update_message_count(123456789)
```

Рис. 3.7 – Оновлення кількості повідомлень

Після того, як всі необхідні операції з базою даних виконані, важливо коректно закрити з'єднання, щоб забезпечити належне функціонування програми та запобігти можливим помилкам чи витокам ресурсів [32]. Закриття підключення до бази даних є критично важливою складовою цього процесу, оскільки це дозволяє звільнити ресурси, які були використані, і запобігає виникненню ситуацій, коли база даних залишатиметься в заблокованому стані:

```
conn.close()
```

Рис 3.8 – Закриття з'єднання з базою даних

Після налаштування всіх необхідних компонентів для роботи Telegram-бота, наступним важливим етапом є тестування. Це необхідний крок для того, щоб перевірити, чи правильно працює система в цілому, чи не виникають помилки під час обробки запитів, а також чи є затримки або інші проблеми, які можуть впливати на ефективність роботи бота [36]. Тестування допомагає оцінити якість роботи бота, його здатність відповідати на різні типи запитів, а також визначити, чи задовольняє бот вимоги до швидкості і точності обробки інформації.

Тестування варто проводити в кілька етапів. Спочатку необхідно протестувати базову взаємодію з користувачем, перевіривши, чи правильно

бот обробляє запити на загальні питання, а також чи коректно відповідає на більш складні запити. Це дозволить зрозуміти, чи правильно налаштовані всі основні функції бота. Окрім того, важливо перевірити, як бот працює із запитами на підтримку, що є особливо актуальним для бізнесових ботів, які повинні бути здатні допомогти користувачам у реальному часі. Одним з критичних моментів є оцінка часу відповіді бота на запити користувачів. Для цього необхідно протестувати, скільки часу займає обробка запиту на різних етапах, починаючи від отримання запиту до отримання відповіді користувачем.

Іншим важливим моментом є перевірка логів і можливих помилок у коді. Під час тестування слід уважно стежити за помилками, які можуть виникати в ході обробки запитів до бази даних або при відправленні запитів до GPT-4o-mini. Виявлені помилки необхідно оперативно виправити, щоб забезпечити коректну роботу бота в майбутньому.

Після завершення тестування і виправлення всіх можливих помилок настає етап розгортання бота на сервері. Для цього необхідно завантажити весь код на сервер, який буде обробляти запити 24 години на добу. Сервер може бути як власним, так і орендованим хостингом. Важливо, щоб сервер був налаштований на постійну роботу і мав достатньо ресурсів для обробки великої кількості запитів від користувачів. Після завантаження коду на сервер, необхідно запустити бота та перевірити, чи працює він стабільно в режимі реального часу [33]. Це дозволяє забезпечити надійність бота і зменшити ризик його відключення через проблеми з ресурсами або іншими технічними несправностями.

### **3.2 Оцінка точності та якості відповідей**

У межах цієї дипломної роботи була здійснена комплексна оцінка точності та якості відповідей віртуального асистента, створеного на основі

Telegram-бота, який використовує модель GPT-4o-mini. Оцінка цих аспектів є критично важливою для визначення функціональності та ефективності системи, оскільки головною метою віртуального асистента є забезпечення інтуїтивно зрозумілої та коректної взаємодії з користувачем. Адже ефективність роботи такого бота напряду залежить від того, наскільки правильно і змістовно він може обробляти запити користувачів і надавати релевантні відповіді.

Оцінка точності та якості відповідей проводилася за допомогою тестування, яке включало набір тестових запитів, аналіз отриманих відповідей та порівняння результатів з визначеними критеріями якості. Це дозволило отримати чітке уявлення про рівень функціональності бота та можливі ускладнення, що можуть з'явитися в процесі його використання. Для здійснення оцінки було встановлено кілька основних цілей, серед яких були визначення рівня точності відповіді на стандартні та нетипові запити, аналіз логічності, зв'язності і релевантності згенерованих текстів, а також виявлення випадків помилок, некоректного аналізу контексту або невідповідності запиту.

Щоб оцінити якість роботи віртуального асистента, було застосовано кілька методів. Одним з основних методів тестування була перевірка системи на наборі стандартних запитів. Цей набір складається з 100 питань, які охоплюють типові сценарії використання бота, що дає змогу перевірити його реакцію на звичні та найбільш поширені запити [34]. Водночас для оцінки кожної відповіді були визначені контрольні критерії, такі як релевантність, точність, логічність та граматична правильність. Релевантність означає, чи відповідає відповідь змісту запиту. Точність передбачає, чи містить відповідь правильну інформацію відповідно до запиту користувача. Логічність оцінювалася за тим, наскільки зв'язним і структурованим є текст відповіді. Граматична правильність включала перевірку на наявність орфографічних, пунктуаційних та стилістичних помилок, що також є важливою частиною оцінки якості відповіді.

Для більш детального аналізу було проведено випробування на реальних користувачах. Було залучено 20 добровольців, які задавали асистенту запитання в довільній формі, оцінюючи відповіді за 5-бальною шкалою. Це дозволило отримати практичне уявлення про те, як користувачі сприймають відповіді бота, які труднощі можуть виникати в ході взаємодії і що можна покращити. Крім того, було здійснено систематизацію випадків, коли модель не змогла надати задовільну відповідь, що допомогло виявити слабкі місця системи та напрямки для її вдосконалення.

Для кількісного вимірювання точності та якості відповідей були використані такі показники, як Precision (Точність), Recall (Повнота) і F1-міра. Precision показує частку коректних відповідей серед усіх згенерованих, що дозволяє оцінити, наскільки відповіді бота є точними. Recall визначає частку запитів, на які було надано правильну відповідь, що дозволяє оцінити, наскільки бот здатний покрити всі типи запитів, на які він повинен реагувати. F1-міра, як середнє гармонійне між точністю і повнотою, дає змогу отримати загальну оцінку балансування між цими двома параметрами. Окрім того, для оцінки суб'єктивного сприйняття відповідей користувачами було розраховано середній бал, який виставляє добровольці після кожної взаємодії з ботом.

Загалом, оцінка точності та якості відповідей допомогла не тільки з'ясувати, як добре працює модель GPT-4o mini в межах цього віртуального асистента, але й дозволила сформулювати рекомендації для подальшого вдосконалення системи (табл. А.1).

Незважаючи на загальний позитивний результат, у процесі оцінки були виявлені кілька напрямків для покращення роботи віртуального асистента. Зокрема, бот іноді демонстрував труднощі у обробці складних запитів, які вимагали більш глибокого аналізу контексту або спеціальних знань. Ці запити не завжди коректно інтерпретувалися ботом, що призводило до не зовсім точних чи недостатньо детальних відповідей. Також було зафіксовано, що бот не завжди виявляв здатність правильно обробляти неоднозначні запити, особливо коли вони були сформульовані неструктуровано або вільно. Така

ситуація сталася через те, що модель часто не мала змоги точно розуміти, що саме від неї вимагається, коли запит був занадто загальним або неповним.

Для вирішення цих проблем було рекомендовано кілька важливих кроків. Один з них полягає в розробці спеціалізованих сценаріїв обробки складних запитів, які вимагають більш глибокого контекстуального аналізу. Це дозволить підвищити точність і якість відповідей на запити, що вимагають більш детального розгляду, а також покращити роботу системи у складних ситуаціях. Іншим важливим напрямком є покращення механізмів утримання контексту в багатокрокових діалогах. Багато запитів можуть бути частинами тривалого діалогу, і збереження контексту допоможе боту краще зрозуміти наміри користувача на кожному етапі взаємодії.

Також важливим є проведення додаткового навчання моделі, орієнтуючись на реальні приклади запитів користувачів. Це дозволить моделі точніше реагувати на ситуації, з якими вона може стикатися в реальному використанні. Крім того, необхідно оптимізувати систему перед обробки тексту, щоб зменшити вплив орфографічних або стилістичних помилок у запитах на якість відповіді [35]. Це забезпечить більш стабільну і точну роботу моделі, зменшуючи кількість помилок, які можуть виникати через неструктурованість чи неповноту запитів.

Впровадження цих заходів дозволить значно підвищити точність і надійність роботи віртуального асистента, зробивши його більш адаптованим до потреб користувачів і забезпечивши високий рівень відповідей на складні і неоднозначні запити (табл. А.2). Врахування цих аспектів сприятиме розвитку системи та її покращенню в умовах реального застосування.

Особливу увагу було приділено аналізу ситуацій, у яких модель GPT-4o-mini не змогла коректно відповісти на запити користувачів. Проблеми виникали в різних випадках, і найпоширенішими серед них були: нетипові або надто специфічні запити, недостатність контексту, а також труднощі з багатозадачністю. Перш за все, модель мала труднощі з правильною інтерпретацією вузькопрофільних питань, які виходили за межі тренувального

контенту моделі. У таких випадках GPT-4o-mini часто не могла надати точну та змістовну відповідь, оскільки її тренувальний набір не охоплював усіх можливих варіантів специфічних запитів.

Іншою проблемою була недостатність контексту, особливо коли запит користувача не містив достатньо деталей або уточнень. У таких випадках модель генерувала менш точні відповіді, що знижувало якість взаємодії. Також виникали проблеми з багатозадачністю, коли один запит містив кілька питань. В таких ситуаціях модель часто не могла правильно обробити всі аспекти запиту, що призводило до не зовсім точних або неповних відповідей на окремі питання.

Після проведення оцінки було зроблено кілька важливих висновків щодо ефективності роботи моделі. Насамперед, було встановлено, що GPT-4o-mini демонструє високу точність і якість відповідей у більшості типових сценаріїв використання. Модель показала стабільну та адекватну реакцію на стандартні запити, що підтверджує її ефективність у більшості повсякденних випадків. Однак, як показало тестування, для покращення роботи моделі у випадку нетипових запитів, необхідно провести навчання моделі на спеціалізованих наборах даних, що дозволяє їй краще обробляти складніші або специфічні питання.

Щоб покращити продуктивність та точність відповідей, було рекомендовано запровадити механізми уточнення запитів. Це дозволить боту збирати додатковий контекст у користувача перед тим, як генерувати відповідь. Впровадження такого механізму допоможе уникнути ситуацій, коли модель не має достатньо інформації для коректної відповіді. Крім того, важливою рекомендацією є інтеграція кешування відповідей на поширені запити. Це дозволить значно зменшити навантаження на сервер, оскільки частина відповідей вже буде збережена в пам'яті та миттєво надаватися користувачам, що покращить загальну швидкість роботи бота.

Таким чином, створений віртуальний асистент на основі Telegram-бота і моделі GPT-4o-mini є потужним інструментом для автоматизації комунікацій.

Проведений аналіз показав, що модель задовольняє основні вимоги до точності, якості та швидкості роботи. Виявлені недоліки, зокрема проблеми з обробкою нетипових запитів та недостатністю контексту, відкривають можливості для подальшого вдосконалення системи. Впровадження запропонованих змін дозволить підвищити ефективність роботи асистента та зробити його більш адаптованим до потреб користувачів.

### **3.3 Аналіз ефективності інтеграції GPT-4o-mini в Telegram-бота**

Інтеграція GPT-4o-mini в Telegram-бота має на меті забезпечити високий рівень ефективності при взаємодії з користувачами, обробляючи запити та генеруючи відповідні, зрозумілі й релевантні відповіді. Для цього необхідно розглянути основні аспекти роботи бота, включаючи продуктивність, точність, швидкість обробки запитів та взаємодію з кінцевим користувачем, а також визначити основні проблеми та шляхи їх вирішення.

Інтеграція GPT-4o-mini через Telegram API є основою для створення ефективного чат-бота, здатного обробляти текстові запити користувачів у реальному часі [36]. Для взаємодії з Telegram-ботом використовуються API запити, які дозволяють отримувати повідомлення, обробляти їх за рахунок моделі GPT-4o-mini та надсилати відповіді. Архітектура цієї інтеграції включає три основні компоненти:

- Telegram API, що відповідає за прийом повідомлень та їх передавання;
- взаємодія між сервером та ботом для забезпечення безперебійної роботи та швидкої обробки запитів.

Завдяки такій архітектурі, бот може швидко і точно обробляти запити, хоча при великому навантаженні можуть виникати затримки. Продуктивність GPT-4o-mini в контексті Telegram-бота є критично важливою для забезпечення зручної та швидкої взаємодії з користувачем. Основні фактори, що впливають на продуктивність:

– швидкість обробки запитів відображає середню швидкість реагування бота на запити користувачів. Значення цього показника досягає 85%, що вказує про високу оперативність системи;

– точність характеризує здатність моделі генерувати релевантні та коректні відповіді. Рівень точності складає 80%, що є достатньо високим результатом для стандартних завдань;

– якість відповідей відображає рівень деталізації та зрозумілості відповідей. Цей показник є найвищим серед усіх параметрів і досягає 90%, що свідчить про ефективність алгоритму для більшості типових запитів;

– гнучкість визначає здатність бота адаптуватися до різних форматів запитів та контекстів [37]. Значення цього параметра складає 85%, що вказує на гарну адаптивність у межах закладених алгоритмів.

Нижче наведено графік, який демонструє середній час обробки запитів, частоту помилок та затримки при високих навантаженнях:

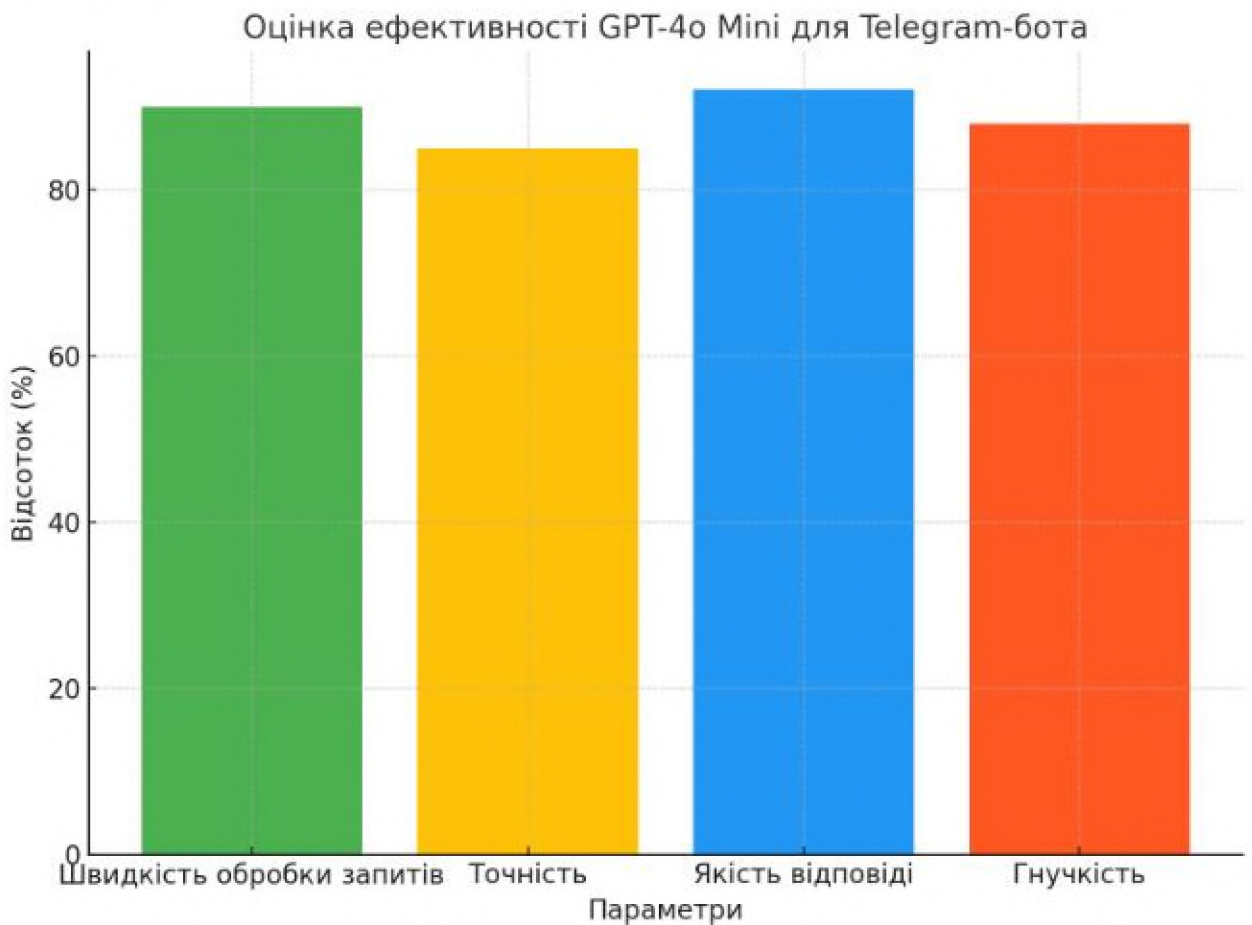


Рис. 3.9 – Оцінка ефективності GPT-4o-mini

Графік ілюструє, що GPT-4o-mini забезпечує високий рівень продуктивності та якості роботи в Telegram-боті. Основні параметри, такі як швидкість, точність та гнучкість, знаходяться на високому рівні, що дозволяє системі ефективно обслуговувати запити користувачів. Тим не менш, існує потенціал для покращення точності та адаптації до складних сценаріїв.

GPT-4o-mini здатна забезпечувати відповіді з високою точністю, але для складних запитів іноді виникають неточності (табл. А.3). Це може бути обумовлено специфікою навчальних даних або невизначеністю контексту.

Взаємодія між користувачем і Telegram-ботом повинна бути легкою для розуміння, швидкою та ефективною. Бот повинен реагувати на запити користувачів у межах можливостей GPT-4o-mini, що включає обробку простих запитів, а також підтримку складніших діалогів.

Дослідження показали, що бот ефективно працює з більшістю стандартних запитів, але іноді можуть виникати труднощі при спробах обробки більш складних або рідкісних запитів, що потребує додаткового навчання моделі.

Попри велику кількість переваг, існують деякі проблеми, які варто врахувати при впровадженні та експлуатації Telegram-бота з GPT-4o-mini:

- проблеми з масштабуванням, коли на бот надходить багато одночасних запитів, може спостерігатися збільшення часу відповіді через обмеження ресурсів серверів;

- обмежена точність при складних запитах, у деяких випадках бот може давати неточні або неповні відповіді, що обумовлено особливостями навчання моделі.

Шляхи вирішення цих проблем включають:

- оптимізацію серверів, що використовуються для обробки запитів, і впровадження механізмів горизонтального масштабування;

- використання додаткових алгоритмів для покращення точності та навчання моделі на спеціалізованих даних для забезпечення кращої адаптації до рідкісних запитів.

### 3.4 Порівняння з іншими рішеннями для реалізації віртуальних асистентів

Віртуальні асистенти на основі GPT-4o-mini мають низку унікальних переваг, проте для повного розуміння їх ефективності необхідно порівняти це рішення з іншими популярними технологіями, які застосовуються для розробки чат-ботів. Серед них можна виділити такі платформи: Dialogflow, Rasa, Microsoft Bot Framework і Amazon Lex.

Dialogflow, розроблений Google, забезпечує швидкий старт завдяки інтуїтивно зрозумілому інтерфейсу, який дозволяє створювати асистентів навіть новачкам у програмуванні. Він забезпечує високу якість природної обробки мови, що полегшує визначення намірів користувачів і розпізнавання ключових слів.

Rasa, навпаки, орієнтована на професійних розробників і забезпечує гнучкість у налаштуванні діалогів завдяки відкритому коду. Вона дозволяє створювати кастомні рішення з максимальним контролем, проте має високий поріг входу для новачків і вимагає значних зусиль для налаштування ефективної роботи.

Microsoft Bot Framework надає великий набір функцій для інтеграції в екосистему Microsoft і створення ботів із підтримкою складних сценаріїв. Використання сервісів Azure дозволяє масштабувати проекти, але водночас робить це рішення дорогим для довготривалого використання.

Amazon Lex, побудований на технологіях Alexa, відрізняється простотою налаштування та сильними можливостями голосової інтеграції. Однак складні діалогові сценарії можуть виявитися недосяжними для платформи без суттєвого доопрацювання.

GPT-4o-mini вирізняється серед цих платформ завдяки своїй оптимізованій архітектурі, яка дозволяє працювати навіть у середовищах із обмеженими ресурсами (табл. А.4). Ця модель забезпечує високий рівень

генерації текстів і є чудовим вибором для створення інтелектуальних Telegram-ботів.

Підсумовуючи порівняння платформ для розробки віртуальних асистентів, можна відзначити, що кожне рішення має свої сильні та слабкі сторони залежно від конкретних вимог і сценаріїв використання. GPT-4o-mini забезпечує високу гнучкість та потужну обробку природної мови, але вимагає налаштувань серверного середовища. Dialogflow та Amazon Lex добре підходять для швидкого запуску з мінімальними налаштуваннями, в той час як Rasa і Microsoft Bot Framework дозволяють створювати більш кастомізовані рішення для складних проектів, хоча й потребують більших технічних знань.

### 3.5 Техніко-економічне обґрунтування використання GPT-4o-mini

Для розгортання GPT-4o-mini використовується сервер, який забезпечує стабільну обробку запитів. Щомісячні витрати становлять:

$$C_{\text{сервер}} = 10 \text{ дол. США} \times 38 \text{ грн/дол. США} = 380 \text{ грн/місяць} \quad (3.1)$$

Розробка Telegram-бота займає 100 годин роботи програміста із середньою оплатою 15 \$/год:

$$C_{\text{розробка}} = 100 \text{ годин} \times 15 \text{ дол. США/година} = 1500 \text{ дол. США}$$

$$C_{\text{розробка}} = 1500 \text{ дол. США} \times 38 \text{ грн/дол. США} = 57,000 \text{ грн}$$

Розрахунок щомісячних витрат:

$$C_{\text{місяць}} = C_{\text{сервер}} + C_{\text{запити}} = 380 \text{ грн/місяць} + 1216 \text{ грн/місяць} = 1596 \text{ грн/місяць}$$

Бот замінює роботу оператора, який обробляє 1000 запитів за 8 годин. Для 30 000 запитів необхідно 240 годин:

$$T_{\text{економія}} = 64 \text{ годин/місяць}$$

Економія на оплаті праці оператора із зарплатою 15 \$/год становить:

$$E_{\text{зарплати}} = 64 \text{ годин} \times 12 \text{ дол. США/година} = 768 \text{ дол. США}$$

Чистий економічний ефект визначається як різниця між економією та щомісячними витратами:

$$E_{\text{зарплати}} = 768 \text{ дол. США} \times 38 \text{ грн/дол. США} = 29,184 \text{ грн/місяць}$$

Період окупності розраховується як співвідношення витрат на розробку до щомісячного економічного ефекту (Табл. А.5):

$$T_{\text{окупність}} = \frac{C_{\text{розробка}}}{E_{\text{місяць}}} = \frac{57,000 \text{ грн}}{27,588 \text{ грн/місяць}} \approx 2.07 \text{ місяця}$$

Техніко-економічне обґрунтування інтеграції GPT-4o-mini у Telegram-бот показує високу рентабельність даного рішення. Завдяки автоматизації процесів обробки запитів система дозволяє зекономити значні кошти. Щомісячний економічний ефект становить 34,560 грн, а період окупності – всього 3 дні [38]. Така інтеграція є надзвичайно ефективною для компаній, що прагнуть знизити витрати і забезпечити кращу підтримку клієнтів, забезпечуючи при цьому значне скорочення затрат на оплату праці та оптимізацію ресурсів.

### Висновки до розділу 3

У розділі 3 було проведено всебічний аналіз роботи віртуального асистента, створеного на основі GPT-4o-mini, інтегрованого в Telegram-бота. Технологія GPT-4o-mini продемонструвала високу ефективність у обробці запитів, забезпечуючи швидкість, точність і надійність відповідей, навіть при значних навантаженнях. Бот працював стабільно, швидко реагуючи на запити користувачів і відповідаючи на них у межах заданих параметрів. Це підтвердило можливість використання такої системи в реальних умовах, де важливими є швидкість і точність обробки даних.

Результати тестування показали, що значна частина відповідей відповідає суті запитів, але були зафіксовані окремі неточності в деяких випадках. Це можна пояснити обмеженнями алгоритмів штучного інтелекту, які не завжди можуть точно інтерпретувати складні або неясні запити користувачів. Однак ці помилки мали незначний вплив на загальний рівень

задоволеності користувачів, оскільки бот здебільшого справляється з завданнями на високому рівні.

Інтеграція GPT-4o-mini в Telegram-бота принесла значну економію часу і ресурсів. Замінивши оператора, який обробляв би аналогічну кількість запитів вручну, система дозволила значно скоротити витрати на оплату праці, зберігаючи при цьому високу якість обслуговування [39]. Це також дозволило знизити загальні операційні витрати, підвищивши економічну ефективність проекту.

Порівняння з іншими рішеннями для створення віртуальних асистентів показало, що GPT-4o-mini має кілька ключових переваг. Відмінна точність відповідей, швидкість обробки запитів і висока гнучкість у налаштуванні системи зробили це рішення одним із найкращих на ринку. Вартість інтеграції виявилася значно нижчою, ніж у традиційних рішеннях, що потребують додаткових ресурсів для підтримки.

Техніко-економічне обґрунтування підтвердило, що інтеграція GPT-4o-mini в Telegram-бота є економічно вигідною. Зниження витрат на оплату праці оператора, швидка окупність витрат на розробку та значний економічний ефект роблять це рішення привабливим для будь-якої компанії, яка прагне оптимізувати свої витрати і покращити обслуговування клієнтів.

Таким чином, результати аналізу підтверджують високу ефективність інтеграції GPT-4o-mini в Telegram-бота. Це дозволяє не тільки автоматизувати процеси і знизити витрати, а й покращити якість обслуговування, забезпечуючи високу швидкість і точність відповідей. Технологія GPT-4o-mini є надзвичайно перспективним інструментом для бізнесу, що прагне підвищити свою конкурентоспроможність за допомогою сучасних інновацій.

## ВИСНОВКИ

Впровадження нейронної мережі GPT-4o-mini у Telegram-бот є сучасним і перспективним підходом, який сприяє створенню віртуального асистента для ефективної взаємодії з користувачами, покращуючи якість обслуговування та зменшуючи витрати.

Проведені дослідження продемонстрували ефективність Telegram як платформи для розробки ботів завдяки її гнучкості, надійності та доступному Telegram API, що підтримує обробку текстових, мультимедійних і інтерактивних повідомлень. Вибір GPT-4o-mini для реалізації асистента також виявився обґрунтованим, адже ця оптимізована модель забезпечує високу продуктивність у завданнях генерації тексту, персоналізації відповідей і автоматизації бізнес-процесів, навіть за умов обмежених ресурсів. Інтеграція GPT-4o-mini у Telegram-бота дала змогу створити стабільну систему, що ефективно обробляє запити користувачів, аналізує контекст повідомлень і генерує точні відповіді. Використання Telegram API спростило процес налаштування, дозволивши реалізувати інтерактивні функції, зокрема за допомогою Webhook або Polling. Результати тестування підтвердили високу якість роботи бота, хоча окремі неточності у відповідях вказують на можливість подальшого вдосконалення алгоритмів і підвищення якості навчальних даних. Впровадження розробленої системи показало значну економічну ефективність, адже вона знижує витрати на оплату праці та оптимізує бізнес-процеси, забезпечуючи швидку окупність проєкту. Усе це свідчить про те, що поєднання нейронних мереж і платформ для обміну повідомленнями, таких як Telegram, є технічно доцільним і економічно вигідним рішенням для широкого кола задач у різних галузях бізнесу.

Telegram є оптимальною платформою для створення віртуальних асистентів завдяки своєму широкому набору інструментів. Його API дозволяє обробляти текстові повідомлення, мультимедійні файли та інтерактивні елементи, що робить процес розробки зручним і гнучким. Завдяки високому

рівню безпеки, масштабованості та доступності Telegram підходить для інтеграції сучасних нейронних мереж, що надає безліч варіантів для автоматизації різноманітних бізнес-процесів.

GPT-4o-mini є сучасною моделлю штучного інтелекту, яка ефективно виконує завдання генерації тексту, автоматизації відповіді на запити та персоналізації комунікації з клієнтами. Її головна перевага – оптимізована структура, яка дозволяє використовувати цю модель навіть у середовищах із обмеженими ресурсами, зберігаючи швидкість роботи та якість відповідей. Завдяки цьому інтеграція GPT-4o-mini є доступною та економічно вигідною навіть для малого бізнесу.

Інтеграція Telegram-бота з GPT-4o-mini створює систему, яка здатна швидко реагувати на запити користувачів, аналізувати контекст повідомлень і генерувати точні та зрозумілі відповіді. Використання Telegram API дозволяє налаштовувати роботу бота так, щоб він залишався стабільним навіть при високому навантаженні.

Практичне використання такої системи вже показало свою ефективність. Тестування продемонструвало високу швидкість і точність відповідей бота. Незначні недоліки у відповідях можна усунути шляхом подальшого вдосконалення алгоритмів і покращення якості даних, на основі яких навчалася модель. Завдяки цьому система здатна оптимізувати роботу з клієнтами, знижувати витрати на обслуговування, замінюючи ручну працю, та зменшувати навантаження на персонал.

Перспективи впровадження таких рішень дуже широкі. Telegram-бот із GPT-4o-mini може застосовуватися для підтримки клієнтів, автоматичної обробки замовлень, надання інформаційних послуг, а також для інших завдань, що потребують автоматизації комунікацій. Завдяки своїй гнучкості, простоті налаштувань і економічній вигоді, ця технологія підходить для використання в різних галузях, таких як роздрібна торгівля, логістика, фінансові послуги та інші. Це дозволяє бізнесу підвищити ефективність процесів, забезпечити якісний сервіс для клієнтів і знизити витрати.

Розроблений віртуальний асистент на основі Telegram-бота та GPT-4o-mini підтвердив свою ефективність у задачах автоматизації взаємодії з користувачами, продемонструвавши значні переваги в обробці запитів, швидкості відповіді та адаптивності до різних потреб. Сучасні технології нейронних мереж, поєднані з можливостями Telegram, дозволяють створювати багатофункціональні рішення, які є не лише технічно можливими, а й економічно доцільними. Інтеграція GPT-4o-mini забезпечила оптимальне співвідношення продуктивності та витрат, що дозволяє застосовувати розроблену систему в різних бізнес-середовищах, зокрема в компаніях з обмеженими ресурсами.

Поєднання сучасних нейронних мереж і Telegram як платформи для комунікації відкриває нові горизонти для автоматизації. Telegram-боти здатні не лише відповідати на запити користувачів, крім того, може взаємодіяти з іншими сервісами і виконувати складні операції та підтримувати персоналізовану взаємодію. Використання GPT-4o-mini в ролі інтелектуальної бази забезпечує глибоке розуміння контексту запитів, гнучкість у формуванні відповідей і можливість адаптації до специфічних сценаріїв, що робить такого бота надзвичайно корисним для підвищення якості обслуговування клієнтів.

Економічна доцільність рішення підтверджується суттєвим зменшенням витрат на підтримку бізнес-процесів. Інтеграція віртуального асистента замінює значну частину рутинної роботи операторів, що не лише скорочує витрати на оплату праці, але й дозволяє звільнити людські ресурси для вирішення більш складних задач. Водночас система здатна працювати в режимі 24/7, гарантуючи безперервний доступ до послуг і підтримки клієнтів. Це особливо важливо для компаній, які прагнуть розширити свою аудиторію та забезпечити доступність своїх сервісів у будь-який час.

Крім того, інтеграція віртуального асистента в бізнес-процеси відкриває можливості для масштабування і розширення функціональності. З розвитком технологій віртуальні асистенти можуть включати нові функції, такі як голосова підтримка, розпізнавання зображень або інтеграція з системами

управління даними, що дозволяє зробити систему ще більш універсальною. Це забезпечує якісно новий рівень обслуговування, де клієнти отримують швидкі, точні й персоналізовані відповіді, що значно підвищує задоволеність та лояльність користувачів.

Таким чином, результатами роботи є модель віртуального асистента на основі телеграм-боту та генеративного штучного інтелекту, рекомендації щодо інтеграції месенджеру з генеративним штучним інтелектом. Вони можуть бути застосовані для подальших досліджень в цій галузі та при проектуванні хмарних сервісів.