Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *інфокомунікацій*
(повна назва)

Кафедра *інформаційно-мережної інженерії*
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Загальний аналіз напрямків використання нейронних мереж в інфокомунікаціях
(тема)

Виконав:
здобувач *3* року навчання,
групи *ТРІМІу-22-1*
Денис ДУБОГРАЙ
(власне ім'я, прізвище)

Спеціальність *172 Телекомунікації та радіотехніка*
(код і повна назва спеціальності)

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

Освітня програма *Інформаційно-мережна інженерія*
(повна назва освітньої програми)

Керівник: *ст. викл. Олексій ФЕДОРОВ*
(посада власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри _____ *Валерій БЕЗРУК*
(підпис) (власне ім'я, прізвище)

2025 р.

Не містить відомостей заборонених до відкритого публікування

Здобувач / *Дениса ДУБОГРАЯ*/

Керівник / *Олексій ФЕДОРОВ* /

Харківський національний університет радіоелектроніки

Факультет *Інфокомунікацій*
Кафедра *Інформаційно-мережної інженерії*
(повна назва)
Рівень вищої освіти *перший (бакалаврський)*
Спеціальність *172 Телекомунікації та радіотехніка*
(код і повна назва)
Тип програми *освітньо-професійна*
(освітньо-професійна або освітньо-наукова)
Освітня програма *«Інформаційно-мережна інженерія»*
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«*26*» *травня* 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувача *Дубограя Дениса Дмитровича*
(прізвище, ім'я, по батькові)
1. Тема роботи *Загальний аналіз напрямків використання нейронних мереж в інфокомунікаціях*
затверджена наказом університету від «*23*» *травня* 2025 р. № 411 *Ст*
2. Термін подання студентом роботи до екзаменаційної комісії *10 червня 2025 р.*
3. Вихідні дані до роботи *Нейронні мероежі, що аналізуються: ChatGPT від OpenAI, NVIDIA Aerial, vRAN від DeepSig. Інструментарій проведення аналізу: машинне навчання (МН). Основний алгоритм МН, що застосовується: глибинне навчання. Проаналізувати можливості та напрямки прикгичного використання нейронних мереж в інфокомунікаціях. Розробити додаток чат-боту для месенджера Телеграм щодо підтримки взаємодії оператора Internet з користувачами з використанням ChatGPT. Мова створення додатку: Python. Налаштування підключень: бібліотека OpenAI API*

4. Перелік питань, що потрібно опрацювати в роботі
Вступ
1. Загальні особливості розвитку і реалізації технологічних принципів штучного інтелекту.
2. Нейронні мережі для використання в інфокомунікаціях.
3. Аналіз напрямків практичного застосування нейронних мереж в інфокомунікаціях.
4. Розробка додатку підтримки користувачів провайдером internet з використанням ChatGPT
Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) *Слайди у форматі Power Point (назва, мета та етапи виконання роботи, особливості нейронних мереж, класифікація архітектур нейронних мереж, практичні реалізації сучасних нейронних мереж, вид Nvidia Aerial та vRAN, загальний вид архітектури ChatGPT, етапи обробки даних, практичне використання Nvidia Aerial в ІК, практичне використання vRAN в ІК, практичне використання ChatGPT в ІК, процес реєстрації у Telegram-боті підтримки користувачів, основні функції Telegram-боту, висновки)*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	*Ознайомлення із завданням. Уточнення ТЗ.*	26.05 – 27.05.25	*виконано*
2	*Підбір літератури за темою роботи.*	28.05 – 01.06.25	*виконано*
3	*Виконання розділу 1*	02.06 – 07.06.25	*виконано*
4	*Виконання розділу 2*	08.06 – 15.06.25	*виконано*
5	*Виконання розділу 3*	16.06 – 23.06.25	*виконано*
6	*Виконання розділу 4*	24.06 – 06.06.25	*виконано*
7	*Оформлення пояснювальної записки*	07.07 – 12.07.25	*виконано*
8	*Оформлення презентаційного матеріалу,*		
	підготовка до захисту у ЕК, захист.	13.07 – 16.07.25	*виконано*

Дата видачі завдання *26 травня 2025 р.*

Здобувачка _____
(підпис)

Керівник роботи _____ *(ст. викл. Олексій ФЕДОРОВ)*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 80 с., 35 рис., 28 джерел, 3 додатки.

НЕЙРОННА МЕРЕЖА, ШТУЧНИЙ ІНТЕЛЕКТ, AI ШТУЧНІ НЕЙРОННІ МЕРЕЖІ CHATGPT, NVIDIA AERIAL, VRAN, ГЕНЕРАТИВНО ЗМАГАЛЬНА МЕРЕЖА, GAN, РЕКУРЕНТНА НЕЙРОННА МЕРЕЖА, RNN, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, CNN, ТРАНСФОРМЕРИ, TELEGRAM-БОТ.

Мета роботи – аналіз напрямів застосування нейронних мереж ChatGPT від OpenAI, NVIDIA Aerial, vRAN від DeepSig в інфокомунікаційних технологіях.

Об’єкт дослідження – нейронні мережі ChatGPT, NVIDIA Aerial, vRAN.

Досліджено сучасні підходи щодо застосування нейронних мереж, у сфері інфокомунікацій. Проведено аналіз технологічних принципів реалізації та архітектурних різновидів штучних нейронних мереж, при цьому приділена увага питанням машинного навчання, типам нейромереж (CNN, RNN, GAN, трансформерам) та їх особливостям функціонування. Розроблено додаток підтримки користувачів провайдером Internet з використанням ChatGPT у вигляді Telegram-боту, який виконує функції цифрового помічника для користувачів інфокомунікаційних послуг.

THE ABSTRACT

Explanatory note: 80 p., 35 fig., 28 sources, 3 app.

NEURAL NETWORK, ARTIFICIAL INTELLIGENCE, AI, ARTIFICIAL NEURAL NETWORKS, CHATGPT, NVIDIA AERIAL, VRAN, GENERATIVE ADVERSARIAL NETWORK, GAN, RECURRENT NEURAL NETWORK, RNN, CONVOLUTIONAL NEURAL NETWORK, CNN, TRANSFORMERS, TELEGRAM BOT.

The purpose of work – analyze the areas of application OpenAI's ChatGPT, NVIDIA Aerial, and DeepSig's vRAN neural networks in information and communication technologies.

The object of study is ChatGPT, NVIDIA Aerial, and vRAN neural networks.

Modern approaches to the application of neural networks in the field of information and communications have been studied. An analysis of the technological principles of implementation and architectural varieties of artificial neural networks has been carried out, with attention paid to machine learning issues, types of neural networks (CNN, RNN, GAN, transformers) and their functional features. An Internet provider user support application was developed using ChatGPT in the form of a Telegram bot, which performs the functions of a digital assistant for users of information and communication services.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

AI (Artificial Intelligence) – штучний інтелект;

API (Application Programming Interface) – інтерфейс програмування додатків;

ANN (Artificial Neural Networks) – штучні нейронні мережі;

CNN (Convolutional Neural Networks) – згорткові нейронні мережі;

DL (Deep Learning) – глибоке (глибинне) навчання;

DOCA (Data-center composable architecture) – компонована архітектура центру обробки даних;

DPU (Data Processing Unit) – спеціалізований процесор даних;

EM (Electromagnetic simulation) – електромагнітна симуляція;

GAN (Generative Adversarial Networks) – генеративно-змагальна мережа;

GPT (Generative Pre-trained Transformer) – генеративний попередньо тренований трансформер;

GPU (Graphics Processing Unit) – графічний процесор;

LLM (Large Language Model) – велика мовна модель;

MAC (Media Access Control) – рівень управління середовищем;

MIMO (Multiple-Input and Multiple-Output) – системи зв'язку з рознесеними передавальними і приймальними антенами;

ML (Machine Learning) – машинне навчання;

NN (Neural Networks) – нейронна мережа;

QoS (Quality of Service) – якість обслуговування;

RAN (Radio Access Network) – радіодоступна мережа;

PHY (Physical Layer (Level 1, L1)) – фізичний рівень;

RLC (Radio Link Control) – рівень управління радіоканалом;

RLHF (Reinforcement Learning from Human Feedback) – навчання з підкріпленням на основі зворотного зв’язку від людей;

RNN (Recurrent Neural Networks) – рекурентні нейронні мережі;

SDK (Software Development Kit) – комплект для розробки програмного забезпечення;

SDN (Software-Defined Networking) – програмно-визначена мережа;

SINR – (Signal-to-Interference-plus-Noise Ratio) – відношення сигналу до шуму та інтерференції;

UE (User Equipment) – пристроїв користувача;

UI (User Interface) – інтерфейс користувача;

vRAN (virtualized Radio Access Network) – віртуалізована радіодоступна мережа.

ГН – глибоке (глибинне) навчання;

МН – машинне навчання;

НМ – нейронна мережа;

ПЗ – програмне забезпечення;

ШІ – штучний інтелект.

ВСТУП

Сучасний розвиток інфокомунікацій відбувається в умовах стрімкого зростання обсягів переданої інформації, постійного ускладнення мережної інфраструктури та підвищених вимог до швидкості, надійності і безпеки передачі даних. У таких умовах класичні підходи до управління телекомунікаційними системами, обробки трафіку, забезпечення якості обслуговування (Quality of Service, QoS) та безпеки стають малоефективними. Це створює нагальну потребу у впровадженні нових інтелектуальних технологій, які здатні самостійно адаптуватися до змін середовища, виявляти закономірності у великих об’ємах даних та приймати оптимальні рішення в реальному часі.

Штучний інтелект, а зокрема його напрямок – нейронні мережі, відкриває нові можливості для автоматизації та оптимізації процесів в інфокомунікаційних системах. Нейронні мережі вже знаходять застосування у таких напрямках, як інтелектуальна маршрутизація, прогнозування навантаження, управління ресурсами, інформаційна безпека та у різних підходах щодо забезпечення стабільного і надійного зв’язку. Їх здатність до навчання на прикладах, можливості адаптації до нових умов та опрацювання великих об’єм даних, – дозволяє значно підвищити ефективність функціонування інфокомунікаційних мереж.

Варто зазначити, що нейронні мережі є надзвичайно важливими для сучасних інфокомунікацій завдяки своїй гнучкості, масштабованості та здатності обробляти складні, нелінійні задачі. Проте одночасно вони залишаються складними в реалізації, тобто їх налаштування вимагає точного підбору архітектури, безлічі різноманітних параметрів, алгоритмів навчання та оптимізації, а також значних обчислювальних ресурсів. Через це впровадження таких рішень потребує глибокої фахової підготовки, експериментальної перевірки та технічної підтримки, що робить цю сферу складною, але надзвичайно перспективною для постійного моніторингу та аналізу.

Метою цієї кваліфікаційної роботи є аналіз напрямів застосування нейронних мереж в інфокомунікаційних технологіях, оцінка їх потенціалу та визначення практичних переваг для розвитку сучасних інфокомунікацій. Зокрема, у роботі буду приділена увага до аналізу того, яким чином нейронні мережі можуть покращити процеси управління мережею, здійснювати обробку великих

масивів трафіку, прогнозувати навантаження та підвищувати QoS для кінцевих користувачів. Окрему увагу приділено вивченню основ побудови нейромережних моделей, їх архітектурних типів, а також розглядаються методи навчання та адаптації щодо змін, які виникають у інфокомунікаційній інфраструктурі.

Результати проведеного дослідження дозволять краще зрозуміти роль нейронних мереж в інфокомунікаційних системах, оцінити їх ефективність, виявити проблемні аспекти впровадження та обґрунтувати перспективи їх подальшого розвитку. Це говорить про актуальність теми роботи як для загального теоретичного аналізу, так і для практичного застосування в сучасних інфокомунікаційних структурах.

1 ЗАГАЛЬНІ ОСОБЛИВОСТІ РОЗВИТКУ І РЕАЛІЗАЦІЇ ТЕХНОЛОГІЧНИХ ПРИНЦИПІВ ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Етапи появи і розвитку штучного інтелекту

Розвиток сучасного штучного інтелекту (ШІ) починається з XIX століття, коли Чарльз Беббідж створив «різницевий двигун» перший у світі механізм для автоматичного обчислення. Цей винахід заклав підґрунтя для ідеї машинної обробки інформації. У 1950 році Алан Тюрінг запропонував концепцію, відому як тест Тюрінга, що стала фундаментом для розробки методів оцінки інтелектуальних здібностей машин. З того часу розвиток ШІ відбувався стрімко та нерівномірно, характеризуючись періодами активного прогресу і тимчасового згасання інтересу через технічні обмеження. Історія штучного інтелекту охоплює низку ключових етапів, які відображають як наукові прориви, так і соціально-технологічні виклики. Поява глибокого навчання, зростання обчислювальних потужностей і розвиток алгоритмів машинного навчання створили основу для застосування ШІ в різних сферах від обробки мови до автономних систем [1].

Етапи розвитку штучного інтелекту у хронологічному порядку можна сформувати наступним чином (рис. 1.1):

- XIX ст. – Створення Чарльзом Беббіджем «різницевого двигуна», першої спроби реалізації автоматичних обчислень;
- 1950 – Алан Тюрінг публікує роботу «Обчислювальна техніка та інтелект», у якій формулює відомий тест Тюрінга;
- 1956 – Дартмутський літній дослідницький проєкт вважається офіційним початком досліджень у галузі штучного інтелекту;
- 1966–1974 – Період, що отримав назву «перша зима ШІ», коли через завищені очікування знижується фінансування і темп наукових відкриттів. 1980-ті роки – Відродження інтересу до ШІ завдяки розширенню алгоритмів і збільшенню фінансування;
- 1997 – Комп'ютер Deep Blue від IBM перемагає чемпіона світу з шахів Гаррі Каспарова; впровадження системи розпізнавання мови від Dragon Systems у Windows;
- 2011 – ШІ-система Watson від IBM виграє телевізійну вікторину «Jeopardy!», продемонструвавши здатність розуміння природної мови;

- 2012 – Глибоке навчання (deep learning) започатковує нову еру в розвитку ШІ, спричиняючи вибух інтересу до галузі;

- 2016 – AlphaGo, розроблений компанією DeepMind, перемагає чемпіона світу в го – одній із найскладніших ігор;

- 2017 – Швидкий розвиток напрямів комп'ютерного зору, обробки природної мови, робототехніки та автономних систем;

- 2023 – Поява великих мовних моделей (LLM), зокрема GPT-3 та GPT-4 від компанії OpenAI, демонструє потенціал систем ШІ для генерації людиноцентричного тексту, відповідей на запитання, обробки інформації та творчих завдань. Альтернативи до ChatGPT також набувають широкого поширення: Claude від Anthropic, Bard (тепер Gemini) від Google, LLaMA від Meta, Ernie Bot від Baidu, Mistral, а також відкриті моделі, як-от Vicuna, Falcon та інші;

- 2024 – Активне впровадження мультимодального ШІ, здатного працювати одночасно з текстами, зображеннями, аудіо та відео. ШІ-помічники досягають рівня природної взаємодії в реальному часі, що розширює спектр їхнього застосування в освіті, медицині, бізнесі та побуті [2].

Хронологія розвитку штучного інтелекту демонструє поступовий перехід від теоретичних ідей до практичних рішень. З кожним етапом зростали обчислювальні можливості, алгоритми ставали ефективнішими, а сфери застосування – ширшими. Особливо стрімкий розвиток розпочався з 2012 року завдяки глибокому навчанню. Таким чином, історія ШІ – це не лише послідовність технічних досягнень, а також еволюція людських уявлень про інтелект, автоматизацію, творчість і взаємодію людини з машиною. Вивчення цієї хронології дає змогу глибше зрозуміти контекст сучасних технологічних проривів і майбутні перспективи у сфері інфокомунікацій.

Далі розглянемо загальні особливості підходів і моделей щодо реалізації концепції штучного інтелекту.

1.2 Загальні особливості реалізації ШІ в аспекті машинного навчання

Штучні нейронні мережі (Artificial Neural Networks, ANN) є ефективним інструментом машинного навчання (МН), який використовується для виконання завдань, таких як розпізнавання зображень, опрацювання природної мови та ігри.

Рисунок 1.1 – Етапи розвитку ШІ

Навчання ANN, як і інші алгоритми МН, походить від тренувальних даних. Найкраще використовувати ці мережі для неструктурованих даних, де складно зрозуміти, яким чином ознаки пов’язані одна з одною. Щоб це краще зрозуміти, слід звернути увагу на категорії алгоритмів МН і їх взаємозв’язки. Одним із різновидів таких категорій алгоритмів МН є так зване «глибоке навчання», який використовує більш складні архітектурні моделі для створення і навчання

штучних нейронних мереж. Зокрема до таких архітектурних моделей глибоких нейронних мереж відносяться наступні:

- GAN (Generative Adversarial Networks) – генеративно змагальна мережа;
- CNN (Convolutional Neural Network) – згорткова нейронна мережа;
- RNN (Recurrent Neural Network) – рекурентна нейронна мережа;
- трансформормери – архітектура, що заснована на механізмі уваги без використання рекуррентних нейронних мереж [3].

Кожна з цих архітектур створена для певного типу завдань, що обумовлюється характером даних та поставленою проблемою. Наприклад, нейронні мережі на базі архітектури CNN спеціалізуються на обробці зображень, мережі RNN – на аналізі послідовностей, а мережі-трансформери – на роботі з текстовою інформацією. Вибір конкретної архітектури буде залежати від специфіки задачі.

Глибоке навчання, включно з ANN, можна застосовувати для реалізації завдань навчання з учителем, без учителя, а також навчання з підкріпленням. На рис. 1.2 показано, як воно пов'язане з ANN та іншими концепціями МН [4].

Рисунок 1.2 – Схема реалізації глибокого навчання та ANN

Іншими словами ANN можна представити у вигляді моделі життєвого циклу машинного навчання, який показаний на рис. 1.3 [4].

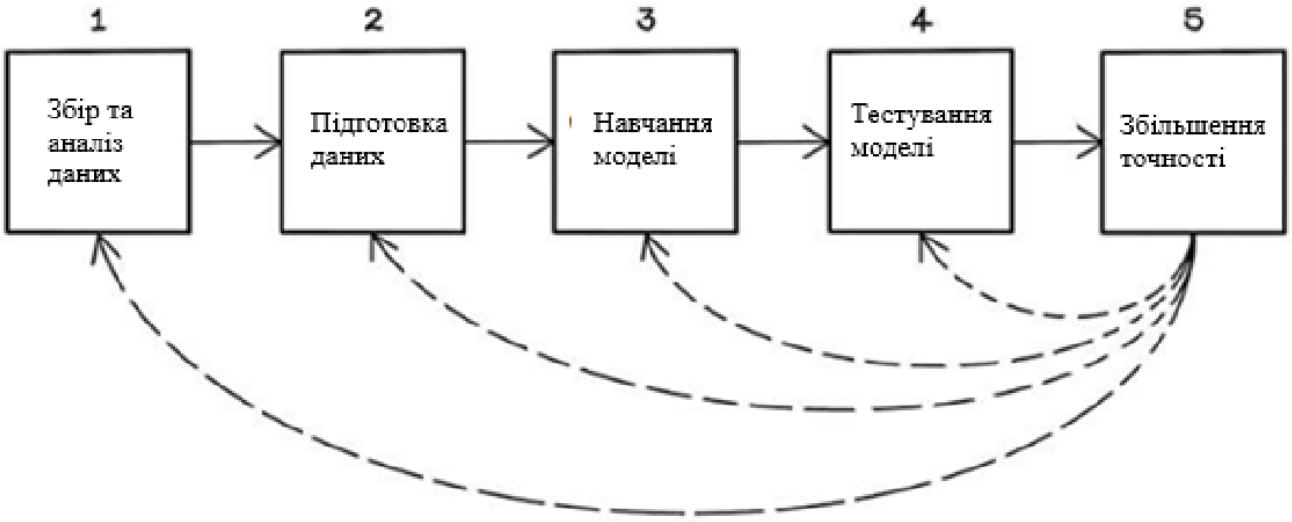

Рисунок 1.3 – Життєвий цикл МН

Після того, як буде сформульоване завдання відбувається збір, аналіз і підготовка відповідних даних. Потім здійснюємо навчання моделі ANN, а на завершення життевого циклу її тестують і за потреби доопрацьовують [4].

Таким чином вище було показано місце і роль ANN, як інструментарію машинного навчання, а також уточнили, що такі нейронні мережі є ще однією моделлю, що навчається за тим же самим життєвим циклом. Тепер перейдемо до розгляду її суті і принципів роботи [4].

1.3 Особливості роботи узагальненої моделі мережі штучного інтелекту

Подібно до генетичних і ройових алгоритмів, ANN були розроблені на основі природних процесів – у цьому разі роботи мозку і нервової системи. Нервова система – це біологічна структура, яка дає змогу відчувати відчуття і лежить в основі функціонування мозку. Людське тіло містить незліченну безліч нервових закінчень і нейронів, які діють за єдиним принципом. Як показано на рис. 1.4, нейрон складається з дендритів (відростків), які отримують сигнали від інших нейронів. Зокрема ці сигнали генеруються наступними їх елементами:

- клітинним тілом та ядром нейрона, що активує та коригує сигнал;

- аксоном, що передає сигнал іншим нейронам та ректирує сигнал;

- синапсами, які переносять і одночасно підлаштовують сигнал перед його передачею дендритам наступних нейронів [4].

Мозок людини складається приблизно з 90 мільярдів взаємодіючих нейронів, завдяки чому вона володіє високорозвиненим інтелектом.

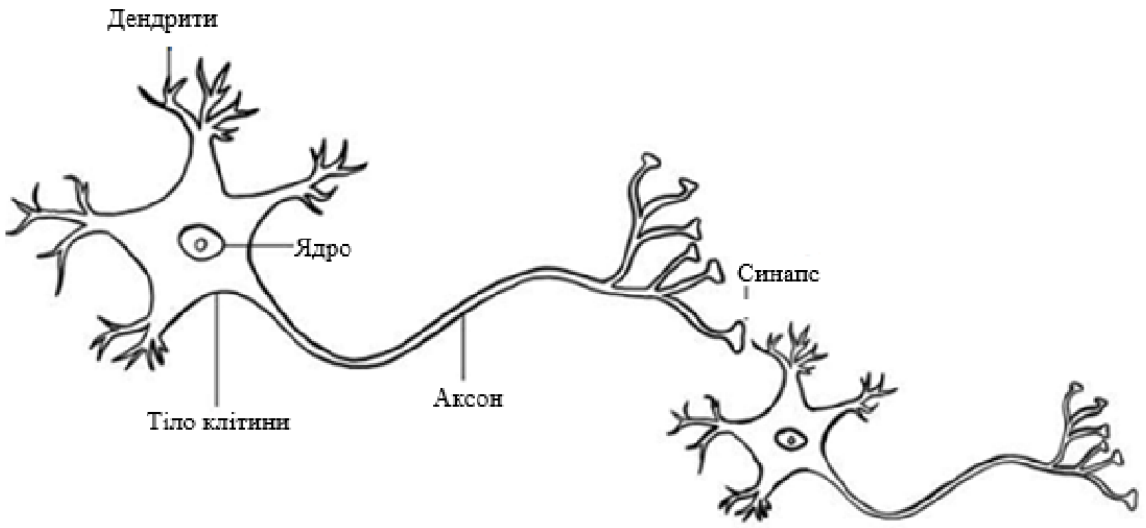

Рисунок 1.4 – Загальна структура зв'язків нейронів

Нейронні мережі (Neural Networks, NN) складаються зі зв'язаних нейронів, які передають інформацію за допомогою електричних або хімічних сигналів. У процесі передачі інформації між нейронами вона коригується, щоб виконувати певну функцію. Тобто коли людина бере чашку і робить ковток води, то за обробку всіх пов'язаних із цим дій і надання зворотного зв'язку про результат відповідають мільйони нейронів, що знаходяться у її мозку, та володіють відповідною інформацією щодо цього процесу. Аналогічно нейронна мережа навчається на прикладах технічної системи. Зокрема, наприклад, робот-маніпулятор спочатку виконує дії з низькою точністю: неправильно захоплює об'єкти або не влучає у ціль (тобто накопичує вміння, щодо здійснення реалізації цього процесу та подальших подібних робочих процесів). Здійснення цих дій на початкових етапах навчання мають багато спроб і помилок, але це дозволяє відпрацювати оптимальність рухів та їх межі корекції. Далі, на основі накопичених даних і зворотного зв’язку, система вдосконалює свою поведінку, і

вже після багатьох ітерацій може точно, стабільно та ефективно виконувати поставлене завдання щодо виконання робочого процесу. На основі наведених прикладів навчання можна показати спрощену модель отримання вхідних сигналів (стимулів), здійснення їх обробки в нейронній мережі з подальшою видачою результатів (відповіді) (рис. 1.5) [1].

Рисунок 1.5 – Спрощена модель біологічної NN [4]

Таким чином, можна бачити, що ANN побудовані та функціонують за принципом біологічних нейронних мереж, але їх не можна назвати повними аналогами. Вони є спрощеною моделлю, адаптованою для розв'язання конкретних задач у сфері машинного навчання. Проте, незважаючи на це спрощення, саме завдяки такій структурі і гнучкості в налаштуваннях штучні нейронні мережі здатні ефективно виконувати складні завдання – від аналізу зображень до розпізнавання мови.

Проектування ANN і її навчання є процесом, що потребує проведення багатьох експериментів, які у значній мірі залежать від конкретної задачі. Ці експерименти дозволяють здійснювати прогнозування архітектури і параметрів моделі, точність якого залежить від кількості зроблених проб і отриманих помилок, які, у свою чергу, дозволяють робити постійне коригування результату. Після завершення навчання мережа може використовуватися різноманітними користувачами в різних системах [1].

Основними елементами будь-якої мережі є її входи та виходи – саме вони визначають, як користувач або система взаємодіятиме з моделлю. Тому важливо правильно визначити, що саме буде подаватися на вхід і які результати очікується побачити на виході. Наприклад, якщо мережа призначена для розпізнавання рукописних цифр (від 0 до 9-ти), то вхідними даними можуть бути значення пікселів зображення розміром 16 × 16, що дозволяє утворити 256 вузлів. Треба зазначити, що кожен вузол знаходиться у своєму шарі (площині) нейронної

мережі, тобто іншими словами кожен вузол являє собою елемент вхідного шару, що відповідає одному пікселю зображення. Тому, як зазначалося на вході утворюється 256 вузлів, а на виході матимемо 10 вузлів, кожен з яких відповідає одній цифрі від 0 до 9. У цьому випадку результат передбачає ймовірність того, що зображення належить до певної цифри [1].

Зауважимо, що нейронна мережа може мати кілька прихованих шарів, і кількість нейронів у кожному з них може бути різною. Додавання шарів допомагає краще впоратися зі складними завданнями, особливо коли потрібно розділити дані на кілька класів. Це пов'язано з поняттям «лінія поділу класів», яка являє собою функцію, що встановлює межу між різними групами даних. Наприклад, якщо є два класи, то лінія поділу буде відокремлювати їх один від одного. У простому випадку – це може бути пряма, але у реальних задачах вона часто має нелінійну форму. Якщо ж дані мають багато вимірів, то візуалізувати таку межу стає неможливо, і тут у нагоді стають додаткові шари мережі. Вони дозволяють знаходити ці складні, багатовимірні функції поділу [5].

Кількість шарів (площин нейронної мережі) і, відповідно, нейронів у NN зазвичай також визначається експериментальним шляхом. З часом, у процесі набуття досвіду, можна навчитися прогнозувати, яка конфігурація краще підходить для певного типу задач. Але ще одним важливим моментом є початкова ініціалізація вагових коефіцієнтів NN, значення яких дозволяє задати початкову точку, від якої алгоритм навчання починає здійснювати налаштування моделі. Якщо значення вагових коефіцієнтів занадто малі, то може виникнути проблема «затухаючого градієнта», тобто сигнали просто буду втрачатися при проходженні через шари. Якщо ж початкові значення занадто великі, то може трапитися «вибух градієнта», тобто коли вагові коефіцієнти «стрибають» і не можливо далі знайти правильне рішення [5].

Існує багато методів ініціалізації вагових коефіцієнтів, що відрізняються своїми переваги і недоліками. Однак є загальне правило, що трактується наступним чином: середнє значення виходів активації кожного шару має бути близьким до нуля, а дисперсія має залишатися однаковою для всіх нейронів. Це допомагає уникнути стрибків у процесі навчання NN і забезпечити стабільнішу роботу мережі протягом усього процесу навчання [1].

Таким чином, можна бачити, що побудова та налаштування штучної нейронної мережі – це не просто питання у кількості шарів чи нейронів, а більш складний процес, який потребує врахування багатьох факторів. Проте, саме

завдяки такій гнучкості і адаптивності різні типи ANN можуть ефективно виконувати широкий спектр завдань – від аналізу зображень до генерації нових даних.

Розглянемо далі більш докладно основні типи архітектур ANN.

1.4 Класифікація архітектурних рішень ANN

Із вищевикладеного можна бачити, що штучні нейронні мережі є універсальними, і їх різні види архітектурних рішень можуть застосовуватися для вирішення відповідних завдань. Вище вже були перераховані типи архітектурних рішень ANN. Зокрема були згадані наступні архітектури, такі як:

1) Згорткові нейронні мережі (CNN) призначені для розпізнавання зображень. Ці мережі використовують для пошуку зв'язків між різними об'єктами та характерними областями зображення. У процесі розпізнавання зображень операція згортки проводиться для кожного пікселя та його сусідів у певному радіусі. Ця техніка широко застосовується для виявлення контурів, підвищення різкості зображення або його розмиття. У CNN використовують згортку та пулінг (об'єднання) для знаходження зв'язків між пікселями. Згортка виявляє ознаки зображення, а пулінг зменшує їх кількість, узагальнюючи знайдені патерни. Це дає змогу ефективно кодувати унікальні ознаки з різноманітних навчальних зразків зображень.

2) Рекурентні нейронні мережі (RNN) – це тип нейронних мереж, у яких зв'язки між елементами утворюють спрямовані послідовності. Така структура дозволяє обробляти часові ряди або інші види послідовностей, наприклад, слова в реченні або звуки у фразі. У RNN реалізовано принцип «пам'яті», який забезпечується через приховані шари, що передають інформацію від одного кроку до наступного. Це дає змогу мережі запам'ятовувати зв'язки між послідовними входами. Під час навчання RNN ваги прихованих шарів коригуються методом зворотного поширення помилки в часі. При цьому одні й ті самі ваги використовуються на різних часових проміжках.

3) Генеративно-змагальні мережі (GAN) – це алгоритми машинного навчання без учителя, що складаються з двох взаємодіючих нейронних мереж. Одна з них – генератор він створює нові дані, наприклад, зображення або ландшафти. Друга – дискримінатор, що аналізує ці згенеровані зразки та порівнює

їх із реальними даними, намагаючись визначити, наскільки правдоподібним є результат. Помилка передається назад, але не просто для корекції, а для покращення здатності генератора створювати реалістичні зображення, а дискримінатора – точніше їх класифікувати. Ці дві частини постійно конкурують між собою, і це призводить до поступового поліпшення якості генерованих зразків.

4) Трансформерні нейронні мережі є одним із найефективніших типів штучних нейронних мереж для обробки послідовних даних. На відміну від згорткових мереж (CNN), які спеціалізуються на аналізі зображень, трансформери призначені для роботи з послідовностями , такими як текст , часові ряди , радіосигнали чи звукові форми хвиль. Саме ця архітектура стала основою для сучасних великих мовних моделей, таких як ChatGPT, BERT, T5 та інших. Трансформери створені для ефективного аналізу взаємозв'язків усередині послідовностей. Наприклад, у завданні обробки природної мови модель має зрозуміти, як слова пов'язані між собою не лише за порядком, а й за значенням. У задачах обробки сигналів трансформери можуть аналізувати часові залежності та виділяти важливі патерни в радіоданих [7].

Описані архітектурні рішення можна розглядати як варіанти конфігурації глибокої нейронної мережі. Вище було зроблене зауваження, що всі архітектурні підходи NN за своєю структурою є багатошаровими, тому що одношарового підходу буде не достатньо для реалізації потрібних її можливостей. Спочатку йде шар, який називають вхідним чи розподільним. Його нейрони (які, як було зазначено у попередньому підрозділі, називають вхідними) приймають елементи вектору ознак і розподіляють їх за нейронами наступного шару. При цьому обробка даних у вхідному шарі не проводиться. Останній шар називається вихідним. На виходах його нейронів (вони називаються вихідними) формується результат роботи мережі – елементи вихідного вектору. Таким чином, загальний фундаментальний підхід щодо формування архітектури нейронної мережі можна показати у наступному вигляді (рис. 1.6) [6]

Між вхідним та вихідним шаром розташовуються один або кілька проміжних або прихованих шарів. Прихованими вони називаються тому, що їх входи та виходи невідомі для зовнішніх по відношенню до нейронної мережі програм і користувача [6].

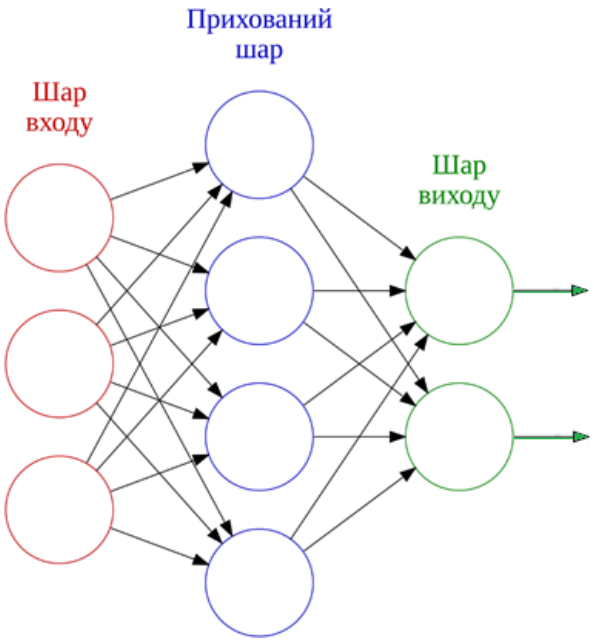

Рисунок 1.6 – Загальний вигляд нейронної мережі

Зазначемо, що багатошарова архітектура нейронних мереж є найбільш популярною та добре розробленою. Вона може моделювати функцію практично будь-якого ступеня складності, причому кількість шарів та число нейронів у кожному шарі визначають складність функції [6].

Таким чином, у процесі опису загальних особливостей і технологічних принципів організації ANN було розглянуто історію розвитку штучного інтелекту, основні принципи його функціонування та ключові типи штучних нейронних мереж. Проаналізовано, як ШІ еволюціонував від теоретичних моделей до практичних реалізацій у формі багатошарових глибоких нейронних архітектур. Проведений аналіз є важливим фундаментом для розуміння сучасних інтелектуальних систем та їх застосування в різних галузях. На основі нього у наступних розділах буде детально розглянуто кілька існуючих практичних реалізацій нейронних мереж, їх структура, принцип роботи та приклади використання.

2 НЕЙРОННІ МЕРЕЖІ ДЛЯ ВИКОРИСТАННЯ В ІНФОКОМУНІКАЦІЯХ

Розглянуто практичні реалізації сучасних нейронних мережі, які широко використовуються в інфокомунікаційній галузі для підвищення ефективності інфокомунікаційних мереж, автоматизації обробки даних та поліпшення взаємодії з користувачами. Серед таких NN можна виділити три ключові рішення:

- платформа NVIDIA Aerial, яка реалізує фізичний рівень мереж 5G/6G на основі графічних процесорів (Graphics Processing Unit, GPU) та технологій штучного інтелекту;

- OmniSIG від компанії DeepSig – нейромережа, що призначена для розпізнавання радіосигналів у реальному часі за допомогою глибокого навчання;

- ChatGPT, що є прикладом потужної мовної моделі. Ця мережа знаходить має широке практичне застосування, а в аспекті аналізу, що проводиться в цій кваліфікаційній роботі, може застосовуватися для створення інтелектуальних чат-ботів, автоматизованих систем підтримки та обробки природної людської мови в інфокомунікаціях.

2.1 Комплексна обчислювальна платформа NVIDIA Aerial

NVIDIA Aerial – це комплекс прискорених обчислювальних платформ, програмного забезпечення (ПЗ) та сервісів для проектування, моделювання та експлуатації бездротових мереж. Ця система містить розширені програмні бібліотеки радіодоступної мережі (Radio Access Network, RAN), що призначені для інфокомунікаційних компаній, постачальників хмарних послуг і підприємств, що будують комерційні мережі 5G. Академічні та галузеві дослідники можуть отримати доступ до Aerial у хмарних або локальних установках для проведення передових досліджень бездротових мереж та розробки алгоритмів штучного інтелекту та машинного навчання для 6G [8].

Мета NVIDIA Aerial полягає в тому, щоб перетворити традиційну апаратно-залежну RAN на програмно-визначену систему, яка може працювати на загальнодоступному комерційному обладнанні. Такий підхід дає можливість: скоротити залежність від виробників спеціалізованого обладнання, підвищити ефективність обчислень, забезпечити більшу гнучкість у тестуванні нових

технологій і це все завдяки унікальній платформі, що має модульну будову, де кожен елемент може бути незалежно налаштований або замінений, що важливо для операторів та дослідників, які хочуть експериментувати з новими методами обробки сигналів, оптимізації параметрів мережі або впровадження алгоритмів штучного інтелекту [8].

NVIDIA Aerial базується на кількох ключових модулях, які забезпечують повноцінне функціонування віртуалізованої мережі. До них входять:

- cuVNF – віртуалізовані функції мережі, що виконуються на GPU;

- cuPHY Layer – програмна реалізація фізичного рівня (Level 1, L1) побудована на основі технології CUDA (дозволяє використовувати GPU для розв'язання задач не пов'язаних з графікою);

- cuBB – базова смуга, реалізована через SDK NVIDIA Aerial;

- DOCA / DPU – використовуються для розвантаження завдань управління потоками та забезпечення безпеки;

- TensorRT, cuDNN, CUDA Graphs – застосовуються для прискорення алгоритмів штучного інтелекту в обробці сигналів [8].

Ці всі компоненти разом формують програмно-визначену RAN-платформу, яка сумісна з сучасними стандартами Open RAN. Більшість з них реалізовано за допомогою C++ та CUDA C, що забезпечує високу продуктивність, особливо при обробці великих обсягів даних у реальному часі. Для створення прототипів, налаштування та інтеграції ML-моделей використовується Python. Інтерфейс ШІ-моделей виконується за допомогою TensorRT, а також бібліотек:

- cuDNN – для операцій з нейронними мережами;

- cuBLAS – для операцій лінійної алгебри, оптимізованих під GPU;

- gRPC та REST API – для взаємодії між компонентами системи, зокрема між control-plane і data-plane [9].

Одним із ключових інструментів реалізації NVIDIA Aerial є Aerial Omniverse Digital Twin, що являє собою масштабовану симуляційну платформу, яка забезпечує фізично точне бездротове середовище. Вона використовує графічні процесори NVIDIA для досягнення максимальної продуктивності, необхідної для реалістичного моделювання. Це значно прискорює розвиток нових функцій у бездротових мережах, надаючи змогу по-новому підходити до проектування, тестування та розгортання мереж [10].

Особливостями Aerial Omniverse Digital Twin є:

- фотореалістичне моделювання – використовується платформа NVIDIA Omniverse, що дозволяє створювати візуально точні моделі оточення;

- висока точність радіоканалів – забезпечується за допомогою трасування променів (ray tracing), що дозволяє точно передбачати затухання сигналу, відбиття, затримки тощо;

- глибока інтеграція з AI/ML – дозволяє аналізувати дані з цифрового двійника та автоматично оптимізовувати параметри мережі;

- моделювання в реальному часі – дозволяє точно відтворювати поведінку мережі в різних умовах, включаючи базові станції, частотні діапазони, активність користувачів, погодні умови тощо [10].

На рисунку 2.1 можна побачити загальну функціональну схему системи Aerial Omniverse Digital Twin та взаємодію її основних компонентів [10].

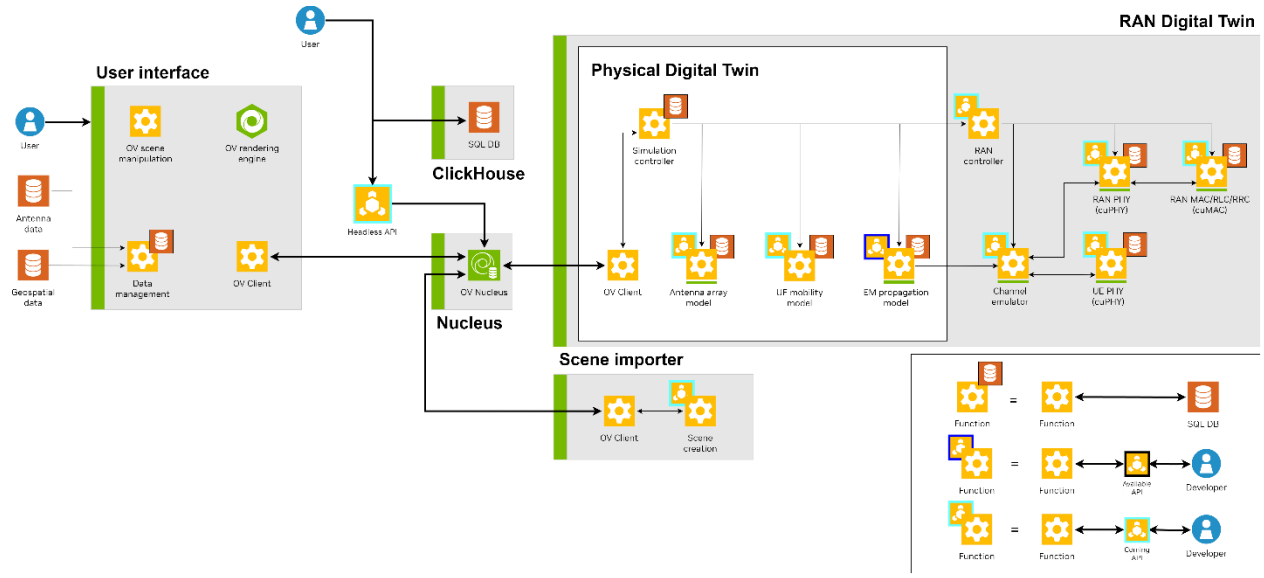

Рисунок 2.1 – Загальна функціональна схема системи Aerial Omniverse Digital Twin

На рис. 2.1, крім розглянутих вище модулів та компонентів, також показані наступні:

- User interface (UI) – графічний інтерфейс, що надає можливість візуалізації та взаємодії зі сценарієм, параметрами, запуску, переривання і зупинки моделювання;

- ClickHouse – у якому результати моделювання зберігаються в SQL-базі на сервері ClickHouse;

- Nucleus – сервер, що надає послуги посередництва повідомлень і забезпечує геометрію сцени для інших компонентів. Має бути розташований на вузлі з доступною IP-адресою;

- Scene importer – імпортер сцени, що приймає геопросторові дані у форматі CityGML і створює активи OpenUSD, необхідні серверу Nucleus для представлення сцени [10].

Для моделювання пристроїв користувача (User Equipment, UE) у цифровому двійнику мережі (RAN) використовується компонент що, відповідає за електромагнітну симуляцію (Electromagnetic simulation, EM). Режим EM-моделювання імітує електромагнітне поширення між передавачами та приймачами у віртуальному 3D-середовищі. Приклад розгортання UE з відповідними інтерфейсами UI можна побачити на рисунку 2.2. На основі UI можна створювати віртуальні смартфони, IoT-пристрої, модеми тощо. Розгортання цих пристроїв можливе двома способами: процедурно, тобто автоматично за допомогою сценаріїв або ж вручну з використанням інтерфейсу користувача [10].

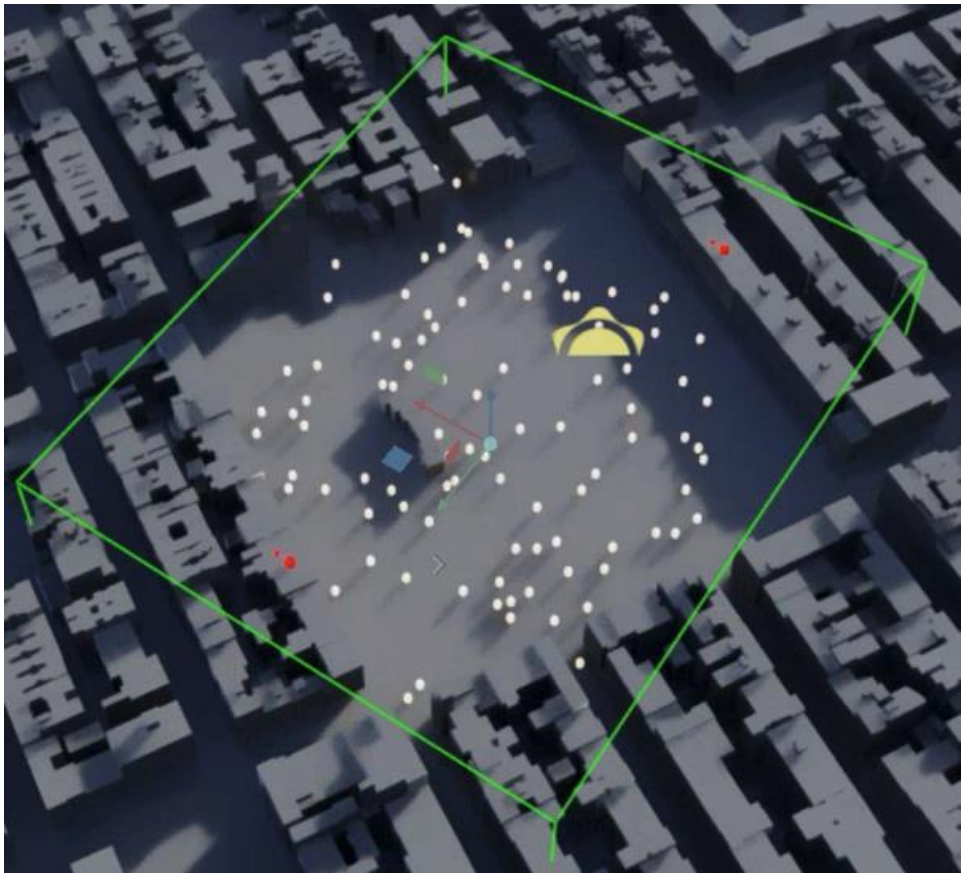

Рисунок 2.2 – Приклад розгортання UE в UI

Ще одним важливим інструментом у складі платформи NVIDIA Aerial є RAN Digital Twin вона являє собою віртуальну модель радіодоступної мережі, побудована на основі фізично точного моделювання та технологій штучного інтелекту. Цей інструмент призначений для тестування, оптимізації та прогнозування поведінки реальної мережі без ризику втручання в її роботу. Це дуже важливо для операторів, які хочуть швидко адаптувати мережі до нових вимог, таких як Open RAN, Massive MIMO або мережі 6G [10].

Цей інструмент має чотири ключові функції, які роблять його особливо корисним у сфері інфокомунікацій:

- моделювання мережі в реальному часі – забезпечує детальне відтворення поведінки всієї RAN-мережі, включаючи базові станції, частотні діапазони, активність користувачів, погодні умови та інші фактори навколишнього середовища;

- оптимізація продуктивності – алгоритми штучного інтелекту та машинного навчання аналізують модель мережі для пошуку найкращої конфігурації обладнання, антен та розподілу трафіку. Це сприяє підвищенню якості сигналу, економії ресурсів та загальної продуктивності системи;

- підтримка автоматизації та тестування – оператори можуть перевіряти нові функції, прошивки та оптимізації в ізольованій мережі, не створюючи ризиків для реальної мережі. Це значно прискорює процес розгортання нових технологій, таких як 5G або Open RAN;

- інтеграція з Omniverse та AI-платформами NVIDIA – для максимально точного та фотореалістичного моделювання середовища використовується платформа NVIDIA Omniverse. Крім того, задіяно SDK NVIDIA Aerial, призначену для розробки та емуляції мереж RAN [10].

Це рішення особливо важливе для наукових досліджень та тестування нових технологій без ризику втручання в реальну мережу. Платформа підходить як для академічних досліджень, так і для мережних операторів та постачальників мережного обладнання, що розвивають так звані мережі наступного покоління.

2.2 Опис технології vRAN від DeepSig

Компанія DeepSig активно займається розвитком інтеграції штучного інтелекту в радіомережі нового покоління, особливо в рамках концепції відкритої

віртуалізованої радіодоступної мережі (virtualized Radio Access Network, vRAN). Завдяки досвіду в галузі машинного навчання і обробки сигналів, DeepSig запропонувала новий підхід до побудови базового діапазону у фізичному рівні L1, де замість традиційних алгоритмів використовуються глибокі нейронні мережі. Це дає можливість не лише значно скоротити обчислювальне навантаження, але також поліпшити пропускну здатність мережі та її стійкість до перешкод [12].

vRAN являє собою підхід до побудови радіомережі, який передбачає віртуалізацію функцій доступу до радіоканалу, щоб забезпечити більшу гнучкість, масштабованість і економічну ефективність порівняно з класичними апаратними системами.

На рисунку 2.3 можна побачити компоненти vRAN, які поділяються на кілька ключових блоків [13]:

- Radio Unit (RU) – відповідає за аналогову обробку сигналів і їх перетворення;
- Distributed Unit (DU) – реалізує верхній фізичний рівень upper L1, MAC- та RLC-рівні;
- Central Unit (CU) – виконує обробку PDCP, SDAP та управління мережею.

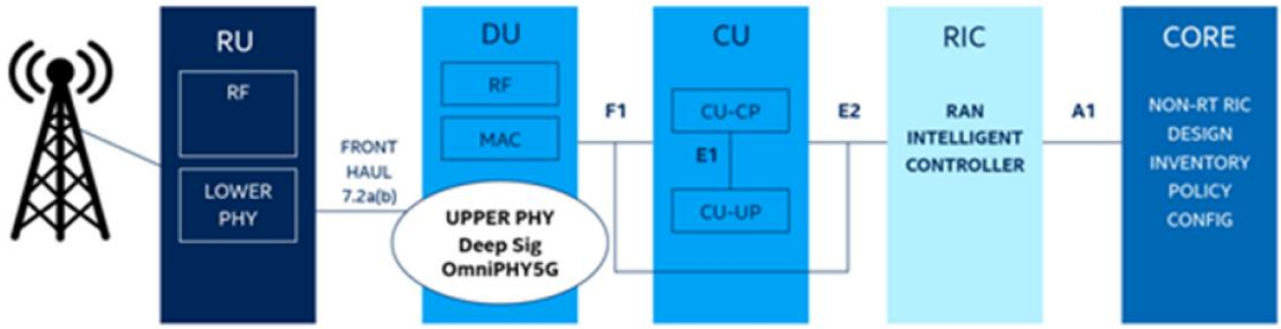

Рисунок 2.3 – Компоненти відкритої архітектури vRAN 5G

Така модульна організація дозволяє операторам будувати мережі, які не залежать від конкретного виробника апаратного забезпечення, а також легко оновлювати, тестувати та оптимізувати окремі компоненти без участі виробників обладнання [12].

У рамках своїх досліджень і розробок DeepSig запропонувала програмне забезпечення OmniPHY 5G AI, яке замінює кілька стандартних алгоритмів обробки сигналів у базовому діапазоні глибокою нейронною мережею. Такий

підхід має кілька суттєвих переваг, а саме менше обчислювального навантаження, підвищення рівня відношення сигнал/перешкода «плюс» шум (Signal-to-Interference-plus-Noise Ratio, SINR), краща спектральна ефективність, зменшення капітальних і операційних витрат. Це рішення сумісне з архітектурою RAN-подібної нейронної мережі Intel FlexRAN Reference Architecture, яка оптимізує використання ресурсів і відокремлює мережні функції від базового обладнання, в якості якого використовується стандартне серверне устаткування. Це дозволяє, в першу чергу, за рахунок розмежування площин управління, які контролюють процеси передачі даних, і площин даних, які фактично здійснюють передачу цих даних, – знизити витрати і підвищити гнучкість та динамічність середовища RAN-мережі. Зокрема архітектура Intel FlexRAN RA дозволяє операторам розгортати компоненти 4G/5G-систем на фізичному рівні (Physical layer, PHY), канальному рівні управління середовищем (Media Access Control, MAC) та рівні управління радіоканалом (Radio Link Control, RLC) у вигляді ПЗ, що працює на процесорах Intel, а також дозволяє легко інтегрувати його в існуючу платформу без потреби змінювати апаратну частину. Це ПЗ вбудовується прямо в SDK Intel FlexRAN, де замінює такі функції, як оцінка каналу, SRS-обробка та передкодування. Таким чином, модель стає частиною стека протоколів і одночасно залишається сумісною з існуючими стандартами [11].

Співпраця DeepSig з Intel стала важливим кроком у масовому впровадженні цієї технології. Компанія інтегрувала OmniPHY 5G AI в Intel FlexRAN SDK, що забезпечило їх сумісність з існуючими системами та легкість впровадження. Додаткові модулі були додані в верхній L1, де замінюють традиційні функції оцінки каналу, SRS-обробки та передкодування. Це забезпечило не лише сумісність із існуючими стандартами, а також додало нові механізми для підвищення якості послуг, скорочення обчислювального навантаження та поліпшення спектральної ефективності [13].

У березні 2021 року DeepSig запустила свій 5G Wireless AI Lab, де було створено повністю функціональну мережу 5G SA на основі Open vRAN. Були проведені OTA-тести, які показали поліпшення SINR на 2 - 5 дБ, збільшення пропускної здатності на 10 - 15%, скорочення обчислювального навантаження на 20 - 30%. Ці дані підтверджують, що впровадження глибокого навчання в фізичний рівень може суттєво вплинути на характеристики RAN, роблячи їх більш адаптивними, швидкодіючими та енергоефективними.

З вищевикладеного можна бачити, що технологія vRAN від DeepSig стала однією із найперспективніших напрямів у сфері бездротових комунікацій. Завдяки використанню глибокого навчання на фізичному рівні, модель може самостійно аналізувати стан мережі, прогнозувати зміни та оптимізувати параметри без додаткового кодування. Треба зазначити, що все ж таки модель має деякі обмеження, зокрема вимагає регулярного оновлення даних для підтримки актуальності. Її інтеграція в існуючі SDK та сумісність із існуючими стандартами роблять її готовою до масового застосування.

Також потрібно зазначити, що DeepSig активно досліджує можливості використання цієї технології в майбутніх поколіннях мереж 5G Advanced та 6G, де роль штучного інтелекту очікується ще більш суттєвою. Таким чином, впровадження ШІ в фізичний рівень RAN не лише поліпшує якість сигналу та ефективність обчислень, а також відкриває нові напрями для розвитку бездротових мереж.

2.3 Архітектура та принципи роботи ChatGPT від OpenAI

ChatGPT – це інтерактивна діалогова система штучного інтелекту, створена компанією OpenAI на основі архітектури Генеративного попередньо навченого трансформеру (Generative Pre-trained Transformer, GPT). Основна задача системи полягає у генерації текстів природною мовою, які були логічними, контекстно узгодженими та максимально наближеними до людських. Для цього модель навчається на величезних масивах текстової інформації, виявляючи статистичні закономірності між словами та їх комбінаціями [14].

Основою технології є трансформер – тип нейронної мережі, запропонований у 2017 році у науковій праці «Attention Is All You Need» [14]. Стисло цей тип архітектури NN був розглянутий у першому розділі цієї кваліфікаційної роботи. Але треба окремо звернути увагу, що саме ця архітектура, стала основою для серії моделей GPT, починаючи з GPT-1 (рис. 2.5).

У 2019 році з'явилася GPT-2, яка могла генерувати текст такого рівня, що були певні заборони, стосовно її загального використання у суспільстві з питань безпеки. Подальшим розвитком цієї технології стала поява GPT-3 у 2020 році, яка вже мала 175 мільярдів параметрів. Пізніше, у 2022 році, OpenAI представила ChatGPT, що побудована на базі модифікованої версії GPT-3.5, яка була

оптимізована для діалогового режиму. Наступний етап – це поява у 2023 році GPT-4, що є більш потужною та багатомодальною моделлю нейронної мережі, яка здатна обробляти не лише текст, а і зображення, забезпечуючи більш точні результати [15].

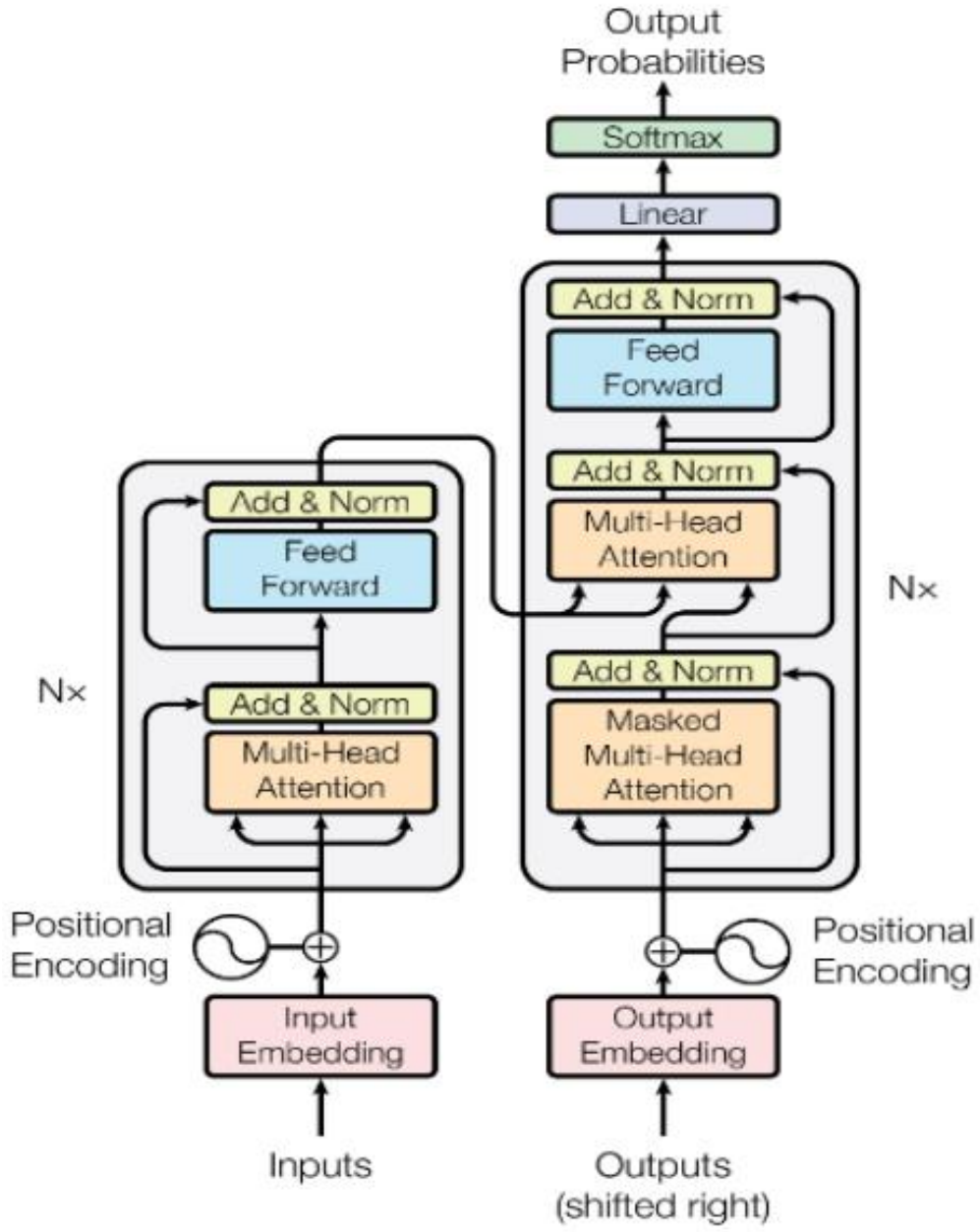

Рисунок 2.5 – Архітектура трансформерної нейронної мережі

У цій NN для аналізу кожного слова в контексті всіх інших у реченні використовується механізм самоуваги, реалізація якого показана на рис. 2.6 [16].

Механізм самоуваги – це ключ до розуміння моделлю смислу речення. Кожне слово в реченні аналізується не ізольовано, а разом з усіма іншими словами, що дає змогу точно визначати залежність між ними.

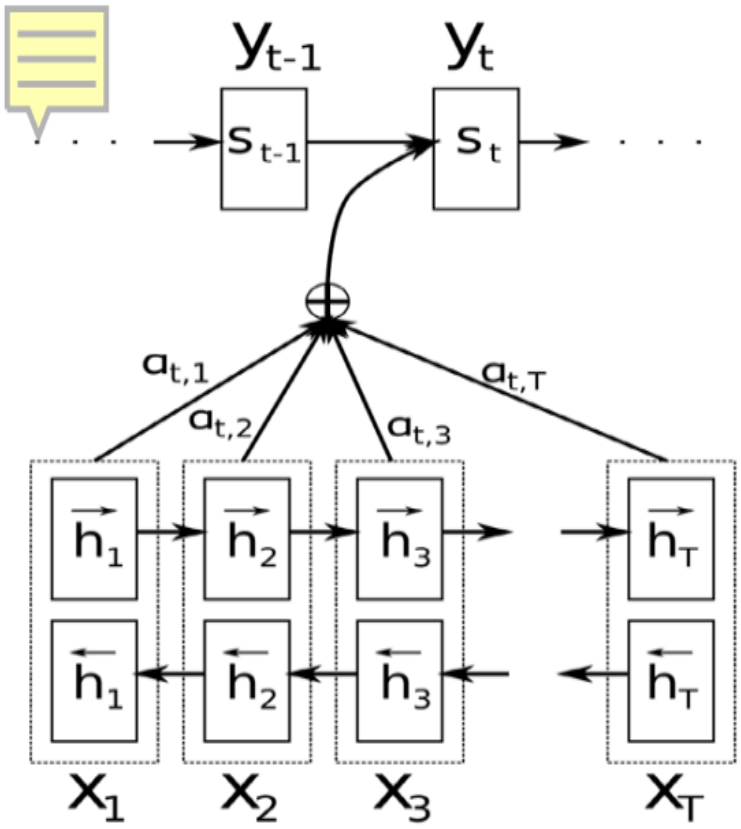

Рисунок 2.6 – Механізм самоуваги

На рис. 2.6 наведені наступні позначення [16]:

- x_j – елемент вхідної послідовності;

- h_j та h_{j*} – приховані стани прямого і зворотнього напряму;

- α_{ji} – активація на момент часу j;

- s_j – вихідний стан на момент часу j;

- y_j – вихідний вектор на момент часу j.

Тепер розглянемо як працює ця схема.

Для кожного вихідного слова x_j створюється анотація h_j шляхом конкатенації (функція з'єднання двох сутностей або результат такої операції) прихованих станів прямого і зворотного проходу. Далі для кожного цільового

слова y_j модель обчислює окремий контекстний вектор c_j, як зважену суму всіх вихідних анотацій, за наступною формулою:

$$c_j = \sum_{j=1}^{T_x} \alpha_{ji} h_j \ , \tag{2.1}$$

де T_x – це кількість елементів (токенів, слів, ознак) у вхідному реченні (або вхідному сигналі), яке обробляє енкодер.

Ваги уваги показують, скільки уваги слід приділити j-му вихідному слову при генерації i-го цільового слова. Ці ваги обчислюються за виразом, який описує модель вирівнювання за softmax-функцією, яка нормалізує ваги у межах одного вихідного кроку [16]:

$$\alpha_{ji} = \frac{\exp(e_{ji})}{\sum_{j=1}^{T_x} \exp(e_{ji})}, \tag{2.2}$$

Із виразу (2.2) можна зробити оцінку вирівнювання (e_{ji}), яка визначає можливість сумісності між попереднім станом декодера та кожною вихідною анотацією. Для цього використовуємо вираз [16]:

$$e_{ji} = (S_{i-1}, h_j). \tag{2.3}$$

Таким чином, на кожному кроці генерації нового слова, мережа не просто дивиться на фіксований вектор, а динамічно фокусується на найбільш релевантних частинах вхідної послідовності. Це дозволяє моделі краще обробляти довгі речення та зберігати смислові залежності між словами. Тобто цей підхід дозволяє моделі ефективно зберігати довгострокову залежність між словами, що особливо важливо для розуміння багатозначних фраз чи складних конструкцій. Іншими словами, така архітектура забезпечує високу якість розуміння мови порівняно з класичними рекурентними мережами.

Архітектура, що наведена на рис. 2.5, передбачає кілька ключових етапів обробки даних (рис. 2.7) [17]:

- токенізація (Tokenizstion) – текст розбивається на окремі частини, які називаються токенами;

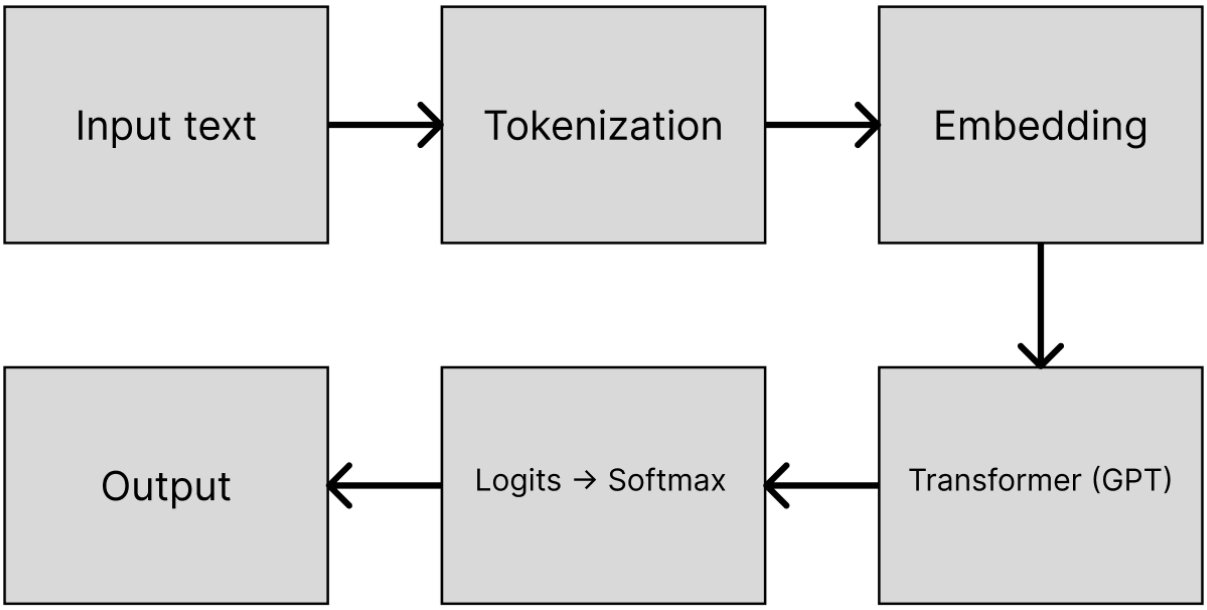

Рисунок 2.7 – Приклад обробки запита користувача

- ембеддинг (Embedding) – кожному токену присвоюється числовий вектор, який кодує значення слова та його місце в реченні;

- обробка за допомогою трансформера (Transformer (GPT)) – саме тут відбувається аналіз контексту, з використанням self-attention та нормалізації шарів

- логіти та Softmax (Logits and Softmax) – система визначає найімовірніше продовження речення, обираючи з кандидатів на основі математичної ймовірності

- генерація вихідного тексту (Output) – результат подається користувачеві

Цей процес повторюється багаторазово, поки модель не сформує повну відповідь, яка має сенс і відповідає заданому запиту [17].

Важливим етапом розробки ChatGPT стало впровадження методу (Reinforcement Learning from Human Feedback, RLHF), тобто навчання з підкріпленням на основі зворотного зв’язку від людей. Після початкового навчання на великих корпусах текстів, модель проходить додаткову оптимізацію: люди-анотатори оцінюють кілька варіантів відповідей, після чого модель навчається вибирати найбільш точні, етично прийнятні та зрозумілі варіанти. Такий підхід значно зменшує кількість помилок, скорочує кількість токсичних відповідей і підвищує відповідність запитам користувача. Для навчання використовувався алгоритм оптимізації наближеної політики (Proximal Policy Optimization, PPO), який є одним із ефективних методів навчання політики в середовищах з підкріпленням. Він дозволяє моделі навчитися не лише правильно

відповідати, але також і вести себе відповідно до етичних норм та очікувань користувачів [14].

Особливістю ChatGPT є те, що вона може адаптуватися до широкого кола задач – від написання текстів до генерації програмного коду, перекладу мов, логічних висновків тощо. Модель не просто відтворює текст, а аналізує контекст, встановлює логічні зв’язки між поняттями, а також може вести тривалі діалоги, запам’ятовуючи попередній контекст. Після початкового навчання, ця модель проходить додаткову оптимізацію з метою кращого узгодження з намірами користувача. Цей процес називається InstructGPT і полягає в тому, що модель навчається дотримуватися інструкцій, які надходять від користувачів. Це дозволило значно поліпшити якість відповіді, зменшити кількість невідповіднеостей та підвищити корисність моделі для виконання практичних задач.

Як було зазначено вище, архітектурні особливості та підхід до навчання ChatGPT дають змогу моделі не лише генерувати тексти, а також ефективно працювати з контекстом, розуміти інструкції та адаптуватися до запитів користувача.

Таким чином, можна бачити, що поєднання механізму самоуваги з навчанням на основі інструкцій, які надходять від користувачів, забезпечує баланс між глибоким лінгвістичним аналізом і практичною корисністю. Такий підхід виводить сучасні мовні моделі, що реалізуються у нейронних мережах, на новий рівень, де вони стають не просто інструментами автоматизації, а повноцінними цифровими асистентами.

3 АНАЛІЗ НАПРЯМКІВ ПРАКТИЧНОГО ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ІНФОКОМУНІКАЦІЯХ

Впровадження нейронних мереж у реальні інфокомунікаційні системи є складним процесом, який потребує не лише правильної побудови моделі, а також її оптимізації для конкретної задачі та середовища виконання Наприклад, при обробці радіосигналів важливим фактором є швидкість реакції системи, тобто латентність, тоді як у чат-ботах переважну роль відіграє точність розуміння мови. Тому при виборі архітектури та способу реалізації, модель має бути адаптована до специфіки домену. Ключовими етапами впровадження ШІ-моделей є підготовка даних, тренування моделі, тестування, оптимізація під пристрій, інтеграція в існуючу систему, а також моніторинг продуктивності. Кожен із цих етапів має свої виклики. У всіх з трьох нейронних мереж які викладені в цій роботі є свої плюси і свої мінуси, тож розглянемо реальні приклади реалізації та проблеми які виникають при інтеграції цих мереж

3.1 Застосування комплексної платформи NVIDIA Aerial

Як зазначалося в попередньому розділі, ця платформа створена для підвищення ефективності обробки сигналів у режимі реального часу, особливо в контексті технологій 5G, де вимоги до якості зв’язку, пропускної спроможності та латентності значно перевищують можливості класичних рішень цифрових сигнальних процесорів (Digital Signal Processor, DSP). Її основною метою створення було забезпечення більш гнучкого та ефективного управління мережею завдяки використанню нейронних мереж, які можуть адаптуватись до змін у середовищі передачі сигналів. Застосування алгоритмів глибокого навчання в рамках цієї моделі дозволяє підтримувати функції демодуляції, кодування, виявлення помилок та оптимізації частотних характеристик. Тобто застосування цієї платформи дозволяє не лише поліпшити якість сигналу, але й також скоротити кількість завад, що виникають у складних умовах поширення радіохвиль [18].

Особливістю цієї платформи є те, що всі компоненти можуть бути розташовані як у хмарному оточенні, так і на серверах, до яких безпосередньо

підключені користувачі та через які вони взаємодіють з дата-центрами. Це дозволяє забезпечити широкі можливості щодо впровадження в різноманітні мережні структури. Крім того, NVIDIA Aerial підтримує інтеграцію з такими фреймворками, як Triton Inference Server, RAPIDS та Docker, що дозволяє забезпечити високу сумісність із сучасними інфраструктурними рішеннями [18].

В цих інфраструктурах саме NVIDIA Aerial може виконувати широкий спектр задач, які пов'язані з обробкою радіосигналів. Зокрема, система може [19]:

- виконувати демодуляцію та модуляцію сигналів;
- обробляти сигнали в реальному часі;
- адаптувати параметри передачі в залежності від умов оточення;
- детектувати та коригувати помилки, що виникають під час передачі даних.

Ці функції мають велике значення для сучасних інфокомунікаціях, особливо в умовах високої завантаженості мережі або складних радіоумовах, коли традиційні методи цифрової обробки сигналів починають давати великі похибки. Саме із-за цих проблем деякі компанії використовують NVIDIA Aerial у своїх системах.

Наприклад, ця система успішно використовувалася телекомунікаційною компанією Verizon, яка надає послуги зв’язку у США, для покращення якості сигналу в міських районах, де велика кількість будівель та інших завад призводить до значного зменшення QoS. Впровадження NVIDIA Aerial дозволило не лише зменшити кількість помилок передачі, а також підвищити стабільність з’єднання для користувачів [19].

Крім компанії Verizon, платформу NVIDIA Aerial активно тестувала Samsung Networks в рамках проектів організації програмно-визначених мереж (Software-Defined Networking, SDN). У рамках одного з проектів NVIDIA Aerial була використана для оптимізації роботи базових станцій у регіонах із високою щільністю абонентів, що дозволило досягти зростання пропускної здатності на 25% та скорочення кількості відключень на 15% [20].

Ще одним прикладом є партнерство NVIDIA з оператором мобільного зв’язку Vodafone, де платформу NVIDIA Aerial була застосована для оптимізації смуги пропускання в умовах сезонного зростання навантаження. Це дозволило автоматизувати процеси управління мережею, що зменшило кількість людських помилок та скоротило витрати на технічне обслуговування [21].

Платформа NVIDIA Aerial в порівнянні з класичними системами на основі DSP-платформ, що використовувалися мережними операторами для обробки сигналів, має наступні переваги [18]:

- висока продуктивність – завдяки GPU-прискоренню система може одночасно обробляти сотні каналів без зниження QoS;

- гнучкість – модель може бути оновлена без зупинки роботи мережі, що забезпечує більшу надійність;

- масштабованість – система підходить як для малих комунікаційних точок, так і для великих центрів обробки даних;

- здатність до адаптації – NVIDIA Aerial може автоматично змінювати параметри обробки сигналів залежно від умов оточення.

Але, разом із перевагами існують і обмеження для використання платформи NVIDIA Aerial [18]:

- високе енергоспоживання – GPU-сервери потребують значного енергетичного живлення та охолодження;

- складність інтеграції – не всі існуючі телекомунікаційні мережі готові до такого рівня модернізації;

вартість обладнання – впровадження потребує суттєвих капіталовкладень.

Таким чином, можна зазначити, що платформа NVIDIA Aerial є однією з провідних технологій у сфері радіомереж нового покоління. Проте, подальший її розвиток має бути спрямований на:

- оптимізацію енергоефективності – зокрема, шляхом удосконалення алгоритмів та зменшення обчислювальної складності моделі;

- забезпечення сумісності з Open RAN – що дозволить інтегрувати модель у більш широкий спектр інфраструктур;

- розробку більш спрощених версій моделі для можливості використання в пристроях із обмеженими ресурсами, які безпосередньо взаємодіють з користувачами.

З урахуванням темпу розвитку штучного інтелекту в телекомунікаціях, можна очікувати, що NVIDIA Aerial у найближчі роки може стати стандартом для 6G-мереж, забезпечившии не лише високу якість зв’язку, але й також нові можливості для автоматизації та самооптимізації мережі.

3.2 Застосування vRAN від компанії DeepSig

vRAN від компанії DeepSig є також прикладом успішного застосування алгоритмів глибокого навчання для реалізації функцій PHY радіомережі у SDN-середовищі. На відміну від традиційних радіомереж, де обробка сигналів здійснюється на основі жорстко заданих правил, vRAN використовує нейронні мережі для адаптації до умов оточення, а це забезпечує більшу гнучкість і ефективність виконання радіомережами своїх функцій [22].

У другому розділі зазначалося, що метою створення архітектури vRAN є заміна традиційних алгоритмів цифрової обробки сигналів на моделі глибокого навчання, які можуть самостійно вчитися на реальних даних, що надходять із базових станцій, аналізувати шуми, завади та оптимізувати параметри передачі сигналу [21]. Це особливо важливо в умовах непостійного радіооточення, де стандартні методи часто не справляються з задачами стабільної передачі даних. Саме тому компанія DeepSig розробила модульну архітектуру.

Вона дозволяє легко інтегрувати модель в існуючу інфраструктуру та адаптувати її до специфіки мережі. Як було розглянуто в другому розділі основними компонентами системи є [22]:

- сигнальний процесор на основі AI – виконує демодуляцію, модуляцію, виявлення сигналів, виправлення помилок;

- інтелектуальний контролер RAN (RAN Intelligent Controller, RIC) – який керує поведінкою моделі та забезпечує взаємодію з іншими частинами мережі;

- інтерфейси програмування додатків (Application Programming Interface, API)– через які система взаємодіє з Open RAN-архітектурою та хмарними платформами;

- програмні засоби розробки (Software Development Kit, SDK) та інструменти тренування – які дозволяють операторам налаштовувати модель під конкретні умови та типи сигналів.

Особливістю vRAN є те, що модель забезпечує максимальну точність у реальних умовах, тому що навчається безпосередньо на даних, які надходять із базових станцій. Такий підхід відрізняється від класичних алгоритмів на основі DSP-процесорів, які не здатні адаптуватись до змін у радіоумовах [22].

Саме завдяки цим особливостям vRAN використовують багато компаній, з яких можна виділити наступні:

- Rakuten Symphony – компанія, що спеціалізується на програмно-визначених мережах;

- T-Mobile – велика міжнародна компанія, що надає послуги мобільного зв'язку;

- Orange – французька телекомунікаційна компанія, яка надає послуги мобільного зв'язку, фіксованого зв'язку та доступу до інтернету.

Так, компанія Rakuten Symphony використала vRAN для оптимізації радіомереж у своєму проекті, який стосувався організації та розгортання SDN-мереж. Метою було створення гнучкої та масштабованої інфраструктури, яка здатна швидко адаптуватися до змін у навантаженні та радіоумовах [23].

Основні результати [23]:

- покращення якості сигналу на 25 - 30% у регіонах із високою щільністю користувачів;

- скорочення кількості помилок передачі на 18 - 22%;

- зменшення часу реакції мережі на зміни у середовищі до 40% за рахунок самоадаптації моделі.

Це дозволило Rakuten Symphony не лише поліпшити якість послуг, а ще скоротити витрати на обслуговування мережі завдяки автоматизації процесів управління.

Компанія T-Mobile використала vRAN у великих міських районах, де щільність будівель і перешкод значно впливає на якість радіосигналу. У рамках тестування система була інтегрована в одну з базових станцій у Нью-Йорку, де аналізувалися показники сигналу до і після впровадження.

Результати тестування показали [24]:

- середній рівень сигналу підвищився на 12 - 15 dBm;

- кількість повторних спроб з'єднання скоротилася на 27%;

- стабільність з’єднання зросла на 30% в умовах високого рівня завад.

Особливим досягненням стало те, що vRAN змогла адаптуватися до сезонного зростання трафіку (наприклад, під час святкових подій), що забезпечило безперебійну роботу мережі без додаткового втручання операторів.

Orange протестував vRAN у сільських районах Франції, де сигнали часто втрачаються через велику відстань між базовими станціями та абонентами. Метою було перевірити, чи зможе система покращити якість зв’язку в умовах слабкого сигналу.

За результатами тестування були отримані дані, які свідчили про покращення параметрів якості зв’язку, зокрема було отримано, що [25]:

- рівень сигналу підвищився на 18 - 20 dBm;

- швидкість передачі даних зросла на 22%;
- відсоток втрачених пакетів даних скоротився на 35%.

Також було у результаті проведеного тестування було зафіксоване значне поліпшення у здатності системи виявляти користувачів у складних умовах, що особливо важливо для сільських територій з обмеженою інфраструктурою.

Наразі DeepSig разом із партнерами розробляє версію vRAN для використання в IoT-пристроях та автономних комунікаційних системах, таких як дрони, автономні машини чи інтелектуальні датчики.

Перевагами впровадження буде можливість роботи в умовах обмеженого зв'язку, автономна адаптація до змін у радіоумовах без участі людини та можливість запуску на легких обчислювальних пристроях.

Аналіз прикладів впровадження vRAN від DeepSig демонструє, що ця технологія має високий практичний потенціал в різноманітних умовах: від міських до сільських та від стандартних мереж до автономних систем. Особливістю є її здатність до адаптації в реальних умовах, що забезпечує кращу якість сигналу у порівнянні з класичними DSP-рішеннями.

3.3 Застосування ChatGPT у функціонуванні сучасних інфокомунікаційних систем та їх послугах і додатках

Як вже було зазначено в другому розділі, ChatGPT був розроблений компанією OpenAI. Він відноситься до групи великих мовних моделей (LLM), яка використовується для генерації тексту, аналізу даних, автоматизації спілкування з користувачами через чат-ботів, а також інтеграції в інші інфокомунікаційні сервіси. На відміну від нейронних мереж, що призначені для обробки сигналів та які розглядались в попередніх підрозділах (NVIDIA Aerial чи vRAN від DeepSig), ChatGPT працює на рівнях взаємодії з користувачем, забезпечуючи не лише технічну ефективність а і високий рівень взаємодії та автоматизації. Тому велика кількість компаній і виробників ПЗ вибирає ChatGPT для впровадження в свої програмні продукти, з яких можна виділити [26]:

- Zendesk – автоматизація служби підтримки;
- Vodafone – аналіз зворотного зв'язку від користувачів;
- Samsung SmartThings – голосові помічники;
- Microsoft Teams – інтеграція в комунікаційні платформи;

- Cisco Webex – інтеграція в корпоративні комунікації.

Окремо треба звернути увагу також на те, що ChatGPT використовується у великій кількості Telegram-ботів та месенджерах.

Наведемо приклад застосування ChatGPT в продуктах компанії Zendesk, яка є однією з найпопулярніших платформ для здійснення управління взаємодією з клієнтами, та яка надає інструменти для створення систем технічної підтримки, обробки запитів через електронну пошту, телефон, чати та месенджери [26]. Наразі Zendesk активно впроваджує штучний інтелект, особливо моделі на основі Transformer, серед яких ChatGPT використовується для автоматизації частини процесів та покращення якості обслуговування користувачів.

Така автоматизації процесів з використанням великих мовних моделей дозволяє [27]:

- скоротити час очікування відповіді;
- покращити точність обробки запитів;
- вивільнити операторів для складніших завдань;
- автоматично класифікувати типи скарг та запитів.

У рамках своєї платформи Zendesk інтегрувала ChatGPT для обробки запитів користувачів. Це реалізовано через так звані AI Sidebars (інтерактивні бічні панелі), де модель аналізує текст запиту і формує рекомендації для оператора або відповідає автоматично, якщо запит є типовим [27].

Основні функції ChatGPT у Zendesk [28]:

- автоматична відповідь на запити – система виявляє запити, які мають стандартні відповіді (наприклад, про активацію послуги, зміну паролів, оплату рахунків) і генерує відповідь без участі спеціаліста з технічної підтримки;

- класифікація запитів – модель визначає категорію проблеми (фінансова, технічна, адміністративна) і направляє її на відповідний канал або до потрібного спеціаліста;

- переклад запитів та відповідь мовою користувача – особливо важливо для глобальних компаній, де багато інтернаціональних клієнтів, які говорять різними мовами;

- генерація відповідей для операторів – ChatGPT допомагає операторам формувати більш точні та узгоджені відповіді, базуючись на внутрішній політиці компанії та інформації, що наведена у вкладці «питання, що часто задаються»;

- сумаризація листування – модель виділяє ключові моменти з довгих переписок, що скорочує час на аналіз запиту оператором.

Telegram-боти реалізуються через Telegram Bot API, який дозволяє приймати повідомлення від користувача, направляти іх до ChatGPT, де він буде аналізувати їх і формувати відповідь. У випадку інтеграції з ChatGPT, процес має такі етапи [28]:

- користувач надсилає запит через Telegram-бота.

- бот отримує повідомлення через Webhook (спосіб, коли сторонній сервіс сам надсилає потрібні дані іншому сервісу в реальному часі) або Polling (метод отримання даних від сервера, при якому програма регулярно надсилає запити до стороннього сервіса).

- запит передається через OpenAI API для генерації відповіді.

- відповідь повертається в Telegram у вигляді текстового або голосового повідомлення.

Така система дозволяє швидко реалізувати інтелектуального помічника без необхідності тренування власної моделі, що особливо актуально для невеликих компаній чи як в моєму прикладі атестаційної роботи [28]. У інфокомунікаційних системах такий телеграм бот може виконувати наступні функції:

- обробка типових запитів – відповіді на питання про тарифи, оплату, блокування SIM-карт тощо.

- аналіз наміру користувача – виявлення, чи це скарга, запит на допомогу, технічна проблема чи рекламація.

- переклад повідомлень – для міжнародних компаній дозволяє взаємодіяти з клієнтами різними мовами.

- формування заявок – автоматичне відправлення запиту до внутрішньої CRM-системи оператора.

- надання рекомендацій – на основі аналізу запиту, бот може порадити користувачеві певну послугу чи варіант вирішення проблеми.

- робота 24/7 – бот доступний у будь-який час (якщо працює сервер на якому встановлений бот), що значно поліпшує досвід користувача.

Ці всі функції дуже допомагають компаніям в реалізації своїх продуктів але є низка преваг та недоліків, з яких до переваг відносяться швидке впровадження, автоматична обробка запитів, висока доступність, багатомовність та простота в тестуванні але є значні недоліки та обмеження, а саме залежність від OpenAI API, вартість використання, конфіденційність даних, некоректні відповіді та затримка в цих відповідях. У процесі виконання роботи виникає проблема, що пов’язана з вартістю використання ChatGPT. Для його повноцінного застосування необхідно

придбати токени, що може стати суттєвою статтею витрат. Проте для великої компанії, яка здатна дозволити собі такі інвестиції, впровадження ChatGPT може суттєво оптимізувати витрати на технічну підтримку – аж до 50%.

Навіть у Україні деякі телекомунікаційні компанії почали використовувати Telegram-ботів для автоматизації підтримки. Так зокрема:

- Київстар – інтегрував бота для оформлення послуг та активації номерів;
- Vodafone Україна – використовувала бота для пошуку найближчих точок обслуговування;
- Lifecell – запустила бота для контролю стану рахунка, поповнення балансу та відстеження трафіку.

Ці приклади демонструють, що Telegram-боти з використанням нейронної мережі ChatGPT мають практичне застосування навіть в українських умовах, особливо для компаній, які хочуть швидко та бюджетно запустити інтелектуальну підтримку.

4 РОЗРОБКА ДОДАТКУ ПІДТРИМКИ КОРИСТУВАЧІВ ПРОВАЙДЕРОМ INTERNET З ВИКОРИСТАННЯМ CHATGPT

У наш час інфокомунікаційні технології швидко розвиваються, а їх користувачі очікують зручних та ефективних способів взаємодії з мережними провайдерами. Особливо це стосується взаємодії користувачів з провайдерами Internet щодо надання останніми сервісної техпідтримки. Треба зазначити, що традиційні підходи на основі реалізації класичних web-сайтів провайдерів Internet, хоча залишаються популярними, але їх функціоналу часто не вистачає для забезпечення всебічної та швидкої підтримки користувачів. Особливо це проявляється в аспекті забезпечення персоналізованого обслуговування в режимі реального часу. Цю проблему можуть вирішити конвергенція сучасних інфокомунікаційних технології, що спрямовані на підтримку концептуальних принципів мереж наступних поколінь на основі технологій присутності (таких як, наприклад, технологій на основі сервісних мультимедійних IP орієнтованих підсистем (IP Multimedia Subsystem, IMS), та технологій нейронних мереж. Як варіант такої конвергенції можна запропонувати створення Telegram-ботів, що дозволяють автоматизувати процес взаємодії між клієнтами та компанією.

У цьому розділі надаються підходи щодо розробки Telegram-бота, який має функціонувати як особистий кабінет провайдера Internet в меседжері Telegram із використанням технологій штучного інтелекту. Також проводиться аналіз основних принципів його створення у процесі проведення розробки та впровадження у Telegram меседжер. Технічним завданням була поставлена задача, щоб бот був простим у використанні, але з максимальною функціональністю. Зокрема, при розробці, акценти ставилися на реалізацію можливостей самостійної перевірки балансу користувачем, тестування швидкості Internet-з’єднань, стану обладнання, зміни персональних даних та технічної підтримки із застосуванням технологій AI. Особлива увага у процесі розробки була приділена автоматизації стандартних процесів та можливостям щодо зменшення навантаження на операторів технічної підтримки.

У процесі розробки передбачається інтеграція Telegram-бота з базою даних SQLite для зберігання даних користувачів, ведення журналів логів для всіх дій та запитів користувачів, а також дій провайдера, щодо вирішення проблем користувача.

Таким чином, такий Telegram-бот дозволяє спростити взаємодію між користувачами та провайдером, зменшує навантаження на операторів-консультантів з питань технічної підтримки, автоматизувати процеси надання відповідей на запити користувачів, тощо.

4.1 Архітектура та модулі додатку

Telegram-бот побудований на мові Python із використанням сучасних бібліотек, таких як [28]:

- `python-telegram-bot`, `telegram` та `telegram.ext` – для взаємодії з Telegram API та додання деякої логіки бота;
- `sqlite3` – для локального зберігання даних користувачів;
- `speedtest-cli` – для тестування швидкості інтернету;
- `openai`– для інтеграції з моделлю ChatGPT.

Архітектура додатка передбачає чітке розділення на модулі, що забезпечує легкість редагування та розширення програмного коду. До його функцій входять наступні (рис. 4.1):

- реєстрація користувача – користувач проходить послідовну реєстрацію, вводячи номер телефону, адресу, номер договору;

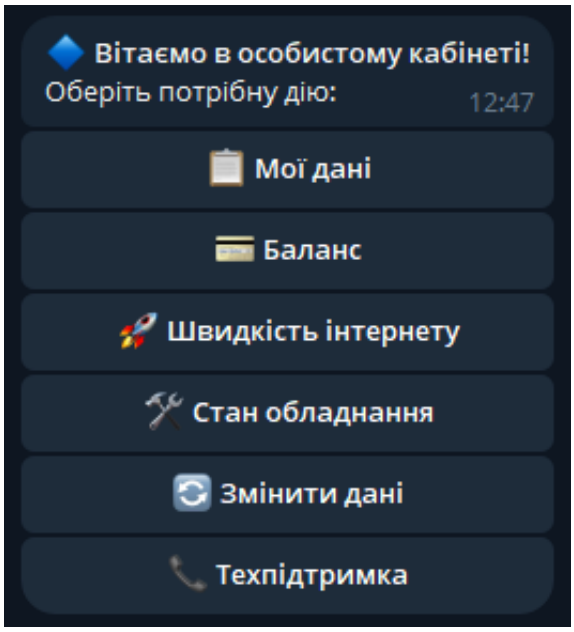

Рисунок 4.1 – Вигляд головного меню

- перевірка балансу та поповнення – клієнти можуть контролювати свій баланс та поповнювати його на певні суми (50, 100, 200, 500 грн);

- тест швидкості Internet – бот виконує відповідний тест з виводом результатів щодо швидкостей upload, download та часових затримок у процесі виконання команди `ping`;

- стан обладнання – система відображає стабільність сигналу до абонента та від нього, що дозволяє виявити можливі проблеми з обладнанням без необхідності викликів майстрів;

- зміна даних – користувач може оновити особисту інформацію: телефон, адресу, номер договору;

- технічна підтримка з використанням AI, що дозволяє надавати швидкі та точні відповіді на типові питання від користувачів.

Далі біль детально проаналізуємо процес створення і функціонування кожного модуля.

4.2 Опис модулів бота

У початковій частині коду (рис 4.2) здійснюється імпорт бібліотек та ініціалізація ключових змінних:

- підключається токен Telegram-бота;

- встановлюється логування подій;

- налаштовується підключення до OpenAI API.

База даних реалізована за допомогою sqlite3. Ініціалізація бази `init_db` створює дві таблиці: `users` та `logs` (рис. 4.3). Перша зберігає облікові дані користувачів (телефон, адреса, договір, тариф, баланс), а друга – журнал дій (`logs`) користувачів.

Основна логіка взаємодії реалізується через кнопки під повідомленням, що вводяться за допомогою функції `InlineKeyboard`. Залежно від натиснутої кнопки бот виконує відповідну дію: перевірку балансу, перегляд даних, тест швидкості, звернення до підтримки та інші.

Однією з ключових функціональних особливостей створеного Telegram-бота є можливість обробки звернень користувачів за допомогою штучного

інтелекту. Для цього використано API від компанії OpenAI, а саме мовну модель GPT-4, яка дозволяє генерувати осмислені, контекстно-залежні відповіді.

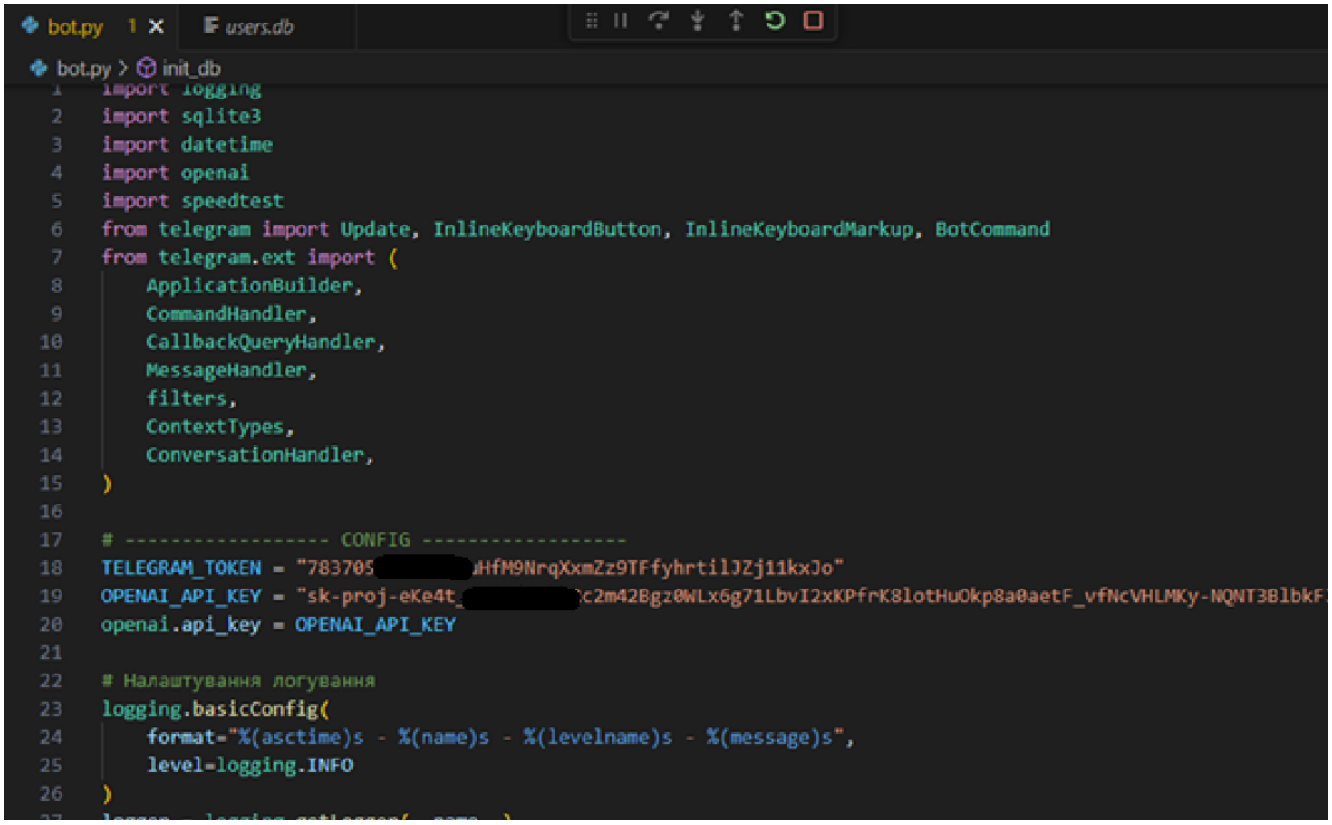

Рисунок 4.2 – Конфігурування бота

```
# ----------------- DB -----------------
def init_db():
    conn = sqlite3.connect("users.db")
    cursor = conn.cursor()
    cursor.execute(
        """CREATE TABLE IF NOT EXISTS users (
            user_id INTEGER PRIMARY KEY,
            phone TEXT,
            address TEXT,
            contract TEXT,
            tariff TEXT,
            balance REAL DEFAULT 150.0
        )"""
    )
    cursor.execute(
        """CREATE TABLE IF NOT EXISTS logs (
            user_id INTEGER,
            action TEXT,
            time TEXT
        )"""
    )
    conn.commit()
    conn.close()
```

Рисунок 4.3 – Створення та підключення бази даних

Запити формуються у вигляді повідомлення користувача , яке надсилається до API через функцію `openai.ChatCompletion.acreate` (рис. 4.4). Відповідь обробляється асинхронно, тобто можна задавати декілька питань, що будуть оброблятись паралельно, саме це забезпечує кращу масштабованість бота.

```python
# ------------------ GPT ------------------
async def ask_chatgpt(prompt):
    try:
        response = await openai.ChatCompletion.acreate(
            model="gpt-4",
            messages=[{"role": "user", "content": prompt}]
        )
        return response.choices[0].message.content.strip()
    except Exception as e:
        logger.error(f"Помилка у запиті до ChatGPT: {e}")
        return "Вибачте, сталася помилка при обробці вашого запиту. Спробуйте ще раз пізніше."
```

Рисунок 4.4 – Реалізація підключення до ChatGPT

Функція `run_speedtest` (рис. 4.5) виконує вимірювання швидкості з'єднання за допомогою бібліотеки `speedtest`. Результати (`download`, `upload`, `ping`) повертаються у простому для розуміння форматі.

```python
# ------------------ SPEEDTEST ------------------
async def run_speedtest():
    try:
        st = speedtest.Speedtest()
        st.get_best_server()
        st.download()
        # Виконуємо тест швидкості
        download_speed = st.download() / 1_000_000  # Конвертуємо в Мбіт/с
        upload_speed = st.upload() / 1_000_000      # Конвертуємо в Мбіт/с
        ping = st.results.ping

        return {
            'download': round(download_speed, 2),
            'upload': round(upload_speed, 2),
            'ping': round(ping, 2)
        }
    except Exception as e:
        logger.error(f"Speedtest error: {e}")
        return None
```

Рисунок 4.5 – Реалізація перевірки швидкості інтернету

Меню команд ініціалізується у функції `set_commands_menu` (рис. 4.6). Серед доступних команд: `/start`, `/register`, `/topup`, `/help`. Команда `/start` відкриває інтерфейс з кнопками для основних дій користувача.

```
# ------------------ STATES ------------------

REGISTER_PHONE, REGISTER_ADDRESS_TYPE, REGISTER_ADDRESS, REGISTER_CONTRACT = range(4)
CHANGE_PHONE, CHANGE_ADDRESS, CHANGE_CONTRACT = range(3, 6)

# ------------------ COMMANDS MENU ------------------
async def set_commands_menu(application):
    commands = [
        BotCommand("start", "Головне меню"),
        BotCommand("register", "Реєстрація нового акаунту"),
        BotCommand("topup", "Поповнення балансу"),
        BotCommand("help", "Довідка по боту")
    ]
    await application.bot.set_my_commands(commands)
```

Рисунок 4.6 – Реалізація блоку стану діалогу та головного меню

За покрокову реєстрацію користувача відповідають функції `register`, `register_phone`, `register_address`, `register_contract` (рис. 4.6). Валідація введених даних здійснюється за допомогою стандартних запитів, на які має дати відповідь користувач.

Функції `change_data_menu`, `change_phone_start`, `change_address_start`, `change_contract_start` ініціалізують діалог зміни окремих даних (рис.4.7). Оновлення значень у базі реалізовано через функцію `save_changes`.

Функція `top_up_balance` дозволяє вибрати суму для поповнення. У великій компанії це могло б включати прив’язку банківського рахунку, створення гіперпосилань для оплати та інші можливості. Проте в моєму проєкті це не потрібно, тому поповнення реалізовано просто як оновлення значення балансу в базі даних. Оскільки ця частина досить проста, вважаю, що можна не показувати її код.

```python
# ------------------ CHANGE DATA ------------------
async def change_data_menu(update: Update, context: ContextTypes.DEFAULT_TYPE):
    query = update.callback_query
    await query.answer()

    keyboard = [
        [InlineKeyboardButton("📱 Змінити телефон", callback_data="change_phone")],
        [InlineKeyboardButton("🏠 Змінити адресу", callback_data="change_address")],
        [InlineKeyboardButton("📝 Змінити договір", callback_data="change_contract")],
        [InlineKeyboardButton("🔙 На головну", callback_data="back_to_main")],
    ]
    reply_markup = InlineKeyboardMarkup(keyboard)

    await query.edit_message_text(
        text="🔷 <b>Оберіть, що бажаєте змінити:</b>",
        reply_markup=reply_markup,
        parse_mode="HTML"
    )
```

Рисунок 4.7 – Реалізація зміни даних

Функція `handle_message` (рис.4.8) виконує фільтрацію текстових повідомлень. Якщо повідомлення не містить ключових слів із переліку дозволених то, бот не надає відповіді. У разі валідного звернення підтягується інформація про клієнта з бази даних та генерується відповідь через ChatGPT.

```python
# ------------------ MESSAGE HANDLER ------------------
def is_valid_topic(message: str) -> bool:
    ALLOWED_TOPICS = [
        "інтернет", "техпідтримка", "тариф", "оплата", "зв'язок",
        "швидкість", "підключення", "відключення", "доступ", "адреса",
        "модем", "роутер", "компанія", "послуга", "обладнання"
    ]
    message_lower = message.lower()
    return any(topic in message_lower for topic in ALLOWED_TOPICS)

# 🔹 Основна обробка повідомлень
async def handle_message(update: Update, context: ContextTypes.DEFAULT_TYPE):
    user_id = update.effective_user.id
    message_text = update.message.text

    # Якщо користувач у режимі підтримки
    if context.user_data.get("awaiting_support", False):
        # Перевірка тематики повідомлення
        if not is_valid_topic(message_text):
            await update.message.reply_text(
                "❌ Вибачте, я можу відповідати лише на питання, що стосуються технічної підтримки або нашої компанії.",
                parse_mode="HTML"
            )
            return
```

Рисунок 4.8 – Реалізація фільтрації питання

У блоці `if __name__ == __main__` (рис. 4.9) запускається бот: ініціалізується база, вмикається Telegram-додаток, додаються обробники команд і повідомлень, виконується запуск через метод `run_pollin`.

```
# ------------------ MAIN ------------------
if __name__ == "__main__":
    init_db()

    app = ApplicationBuilder().token(TELEGRAM_TOKEN).build()
```

Рисунок 4.9 – Функція старту бота

4.3 Функціонування Telegram-боту

Після запуску Telegram-бота користувач взаємодіє з ним через простий інтерфейс у чаті. Першим кроком є натискання кнопки `Start`, що ініціює привітальне меню та пропозицією вибрати одну з доступних опцій (рис. 4.1).

Функція реєстрації реалізована як послідовний діалог із перевірками введених даних на кожному етапі. Користувач може ініціювати реєстрацію через команду /register або відповідну кнопку меню. У разі помилки на будь-якому етапі можна повернутися на головний екран натиснувши кнопку «На головну». На першому етапі бот просить ввести номер телефону у форматі `+380XXXXXXXXX` (рис.4.10). Якщо формат некоректний, з’являється відповідне повідомлення з поясненням. Після правильного введення номера користувач обирає тип житла – «Квартира» або «Приватний сектор». Цей вибір визначає, які саме варіанти адреси будуть доступними надалі (при виборі пункту «Квартира» з’являється вибір номеру квартири) (рис. 4.11).

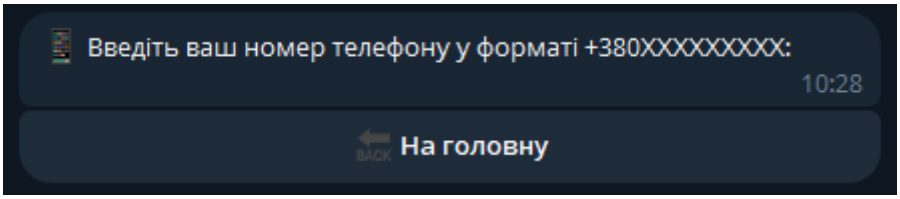

Рисунок 4.10 – Реєстрація номер телефону

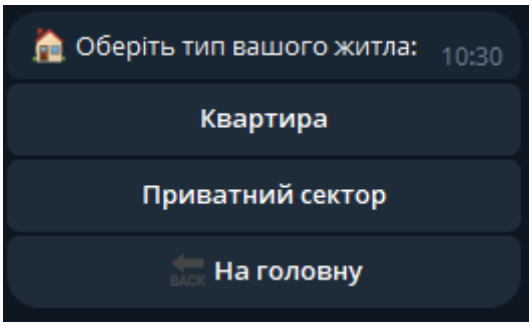

Рисунок 4.11 - Реєстрація адреса проживання

Далі бот запитує повну адресу. Очікується формат «місто, вулиця, будинок або квартира» (рис. 4.12).

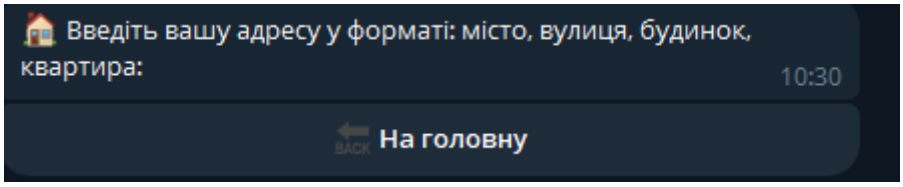

Рисунок 4.12 – Реєстрація продовження адреси проживання

На завершальному етапі користувач вводить номер договору. Після цього бот записує всю зібрану інформацію до бази даних `users.db` (рис. 4.13) та створює лог у таблиці `logs` (рис. 4.14).

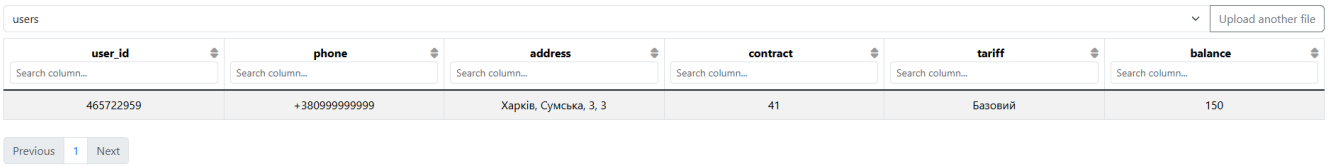

Рисунок 4.13 – Вигляд бази даних

Користувач бачить повідомлення з текстом «Оберіть потрібну дію», під яким розміщено шість інтерактивних кнопок, кожна з яких виконує окрему функцію (див. рис. 4.1):

- «Мої дані» – відображення контактної інформації, адреси та тарифу;

user_id	action	time
	Search column...	Search column...
465722959	check_account	2025-07-08 00:06:15.871357
465722959	check_account	2025-07-08 00:06:35.894044
465722959	register	2025-07-08 00:07:02.175184
465722959	check_account	2025-07-08 00:07:25.073414
465722959	check_equipment	2025-07-08 00:07:37.997695
465722959	check_speed	2025-07-08 00:08:35.506622
465722959	contact_support	2025-07-08 00:09:05.172311
465722959	check_balance	2025-07-08 00:11:04.694165
465722959	topup_50.0	2025-07-08 00:11:20.123863
465722959	topup_50	2025-07-08 00:11:20.199565

Рисунок 4.14 – Вигляд журналу логів

- «Баланс» – перевірка поточного балансу користувача з можливістю його поповнення;

- «Швидкість інтернету» – змодельоване тестування швидкості підключення;

- «Стан обладнання» – повідомлення про стабільність сигналу від і до абонента;

- «Змінити дані» – перехід до оновлення номеру телефону або інших полів;

- «Техпідтримка» – звернення до асистента ChatGPT з описом проблеми.

При виборі кнопки «Мої дані», бот відображає інформацію з бази даних: номер телефону, адресу та назву тарифного плану (рис. 4.15). Якщо дані відсутні, пропонується зареєструватися.

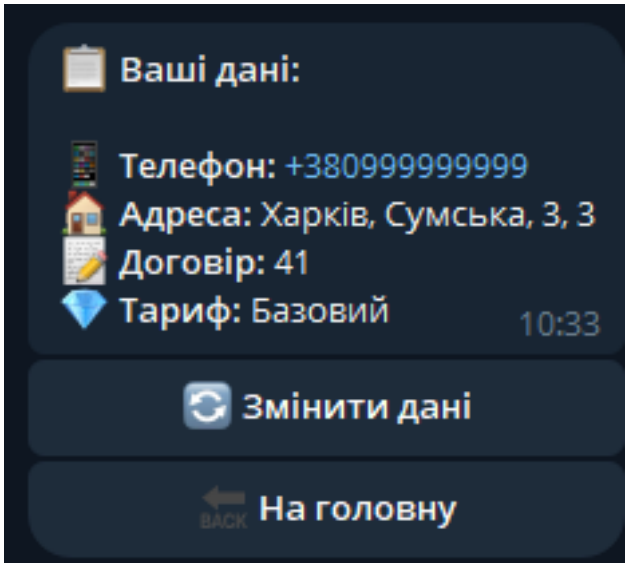

Рисунок 4.15 – Вигляд кнопки «Мої дані»

Опція «Швидкість інтернету» надає у відповідь змодельовані дані, а саме середні показники швидкості download та upload які надаються сервісом Speedtest (рис. 4.16 та 4.17).

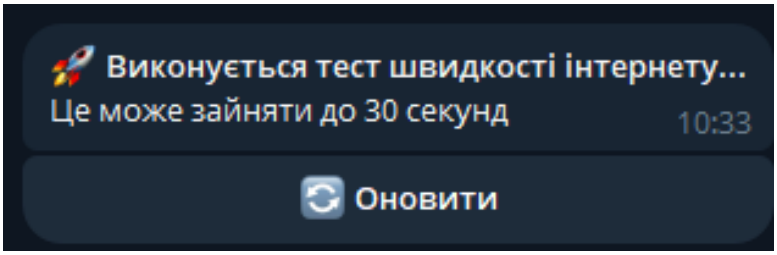

Рисунок 4.16 – Вигляд очікування на тест швидкості

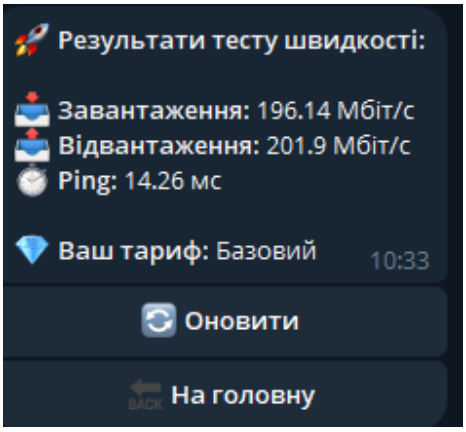

Рисунок 4.17 – Вигляд тесту швидкості

Натискання «Стан обладнання» генерує відповідь щодо стабільності зв'язку, а саме бот повідомляє, що сигнал від і до абонента нормальний або вказує на можливі збої (рис. 4.18).

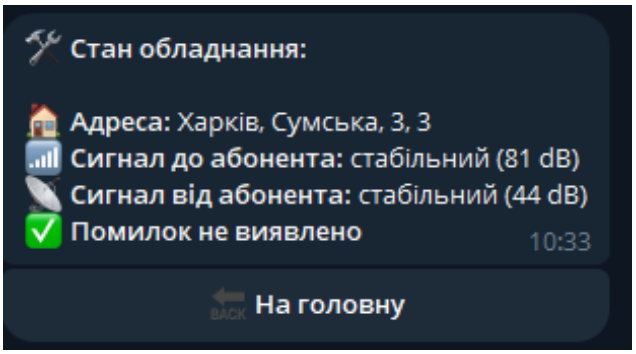

Рисунок 4.18 – Вигляд стану обладнання

При виборі «Змінити дані», користувач переходить до режиму оновлення особистої інформації, зокрема телефону чи адреси, без повторної реєстрації (рис. 4.19).

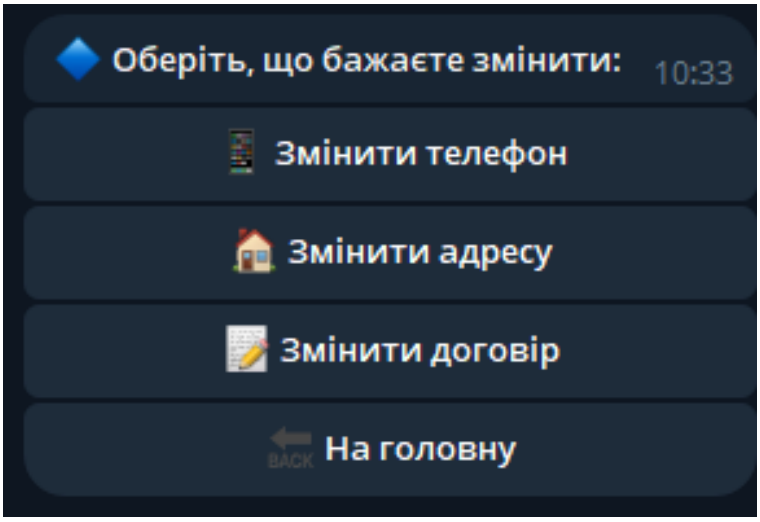

Рисунок 4.19 – Вигляд зміни даних

При натисканні кожної з кнопки буде з’являтись меню зі зміною даних як в реєстрації.

Кнопка «Техпідтримка» активує режим вільного введення. Користувач описує проблему, якщо користувач вводить запит який не стосується питання технічної підтримки або компанії, то йому виводиться попереджувальне вікно з тим, що бот не відповідає на такі питання (рис. 4.20), якщо ж питання задовольняє вимоги то бот передає цей опис до ChatGPT, який повертає відповідь технічного спеціаліста – зрозумілу, логічну та адаптовану до ситуації (рис. 4.21).

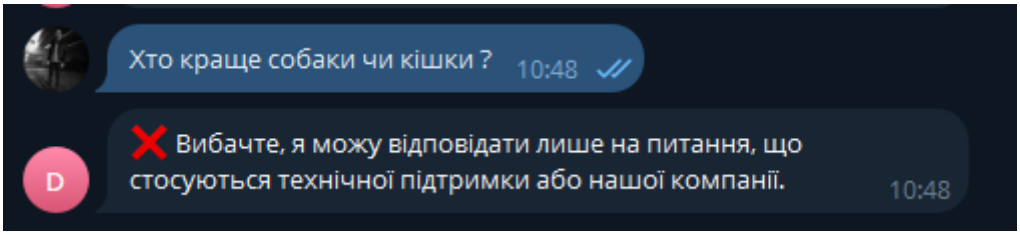

Рисунок 4.20 – Приклад невірного питання

Якщо користувачу не потрібна більше підтримка, то він може натиснути на кнопку «Завершити діалог» і бот завершить його з повідомленням (рис 4.22).

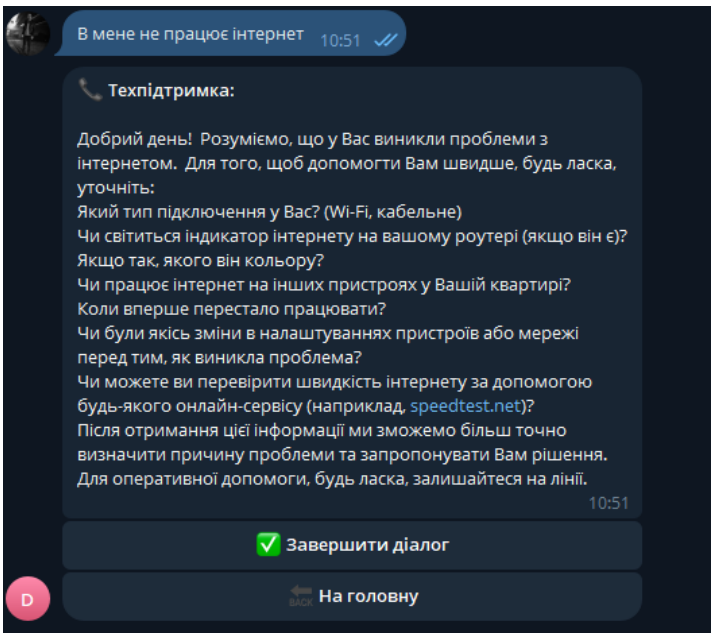

Рисунок 4.21 – Приклад відповіді на правильне питання.

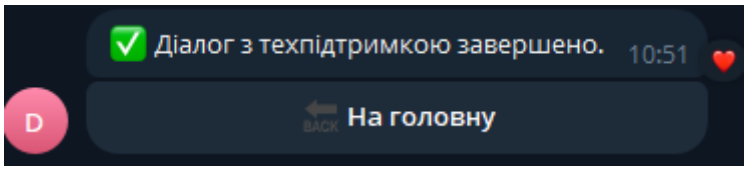

Рисунок 4.22 – Завершення діалогу

Як можна бачити, створений Telegram-бот виконує функції повноцінного цифрового помічника у сфері технічної підтримки користувачів телекомунікаційних послуг. Його інтерфейс реалізовано у формі інтуїтивно зрозумілого меню з шістьма ключовими опціями, що охоплюють як отримання інформації (баланс, дані, обладнання), так і взаємодію з оператором через інтеграцію з ChatGPT.

Завдяки використанню сучасної мовної моделі ChatGPT, бот не просто виконує статичні команди, а здатен гнучко реагувати на індивідуальні запити, пропонуючи логічні, змістовні та корисні відповіді. Такий підхід значно підвищує ефективність автоматизованої підтримки, зменшує навантаження на операторів і покращує користувацький досвід загалом.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досліджено сучасні підходи до застосування штучного інтелекту, зокрема нейронних мереж, у сфері інфокомунікацій. Проведено комплексний аналіз історичного розвитку, технологічних принципів реалізації та архітектурних різновидів штучних нейронних мереж. Окрему увагу приділено питанням машинного навчання, типам нейромереж (CNN, RNN, GAN і трансформери) та їх особливостям функціонування.

На основі вивчення сучасних рішень у галузі інфокомунікацій (NVIDIA Aerial, DeepSig, ChatGPT) доведено, що нейронні мережі відіграють ключову роль у вдосконаленні телекомунікаційних систем. Вони забезпечують покращення QoS, автоматизацію процесів управління мережею, обробку трафіку та інтерактивну взаємодію з користувачем.

Проведене дослідження, що підтверджує актуальність та перспективність використання нейронних мереж у сучасних інфокомунікаційних технологіях. Робота дозволила узагальнити науково-технічні підходи щодо впровадження ШІ в галузі, а також продемонструвала практичні навички здобувача у моделюванні, розгляді прототипів та аналізі інтелектуальних систем.

У четвертому розділі було реалізовано приклад практичного застосування сучасних інформаційних технологій у сфері інфокомунікацій – створено Telegram-бота, який виконує функції цифрового помічника для користувачів інфокомунікаційних послуг. Система включає взаємодію з користувачем, обробку запитів щодо балансу, стану обладнання, швидкості інтернету, зміни особистих даних, а також – надсилання звернень до технічної підтримки із залученням штучного інтелекту (моделі GPT-4 від OpenAI).

У процесі реалізації проєкту було застосовано сучасні бібліотеки Python, серед яких `sqlite3`, `telegram.bot`, `openai`, `speedtest`, а також архітектурно правильно організовану структуру бази даних з таблицями `users` і `logs`. Особливу цінність має можливість асинхронного оброблення запитів до мовної моделі, що забезпечує стабільну роботу системи навіть при значному навантаженні.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Hurbans R. Grokking AI Algorithms: Understand and apply the core algorithms of deep learning and artificial intelligence in this friendly illustrated guide / R. Hurbans – Shelter Island: Manning Publications, 2020. – 376 с.

2. Історія штучного інтелекту: від 1950-х до сьогодні [Електронний ресурс] // FreeCodeCamp. – 2023. – Режим доступу: https://www.freecodecamp.org/ukrainian/news/istoriya-shtuchnoho-intelektu-vid-1950-kh-do-sohodni/.

3. CerboAI's Guide: Understanding CNN, RNN, GAN, Transformer and other architectures [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: https://medium.com/@CerboAI/cerboais-guide-understanding-cnn-rnn-gan-transformer-and-other-architectures-2ded10988eee.

4. WikiDocs. Sequence-to-Sequence Learning with Attention [Електронний ресурс]. – Доступ здійснено 28.05.2025. – Режим доступу до ресурсу: https://wikidocs.net/202289.

5. Акіменко І.С. Архітектура нейронних мереж [Електронний ресурс]. / І. Акіменко – 2020. – Режим доступу до ресурсу: https://csc.knu.ua/media/study/asp/art_net_group_inf_akimenko/lecture/lec1.pdf.

6. Основи нейронних мереж [Електронний ресурс] // Запорізький національний університет 2023. –. – Режим доступу до ресурсу: https://files.znu.edu.ua/files/Bibliobooks/Inshi78/0058677.pdf.

7. Schulman J. et al. Proximal Policy Optimization Algorithms [Електронний ресурс] / J.Schulman // NeurIPS, 2017. – Режим доступу до ресурсу: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

8. NVIDIA. Aerial Omniverse Digital Twin [Електронний ресурс]. – Доступ здійснено 04.06.2025. – Режим доступу до ресурсу: https://docs.nvidia.com/aerial/aerial-dt/text/overview.html#aerial-omniverse-digital-twin.

9. NVIDIA. CUDA-Accelerated RAN Features for 5G gNB [Електронний ресурс]. – Доступ здійснено 08.06.2025. – Режим доступу до ресурсу: https://docs.nvidia.com/aerial/cuda-accelerated-ran/aerial_cubb/product_brief/features_for_5g_gnb.html.

10. NVIDIA. RAN Digital Twin [Електронний ресурс]. – Доступ здійснено 08.06.2025. – Режим доступу до ресурсу: https://docs.nvidia.com/aerial/aerial-dt/text/ran_digital_twin.html.

11. What is Intel FlexRAN? [Електронний ресурс] // Trenton Systems – 2022. – Режим доступу до ресурсу: https://www.trentonsystems.com/en-us/resource-hub/blog/what-is-intel-flexran.

12. DeepSig. AI-Based RF Awareness for Private Wireless Networks [Електронний ресурс]. – Доступ здійснено 10.06.2025. – Режим доступу до ресурсу: https://www.deepsig.ai/download-ai-based-rf-awareness-for-private-wireless-networks-white-paper/.

13. DeepSig. Amplifying 5G vRAN Performance with AI & Deep Learning [Електронний ресурс]. – Доступ здійснено 11.06.2025. – Режим доступу до ресурсу: https://www.deepsig.ai/download-amplifying-5g-vran-performance-with-ai-deep-learning-white-paper/.

14. Vaswani A. et al. Attention Is All You Need. [Електронний ресурс] / A Vaswani – 2017. – Режим доступу до ресурсу: https://arxiv.org/pdf/1706.03762.

15. OpenAI. GPT-4 Technical Report. – [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://openai.com/research/gpt-4.

16. Dey J. A past, present, and future of attention [Електронний ресурс] / J. Dey // Medium. – 2021. – Режим доступу до ресурсу: https://joshdey.medium.com/a-pastpresent-and-future-of-attention-f6e269574a5.

17. OpenAI. InstructGPT: Aligning language models with human intent [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://openai.com/research/instruction-following.

18. NVIDIA. Aerial SDK Platform [Електронний ресурс]. – Доступ здійснено 14.06.2025. – Режим доступу до ресурсу: https://docs.nvidia.com/aerial/archive/aerial-sdk/22-4/text/product_brief/aerial_sdk_platform.html.

19. Verizon hopes NVIDIA will give it the edge in private 5G [Електронний ресурс] // Telecoms.com. – Доступ здійснено 18.06.2025. – Режим доступу до ресурсу: https://www.telecoms.com/5g-6g/verizon-hopes-nvidia-will-give-it-the-edge-in-private-5g.

20. Keysight and Samsung advance AI for RAN based on the NVIDIA AI Aerial Platform [Електронний ресурс] // Keysight. – 2025. – Режим доступу до ресурсу: https://www.keysight.com/us/en/about/newsroom/news-releases/2025/0304-pr25-054-keysight-and-samsung-advance-ai-for-ran-based-on-the-nvidia-ai-aerial-platform.html.

21. Deploy AI-RAN at Cell Sites with NVIDIA ARC Compact [Електронний ресурс] // NVIDIA 2025. – Режим доступу до ресурсу: https://developer.nvidia.com/blog/deploy-ai-ran-at-cell-sites-with-nvidia-arc-compact/.

22. DeepSig White Papers [Електронний ресурс]. – Доступ здійснено 20.06.2025. – Режим доступу до ресурсу: https://www.deepsig.ai/white-papers/

23. Will Enhanced Automation Help Open RAN Find an Audience? [Електронний ресурс] // DevelopingTelecoms. – Доступ здійснено 22.06.2025. – Режим доступу до ресурсу: https://developingtelecoms.com/index.php?option=com_content&view=article&id=17282:will-enhanced-automation-help-open-ran-find-an-audience&catid=33&acm=_1189.

24. Industry Leaders in AI and Wireless Form AI-RAN Alliance [Електронний ресурс] // DeepSig 2024. – Режим доступу до ресурсу: https://www.deepsig.ai/industry-leaders-in-ai-and-wireless-form-ai-ran-alliance/.

25. Orange France, Samsung establish vRAN Open RAN first pilot [Електронний ресурс] // ComputerWeekly. – 2025. – Режим доступу до ресурсу: https://www.computerweekly.com/news/366627115/Orange-France-Samsung-establish-vRAN-Open-RAN-first-pilot.

26. Marr B. 10 Amazing Real-World Examples of How Companies Are Using ChatGPT in 2023 [Електронний ресурс] / B.Marr // Forbes. – 2023. – Режим доступу до ресурсу: https://www.forbes.com/sites/bernardmarr/2023/05/30/10-amazing-real-world-examples-of-how-companies-are-using-chatgpt-in-2023/.

27. Zendesk ChatGPT Integration [Електронний ресурс] // Zendesk Marketplace. – Доступ здійснено 28.06.2025. – Режим доступу до ресурсу: https://www.zendesk.com/marketplace/apps/support/960060/triggerschatgpt/.

28. Telegram Bot API Documentation [Електронний ресурс] // Telegram. – Доступ здійснено 30.06.2025. – Режим доступу до ресурсу: https://core.telegram.org/bots/api.