Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Спеціальність 125 «Кібербезпека»

Освітня програма «Кібербезпека»

«Допущено до захисту»
В. о. завідувача кафедрою БІСТ
Мелкозьорова О. М.

_____

« » 2024 р.

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему: «Аналіз та дослідження засобів забезпечення безпеки у системах штучного інтелекту»

оцінка « » Керівник к.т.н. Єсіна М. В.

Голова ЕК

Лемешко О. В. _____ Рецензент к.т.н. Бобух В. А.

Виконавець: студентка групи КБ-42

_____ Кобилянська О. А.

Харків – 2024

РЕФЕРАТ

Обсяг пояснювальної записки до проєкту бакалавра складає 40 сторінок, у тому числі 17 рисунків, 2 таблиці, 5 формул і 6 лістингів. Використано 13 джерел за переліком посилань.

Метою даної роботи є дослідження методів атак на нейронні мережі та розробка ефективних методів захисту від цих атак. Включаючи аналіз існуючих типів атак, розробку адверсивних прикладів, а також впровадження методів захисту, таких як адверсивне навчання, моніторинг та інші стратегії. У роботі використані аналітичні й порівняльні методи аналізу, а також експериментальні методи для генерації адверсивних прикладів з використанням градієнтних методів.

У результаті проведеного дослідження було визначено, що адверсивні атаки можуть суттєво впливати на точність нейронних мереж. Експериментально підтверджено ефективність адверсивного навчання та бінарного порогового перетворення як методів захисту від таких атак. Навчання моделей на адверсивних прикладах дозволяє значно підвищити їх стійкість до атак. Отримані результати можуть бути корисні для фахівців з кібербезпеки, дослідників у галузі штучного інтелекту, освітніх установ, компаній, що займаються обробкою великих даних, та державних організацій. Вони допоможуть у розробці та впровадженні ефективних методів захисту нейронних мереж від адверсивних атак, підвищенні надійності та безпеки моделей, а також у навчанні студентів сучасним методам кібербезпеки.

Рекомендації, розроблені на основі отриманих результатів, сприятимуть підвищенню рівня кібербезпеки і допоможуть створювати більш надійні та безпечні системи на основі штучного інтелекту, що зменшить ризики, пов'язані з кіберзагрозами.

Ключові слова: КІБЕРБЕЗПЕКА, ШТУЧНИЙ ІНТЕЛЕКТ, АДВЕРСИВНІ АТАКИ, НЕЙРОННІ МЕРЕЖІ, ЗАХИСТ НЕЙРОННИХ МЕРЕЖ.

ABSTRACT

The explanatory note for the bachelor's project comprises 40 pages, including 17 figures, 2 tables, 5 formulas, and 6 listings. A total of 13 sources are cited in the references list.

The aim of this work is to investigate attack methods on neural networks and to develop effective defense methods against these attacks. This includes the analysis of existing types of attacks, the development of adversarial examples, and the implementation of defense methods such as adversarial training, monitoring, and other strategies. Analytical and comparative analysis methods, as well as experimental methods for generating adversarial examples using gradient methods, were used in this work.

As a result of the research, it was determined that adversarial attacks can significantly impact the accuracy of neural networks. The effectiveness of adversarial training and binary threshold conversion as defense methods against such attacks was experimentally confirmed. Training models on adversarial examples significantly enhances their robustness against attacks. The obtained results can be useful for cybersecurity specialists, artificial intelligence researchers, educational institutions, companies involved in big data processing, and government organizations. They will aid in the development and implementation of effective defense methods for neural networks against adversarial attacks, enhance the reliability and security of models, and help educate students on modern cybersecurity methods.

Recommendations based on the obtained results will contribute to the improvement of cybersecurity levels and help create more reliable and secure artificial intelligence systems, thereby reducing risks associated with cyber threats.

Keywords: CYBERSECURITY, ARTIFICIAL INTELLIGENCE, ADVERSARIAL ATTACKS, NEURAL NETWORKS, NEURAL NETWORK DEFENSE.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

CNN	– Convolutional Neural Network
DNN	– Deep Neural Network
FGSM	– Fast Gradient Sign Method
FFNN	– Feed-Forward Neural Network
GAN	– Generative Adversarial Network
GPU	– Graphics Processing Unit
NLP	– Natural Language Processing
PGD	– Projected Gradient Descent
RNN	– Recurrent Neural Network
НМ	– нейронна мережа
МН	– машинне навчання
США	– Сполучені Штати Америки
СШІ	– системи штучного інтелекту
ШІ	– штучний інтелект

ВСТУП

У сучасному світі цифрова трансформація стала невід'ємною складовою повсякденного життя. Завдяки штучному інтелекту, багато процесів стали більш ефективними, автоматизованими та точними. Проте, поряд з цими досягненнями виникають нові виклики, зокрема ті, що пов'язані з кібербезпекою.

Захист даних і систем постало, як критично важливе завдання. Використання великих обсягів даних, необхідних для навчання моделей штучного інтелекту, вимагає надійних механізмів захисту від несанкціонованого доступу та кібератак. Крім того, алгоритми штучного інтелекту можуть бути вразливими до маніпуляцій і втручання, що може призвести до значних негативних наслідків.

Актуальність цієї роботи обумовлена швидким розвитком технологій та збільшенням кількості кіберзагроз, спрямованих на системи, що використовують штучний інтелект. Зловмисники постійно шукають нові способи використання вразливостей для своїх цілей. Тому важливо досліджувати методи захисту нейронних мереж та розробляти ефективні стратегії для забезпечення безпеки даних і систем.

Метою даної роботи є дослідження методів атак на нейронні мережі та розробка ефективних методів захисту від цих атак. Це включає в себе аналіз існуючих типів атак, розробку адверсивних прикладів, а також впровадження методів захисту, таких як адверсивне навчання, моніторинг та інші стратегії.

Результати цієї роботи можуть бути використані в різних галузях, де застосовуються технології штучний інтелект, включаючи медицину, банківську сферу, транспорт, безпеку та багато інших. Розроблені методи захисту можуть допомогти забезпечити більш високий рівень безпеки для систем, що використовують нейронні мережі, а також зменшити ризики, пов'язані з кіберзагрозами.

Гіпотеза цієї роботи полягає в тому, що розробка та впровадження комплексних методів захисту, зокрема адверсивного навчання та моніторингу, можуть значно підвищити стійкість нейронних мереж до атак та забезпечити

більш високий рівень безпеки систем, що використовують технології штучного інтелекту.

1 ОГЛЯД СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

Двома найбільш революційними досягненнями інформаційної епохи стали можливість миттєвого обміну інформацією та спілкування по всьому світу за допомогою Інтернету, а також постійна цифрова трансформація повсякдення. Хоча ці явища значно покращили життя багатьох людей, вони також викликали проблеми зловживання і широкомасштабну експлуатацію з боку урядових та неурядових організацій. Один із прикладів такої проблематики – «золотий вік стеження», під час якого відбувається масштабний збір даних урядами та приватним сектором для внутрішнього та міжнародного моніторингу. Таке подвійне використання притаманне сучасним технологіям, і штучний інтелект не є винятком.

З супротивної точки зору, цифровізація відкрила нові можливості для організацій і злочинців, дозволяючи їм досягати своїх цілей через вдосконалення основних технологій. Це частково пов'язано з тим, що інтернет-інфраструктура і все, що було створено на її основі, спочатку не були спроєктовані з урахуванням безпеки. Машинне навчання, яке може вважатися наступним етапом у еволюції цифрової трансформації, вже інтегроване у бізнес-моделі та військові стратегії. У світлі минулих інцидентів і прогалин в інформаційній безпеці, критично важливо зосередитися на аспектах безпеки штучного інтелекту (ШІ), щоб запобігти його зловживанню і таким чином максимально реалізувати його потенціал. [1]

1.1 Типи систем штучного інтелекту

Більшість людей у сучасному світі вважають ШІ засобом генерації текстів, проте існує різноманітність систем, кожна з яких може бути класифікована за різними критеріями, зокрема за типом задач, які вони вирішують, за ступенем своєї автономності та можливостями. Наприклад, системи обробки природної мови дозволяють комп'ютерам розуміти та реагувати на людську мову, тоді як системи комп'ютерного зору використовуються для розпізнавання об'єктів і сцен з відео чи фотографій. Розуміння цих відмінностей важливе для розкриття

потенціалу ШІ в різних галузях, таких як медицина, фінанси, транспорт та безпека, де він може впроваджувати різноманітні інноваційні рішення, від діагностики хвороб до управління ризиками та автоматизації складних процесів. У цьому підрозділі буде детально розглянуто чотири ключові типи систем штучного інтелекту (СШІ) та їх хронологію (рис. 1.1): машинне навчання, обробку природної мови, комп'ютерний зір та робототехніку.

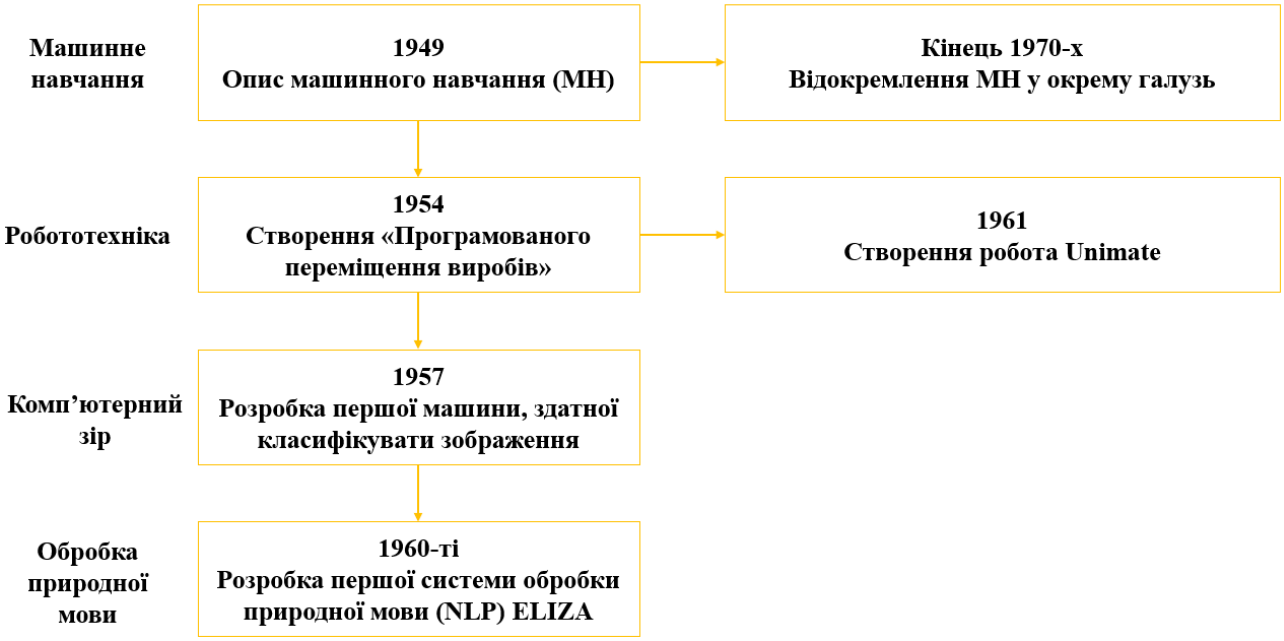


Рисунок 1.1 – Хронологічна послідовність виникнення типів СШІ

Машинне навчання – один з методів функціонування штучного інтелекту, а саме – практичної реалізації його можливостей шляхом створення алгоритмів для виявлення закономірностей під час аналізу великих даних, та їх подальше використання для самонавчання [2]. Основну мету МН можна сформулювати як «не вирішити, а навчити», тобто натренувати модель передбачати результат за вхідними даними. Практика останніх років впровадження МН показує, що чим більше різноманітних даних обробляє модель, тим швидше і простіше їй знайти закономірності і надати правильний результат.

Машинне навчання частково базується на моделі взаємодії клітин мозку. Ця модель була описана в 1949 році Дональдом Геббом в роботі "Організація поведінки". У книзі представлені теорії про збудження нейронів і зв'язок між

ними [3]. У контексті МН, "правило Гебба" часто застосовується для опису способу, у який штучні нейронні мережі (ШНМ) можуть навчатися. Згідно з цим правилом, ваги зв'язків у ШНМ можуть бути скориговані на основі досвіду. Загалом, вага між двома нейронами збільшуються, якщо активація одного нейрона призводить до успішної активації наступного. Цей механізм є основою для таких методів навчання, як зворотне поширення помилки, яке використовується для тренування багатошарових персептронів [3].

До кінця 1970-х років МН було елементом розвитку ШІ, але згодом воно відокремилось та почало розвиватися як самостійна галузь. У сучасності його використовують багатьох аспектах безпеки, таких як виявлення аномалій, протидії фішингу, прогнозуванні нових варіантів шкідливого програмного забезпечення та інше. Наразі наявні наступні типи МН (рис. 1.2).

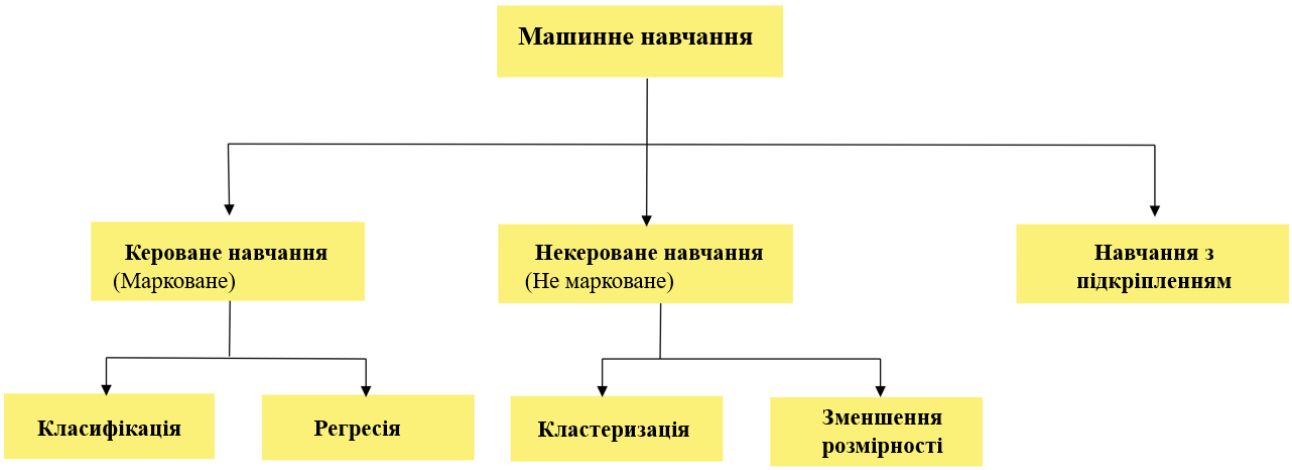


Рисунок 1.2 – Типи машинного навчання

Другим типом СШІ, що буде розглянута в цьому підрозділі виступає система обробки природної мови (NLP).

Обробка природної мови поєднує моделювання людської мови на основі правил, що також називають «обчислювальною лінгвістикою» зі статистичними моделями та МН, що дозволяє комп'ютерам і цифровим пристроям розпізнавати, розуміти, генерувати текст і мовлення [4]. Системи NLP мають за мету зробити технології більш доступними, покращуючи взаємодію між людиною та

машиною через розпізнавання голосових команд. Вони спрощують виконання рутинних завдань, як-от сортування електронних листів та автоматизація відповідей. Також, підтримка багатомовності дозволяє користувачам з різних культур ефективно спілкуватися, а людям з обмеженими можливостями активніше використовувати інтернет.

Однією із перших систем обробки природної мови була ELIZA [5]. У 1960-х роках Джозеф Вайзенбаум розробив чат-бот на основі правил, який використовували для імітації розмови з терапевтом. ELIZA використовувала зіставлення шаблонів і розпізнавання ключових слів для генерування відповідей на запитання користувача. Хоча програма не була здатна до справжнього розуміння, вона стала проривом у цій галузі та викликала інтерес серед дослідників і широкої громадськості.

Застосування технологій NLP у сфері кібербезпеки сприяє значному прогресу в різних аспектах, включаючи вдосконалення у виявленні шахрайських дій, моніторинг та аналіз контенту в соціальних мережах, а також автоматизацію процесів розпізнавання та реагування на інциденти безпеки.

Наступним важливим кроком в аналізі СШІ є огляд системи комп’ютерного зору. Галузь ШІ, яка активно використовує методи машинного навчання та нейронні мережі (НМ), фокусується на тому, щоб навчити комп'ютери та інші системи отримувати інформацію з цифрових зображень, відео та інших візуальних даних. Ключова мета полягає в інтерпретації отриманих візуальних даних і, за необхідності, у вживанні заходів у випадках виявлення відхилень або проблем.

Згідно з [6], першою машиною, яка змогла класифікувати зображення, став Перцептрон Марк 1, створений психологом-дослідником Френком Розенблаттом у 1957 році. Він черпав натхнення з біологічних нейронів, що імітують їхні функції за допомогою механічних аналогів, створюючи машину з 400 фотоелементів, здатних виявляти візуальні патерни. Конструкція Перцептрона була революційною, оскільки вона дозволяла автоматично адаптуватися та вчитися на основі отриманих даних без необхідності перепрограмування. Цей ранній прототип став фундаментом для подальших розробок у галузі НМ і

машинного зору, заклавши основу для сучасних алгоритмів глибокого навчання, які сьогодні використовуються у різноманітних застосуваннях — від автоматичного розпізнавання облич до систем безпеки.

Використання комп'ютерного зору у сфері безпеки є ключовим для поліпшення ефективності багатьох систем. Технології дозволяють здійснювати глибокий аналіз відео з камер спостереження, розпізнавати обличчя і аналізувати поведінку осіб у публічних місцях. Такий підхід застосовують для ідентифікації підозрілих або агресивних дій, що можуть загрожувати громадській безпеці. Комп'ютерний зір також залучається для автоматизації митних перевірок, де він допомагає в ідентифікації та класифікації заборонених предметів у багажі без безпосереднього втручання людини. Крім того, він є невід'ємною частиною біометричних систем контролю для верифікації особистості на основі унікальних фізичних характеристик, таких як відбитки пальців, форма обличчя, райдужка ока.

Останній із розглянутих, проте не менш важливий тип СШІ – робототехніка – галузь, що займається розробкою автоматизованих технічних систем. Мета робототехніки охоплює різноманітні сектори. В промислових виробництвах збільшення ефективності, покращення умов безпеки у потенційно ризикованих умовах, таких як висотні чи токсичні та радіоактивні середовища. Роботи також забезпечують високу точність та якість виконання завдань, що дозволяє мінімізувати помилки, пов'язані з людським фактором. У сфері безпеки та розвідки роботи використовуються для проведення космічних досліджень, океанології та археології у місцях, які є недоступними для людей.

Поворотним моментом в історії промислової робототехніки стало створення Джорджем Деволом у 1954 році «Програмованого переміщення виробів». Цей апарат став фундаментом для створення Unimate, який був запущений у 1961 році. Оснащений гідравлічним приводом, Unimate був негайно встановлений на заводі General Motors в Трентоні (США) , де його застосовували для вилучення компонентів з установки лиття під тиском. Цей робот спершу був обмежений однією функцією через труднощі з перепрограмуванням. У подальші роки, декілька інших Unimates були встановлені на автомобільних заводах, де

вони переважно використовувалися для точкового зварювання автомобілів та транспортування компонентів [7].

Узагальнюючи наведену інформацію, що стосується різних сфер застосування ШІ у минулому та сучасності, виявляється універсальне використання НМ та алгоритмів МН. Такі технології будують базове розуміння для створення комп'ютерних моделей, які емулюють процеси людського сприйняття та когнітивні функції, а також здатні адаптуватися та еволюціонувати, вдосконалюючи свої здібності на основі аналізу накопичених даних. Особливість, зазначена вище, становить ключову компоненту сучасної архітектури ШІ, що підкреслює їх значущість та вплив на розвиток інтелектуальних систем.

1.2 Типи архітектур штучних нейронних мереж

Як зазначалося раніше, ШНМ є алгоритмами, що використовуються для функціонування всіх типів ШІ. Під час обробки даних модель може робити помилки і вдосконалюватися, обчислюючи ці помилки та регулюючи ваги своїх вузлів або нейронів для компенсації. Завдяки циклу помилок та навчанню на них НМ є ефективним засобом ШІ. Кожна НМ має архітектуру, яка починається з вхідного шару і закінчується вихідним. Між ними розташовані кілька прихованих шарів, через які проходять дані. Кожен шар містить вузли або нейрони, які реагують на дані по-різному, причому кожен вузол має вагу, що впливає на його реакцію на вхідні дані (рис. 1.3).

Наприклад, у класифікаційній задачі вихідний шар може мати кількість нейронів, рівну кількості класів, з ймовірностями для кожного класу.

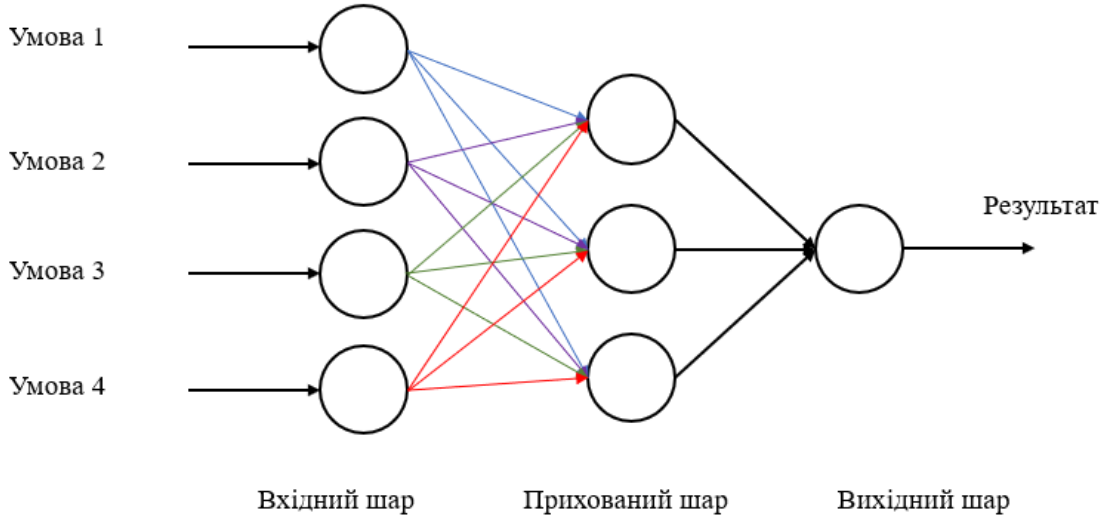


Рисунок 1.3 – Проста архітектура нейронних мереж

Загальна особливість, яка виокремлює ШНМ полягає у кількості та типах шарів, на яких будується їх архітектура. Наразі існують чотири типи, а саме нейронні мережі прямого зв'язку (FFNN), рекурентні нейронні мережі (RNN), згорткові нейронні мережі (CNN) та генеративні змагальні мережі (GAN)

FFNN є однією з основних форм НМ. Архітектура мереж побудована таким чином, що дані передаються від входу до виходу без зворотних зв'язків чи циклів. Хоча це одна з найпростіших структур НМ, приховані шари між вхідним і вихідним рівнями можуть бути досить складними. Такий тип НМ широко використовується для завдань розпізнавання шаблонів і зображень, регресійного аналізу та класифікації.

FFNN складається з вхідного шару, ряду прихованих шарів і вихідного шару. Дані надходять до мережі через вхід і проходять через вузли першого рівня. Перший рівень вузлів обробляє дані на основі ваги вузлів і передає результати наступному шару. Кожен вузол на кожному рівні з'єднаний з усіма вузлами наступного рівня, але дані передаються лише в одному напрямку — до виходу.

RNN використовуються для прогнозування послідовних даних або часових рядів. Вони можуть робити прогнози на основі попередніх даних, наприклад, моделі можуть передбачати рух фондового ринку або перекладати текст, враховуючи послідовність слів у реченні. В архітектурі RNN дані можуть

повертатися через приховані шари завдяки циклам, що дозволяє мережі враховувати попередні стани під час обчислень. Деякі RNN містять спеціалізовані приховані шари, які називаються контекстними, забезпечуючи зворотний зв’язок та підвищуючи точність мережі.

CNN спеціалізуються на розпізнаванні шаблонів та зображень, що робить їх важливими для технологій ШІ, таких як комп’ютерний зір. Наприклад, поштові служби використовують CNN для розпізнавання рукописних поштових індексів. CNN мають спільні ваги та значення зсуву для своїх вузлів, що відрізняє їх від FFNN та RNN. Це означає, що кожен вузол виконує однакову функцію в різних частинах вхідних даних, наприклад, визначає край зображення.

CNN містять два основні типи прихованих шарів: згорткові та об’єднуючі. Згорткові шари фільтрують вхідні дані для вилучення різних ознак, тоді як об’єднуючі шари спрощують параметри, зберігаючи важливу інформацію. Цей процес повторюється кілька разів, іноді включаючи інші шари, такі як багатошаровий перцептрон або випрямлений лінійний блок для активації.

GAN відрізняються від інших моделей тим, що складаються з двох окремих НМ: генератора та дискримінатора. Генератор створює нові зразки даних, такі як зображення або текст, на основі навчальних даних, тоді як дискримінатор оцінює ці зразки, визначаючи, чи є вони реальними чи підробленими. Дві мережі працюють разом, покращуючи свої результати через взаємодію.

GAN можуть створювати 3D-моделі з 2D-зображень, генерувати нові зображення або створювати навчальні набори даних, які відрізняються від існуючих. Основна архітектура GAN передбачає роботу двох НМ у тандемі для досягнення високої якості вихідних даних [8].

Для виконання поставленого завдання основною метою було створення та дослідження атакуючих прикладів для НМ, зокрема, використовуючи FFNN. Хоча CNN відзначаються своєю високою точністю в класифікації зображень, вибір FFNN обґрунтований її меншою вимогою до обчислювальних ресурсів. У цій роботі пріоритет надається простоті та швидкості обробки, а також демонстрації роботи НМ.

1.3 Принципи роботи нейромережі прямого зв’язку

Робота FFNN ґрунтується на ключових принципах і компонентах, які в сукупності дозволяють моделі навчатися, узагальнювати та робити передбачення. Основою побудови мережі є штучний нейрон (рис. 1.4).

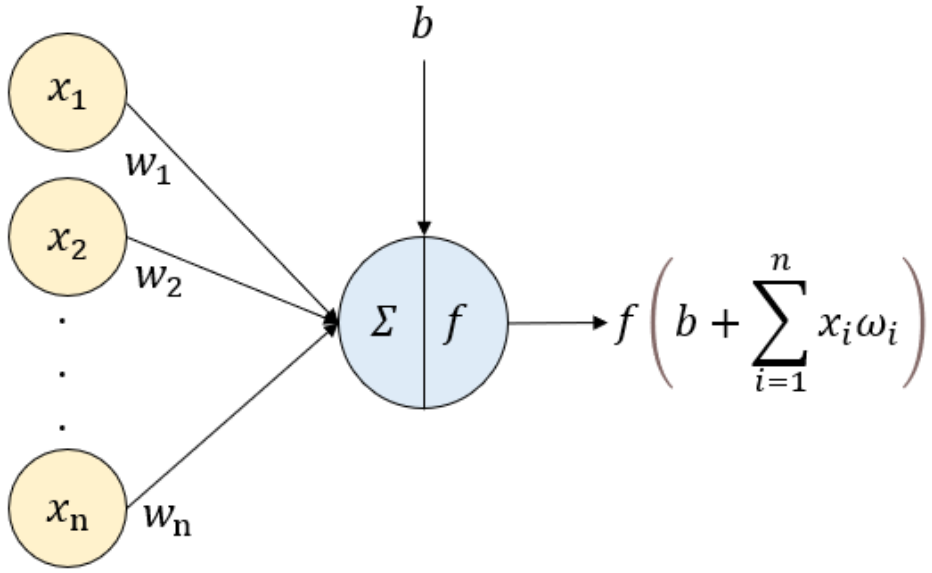


Рисунок 1.4 – Представлення штучного нейрона

Згідно з рис. 1.4 представлення штучного нейрона приймає на вхід значення $x_1 \dots x_n$. Вхідний сигнал x_i має відповідний ваговий коефіцієнт w_i. Вагові коефіцієнти визначають важливість кожного вхідного сигналу. Під час навчання моделі вагові коефіцієнти налаштовуються для мінімізації похибки. Зміщення b додається до зваженої суми вхідних сигналів. Воно дозволяє моделі краще підлаштовуватися під дані і зсувати активаційну функцію.

На першому етапі нейрон обчислює зважену суму вхідних сигналів, до якої додається зміщення. Процес виражений через формулу (1.1)

$$z = b + \sum_{i=1}^{n} x_i w_i \qquad (1.1)$$

На другому етапі результат z передається через активаційну функцію f. Активаційна функція відіграє роль механізму прийняття рішень на виході нейрона. Вона дозволяє нейрону визначати лінійні або нелінійні межі прийняття рішень. Крім того, активаційна функція має нормалізуючий ефект, що запобігає значному зростанню вихідних значень нейронів при проходженні через кілька шарів, таким чином уникнувши ефекту каскаду. Існують три найпоширеніші активаційні функції: сигмоїдальна (рис.1.5), ReLU (рис.1.6) та tanh (рис.1.7).

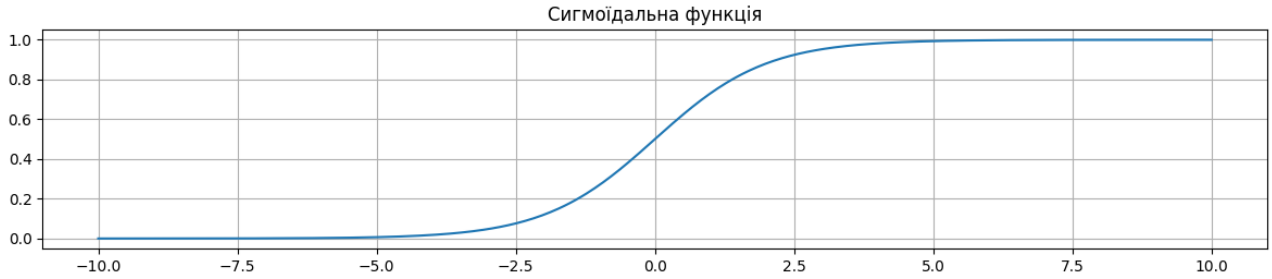


Рисунок 1.5 – Сигмоїдальна функція активації

Сигмоїдальна функція (рис.1.5) активації є однією з класичних функцій активації, яка широко використовується в НМ, особливо у простих моделях. Вихідні значення знаходяться в межах від 0 до 1. Сигмоїда є гладкою, дифернційованою та монотонно зростаючою, що є перевагою для оптимізаційних алгоритмів, таких як градієнтний спуск. Одним із головних недоліків даної функції є проблема зникання градієнта. Тобто при великих або занадто малих вхідних даних, градієнти стають дуже малими, що уповільнює процес навчання НМ.

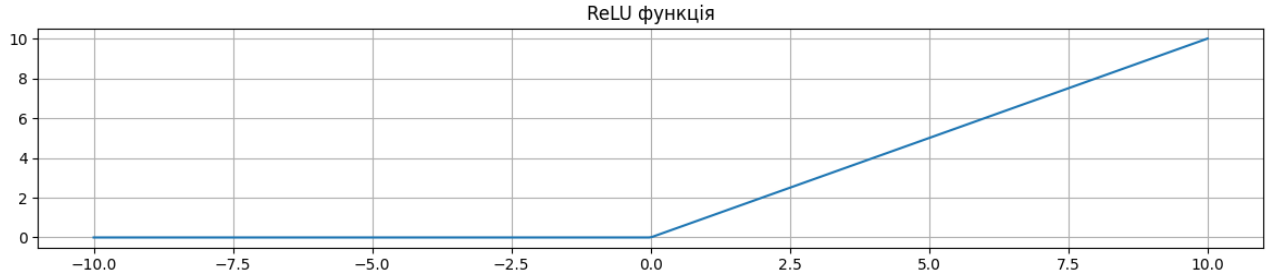


Рисунок 1.6 – ReLU функція активації

Функція ReLU (рис.1.6) дуже проста у виконанні, оскільки вона просто обмежує всі негативні значення нулем і залишає позитивні значення без змін. Завдяки цьому, ReLU легко і швидко обчислюється, що робить її дуже ефективною для великих НМ. Один з недоліків ReLU полягає в тому, що нейрони можуть "вмирати" під час тренування. Якщо вхідні значення постійно негативні, нейрон може ніколи не активуватись, оскільки його вихід буде завжди нульовим. Такий процес називається проблемою "мертвих" нейронів.

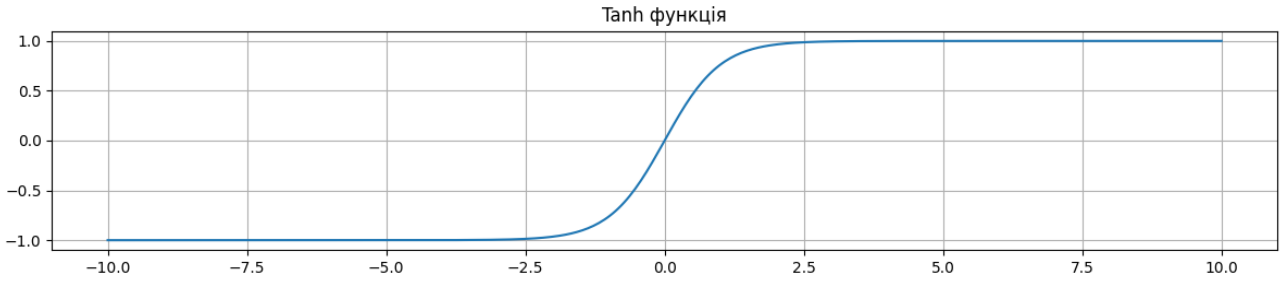


Рисунок 1.7 – Tanh функція активації

Тангенс гіперболічний (рис.1.7) часто використовується в прихованих шарах нейронних мереж для забезпечення нелінійності. Вихідні значення функції tanh знаходяться в межах від -1 до 1. На відміну від сигмоїдальної функції, значення tanh централізовані на нуль, що робить її більш придатною для роботи з даними, оскільки вона допомагає уникати зміщення градієнтів до нуля і прискорювати навчання. Як і у випадку з сигмоїдальною функцією, при великих або малих значеннях вхідних даних градієнти можуть ставати дуже малими (приблизно рівними нулю), що уповільнює навчання нейронної мережі.

Отже, вихідним сигналом нейрона є результат активаційної функції, який може передаватися як вхідний сигнал до наступного шару нейронів або бути кінцевим результатом обчислень.

Ознайомившись із штучним нейроном і можливими активаційними функціями, доцільно розглянути структуру моделі FFNN, яка була використана в цій роботі (рис. 1.8).

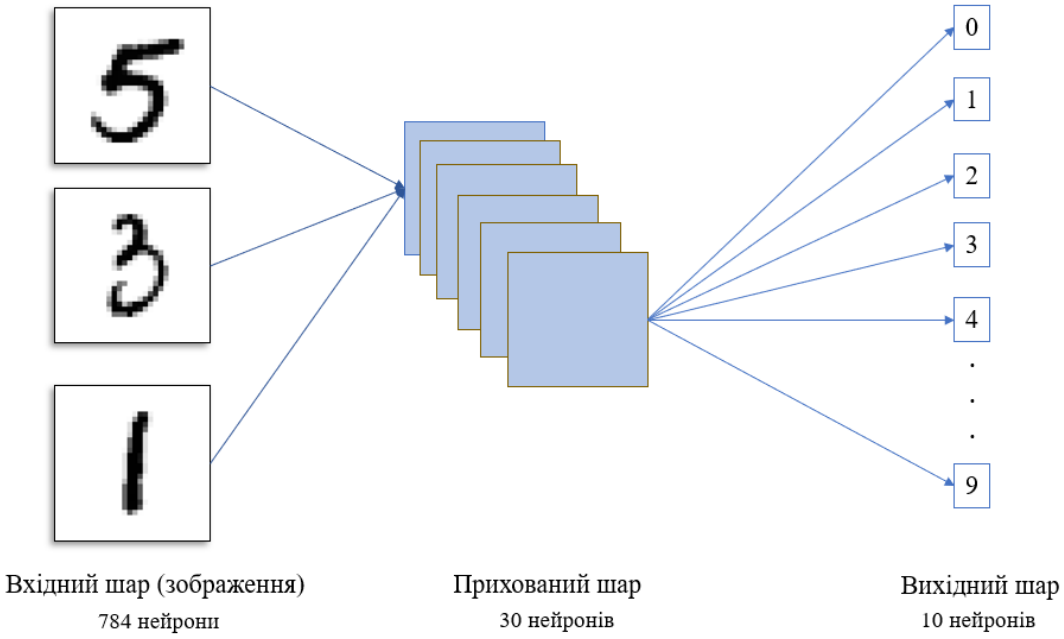


Рисунок 1.8 – Структура моделі FFNN

Вхідний шар нейромережі складається з 784 нейронів, кожен з яких відповідає одному пікселю вхідного зображення розміром 28×28 пікселів[9]. Нейрон вхідного шару отримує значення пікселя і передає його до наступного шару мережі. Задана структура дозволяє моделі безпосередньо обробляти зображення у їх початковій формі, забезпечуючи простоту і ефективність обробки вхідних даних.

Приховані шари складаються з нейронів, які обробляють вхідні сигнали з попереднього шару. Модель використовує лише один прихований шар з 30 нейронів, що дозволяє легко налаштовувати і оптимізувати мережу. Використання одного прихованого шару зменшує обчислювальну складність моделі і знижує ризик перенавчання, що є важливим для задачі розпізнавання рукописних цифр.

У роботі використовується сигмоїдальна функція активації через її простоту та традиційне застосування в нейронних мережах. Оскільки однією з основних задач є бінарна класифікація, діапазон вихідних значень сигмоїди (від 0 до 1) є найкращим вибором для цієї задачі.

Вихідний шар складається з 10 нейронів, кожен з яких відповідає одному з класів (цифри від 0 до 9). Кожен нейрон вихідного шару обчислює ймовірність

належності вхідного зображення до певного класу, що дозволяє моделі робити точні передбачення.

Для класифікації використовується метод найбільшої ймовірності (рис. 1.9), який перетворює вихідні значення нейронів у ймовірності, що дозволяє більш точно визначити клас зображення.

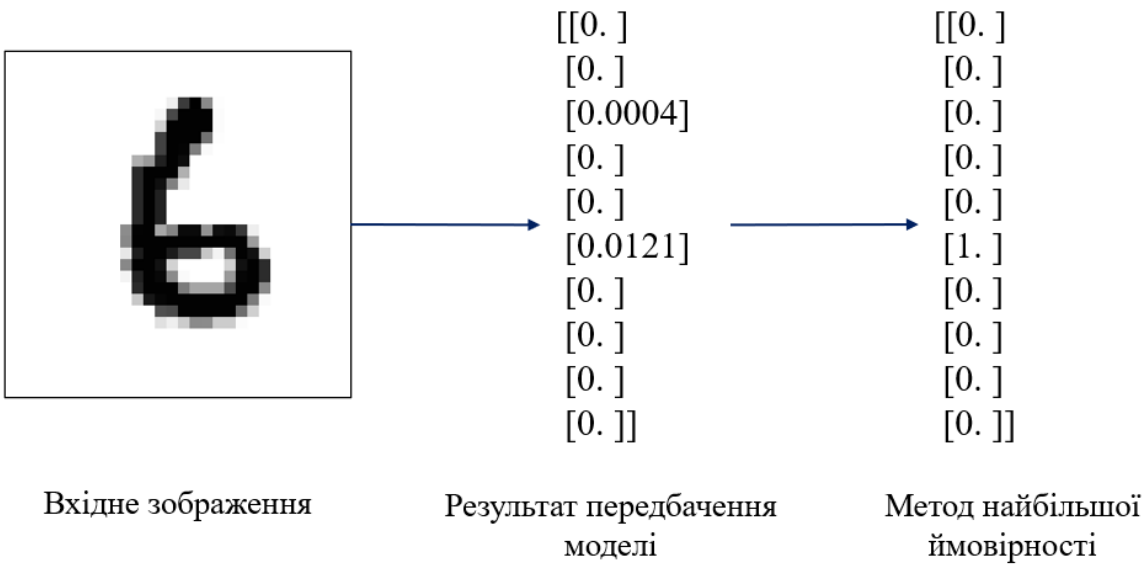


Рисунок 1.9 – Класифікація методом найбільшої ймовірності

У навчанні мережі використовується метод зворотного поширення помилки та оптимізатор. Метод зворотного поширення помилки є основним алгоритмом навчання для ШНМ, який дозволяє ефективно мінімізувати функцію втрат та коригувати ваги мережі. Такий підхід дозволяє швидко і ефективно знаходити оптимальні значення ваг для великих і складних моделей. Проте, метод має і недоліки: для ефективного навчання необхідні великі обсяги даних. У глибоких мережах можуть виникати проблеми з затуханням градієнтів, що ускладнює навчання. Ефективність методу також сильно залежить від вибору параметрів, таких як швидкість навчання і структури мережі. Попри це, зворотне поширення помилки є базовим підходом у сучасному МН і значно сприяє розвитку галузі ШІ.

Отже, у цьому розділі було визначено та класифіковано основні типи систем ШІ, зокрема машинне навчання, обробку природної мови, комп'ютерний зір та робототехніку, із зазначенням їх хронології розвитку та прикладів

застосування. Детально розглянуто принципи роботи та архітектури ШНМ, зокрема архітектури прямих нейронних мереж, рекурентних нейронних мереж, згорткових нейронних мереж та генеративних змагальних мереж. Описано процес навчання та принципи функціонування нейронів, активаційних функцій, а також структуру моделі FFNN, яка використовувалась у дослідженні для розпізнавання зображень.

2 МЕТОДИ АТАК ТА ЗАХИСТ НЕЙРОННИХ МЕРЕЖ

Згідно з [10], станом на 2023 рік кіберзлочини продовжують зростати в порівнянні з попередніми роками розвитку технологій (рис. 2.1). Лідируючими галузями, які піддаються різним видам злочинності, стали сектор охорони здоров'я та урядові організації. Зловмисне програмне забезпечення залишається найпоширенішим типом загрози, підвищившись до 35,9% у порівнянні з 34,7% у попередньому році. Використання вразливостей збільшилося до 16,3% з 7,8%. Захоплення облікових записів знизилося до 9,2% з 15,5%, а цілеспрямовані атаки дещо зменшилися до 7% з 8%. Цікаво, що DDoS-атаки залишилися стабільними на рівні 3,8%, що відповідає показникам 2022 року.

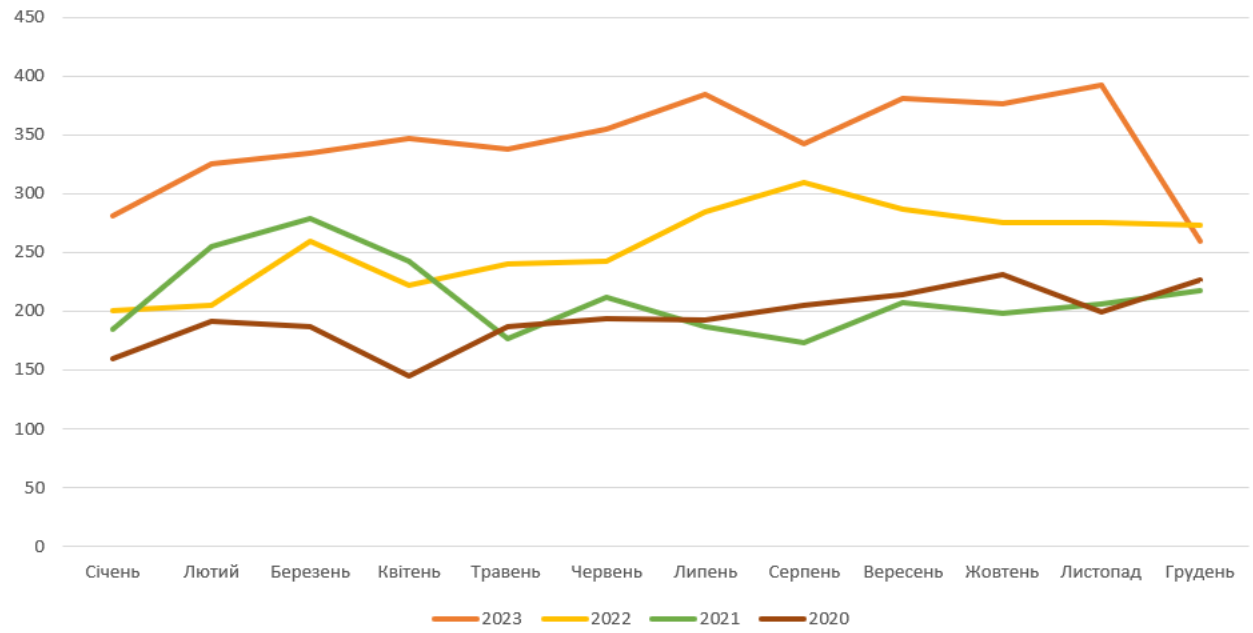


Рисунок 2.1 – Порівняльна діаграма кіберзлочинів за 2020 – 2024 роки

На основі представлених статистичних даних, захист та передбачення можливих хакерських атак є важливими етапами розробки та впровадження будь-якої системи, зокрема систем ШІ. У цьому розділі будуть розглянуті потенційні атаки на НМ та відповідні методи захисту від них.

2.1 Атаки на рівні даних

Атаки на рівні даних відрізняються за своїми підходами, проте всі вони мають спільну рису – маніпулювання даними, на яких навчається модель. До основних типів належать атаки із застосуванням шумів (адверсивні атаки), а саме атаки із використанням градієнтних методів, таких як FGSM або PGD, а також цільові атаки і отруєння чи підміна даних.

Адверсивні атаки спрямовані на слабкі місця моделей машинного навчання та використовують мінімальні зміни до вхідних даних, які можуть бути непомітними для людини, але значно впливають на роботу моделі. З точки зору середовища, атаки можна розділити на атаки чорної, білої або сірої скриньки.

Атака «чорної скриньки» передбачає, що зловмисник не має інформації про базову структуру, параметри чи стратегії захисту атакованої моделі. У цьому випадку він взаємодіє з моделлю лише через її входи та виходи. Атака «білої скриньки», навпаки, відбувається, коли зловмисник володіє повною інформацією про атаковану модель, включаючи функцію втрат і оптимізовані параметри, згодом використовуючи ці знання для здійснення атаки. І остання з розглянутих, проте не менш важлива атака «сірої скриньки» реалізовується тоді, коли зловмисник має лише часткові знання про модель, яку він атакує [11].

Адверсивні атаки можуть відбуватися на етапі моделювання. Зокрема, зловмисник може намагатися ввести модель в оману на основі НМ. У якості прикладу розглянемо класифікатор зображень. У цьому випадку двохфазна атака виконується наступним чином: спочатку генерується адверсивний приклад за допомогою класифікатора зображень, а потім повертається цей приклад назад у модель.

На першому етапі зловмисник модифікує значення пікселів доброякісного прикладу, щоб максимізувати значення функції втрат. Такий підхід призводить до того, що класифікатор зображень неправильно класифікує приклад або мінімізує значення функції втрат для неправильного класу, визначеного зловмисником. Варто зазначити, що зловмисник може використовувати різні стратегії для керування напрямком збурення на основі своїх попередніх знань про атаковану модель НМ. Наприклад, якщо архітектура НМ, параметри (ваги та

зміщення) і функція втрат моделі відомі (зокрема, через скомпрометований сервер або недобросовісного співробітника), хакер може застосувати алгоритм атаки на основі FGSM для розрахунку збурення і створення адверсивного прикладу.

Метод швидкого градієнтного знака (FGSM) є однією з найпоширеніших технік для створення адверсивних прикладів. Вона використовує градієнти функції втрат моделі для внесення малих змін до вхідних даних.

$$x' = x + \epsilon \cdot sign\big(\nabla_x J(\theta, x, y)\big) \tag{2.1}$$

де x – вхідні дані;

ϵ – величина збурення;

y – вказує на клас основної істини;

$sign(\cdot)$ – повертає знак дійсної величини;

$\nabla_x J(\theta, x, y)$ – градієнт функції втрат по відношенню до вхідних даних.

Для створення більш ефективного адверсивного прикладу, зловмисник може поступово змінювати пікселі зображення, запускаючи складніший алгоритм атаки через проектований градієнтний спуск (PGD).

$$x_{t+1} = Proj_{B_\epsilon(x)}(x_t + \alpha \cdot sign(\nabla_{x_t} J(\theta, x_t, y))) \tag{2.2}$$

де x_t – поточний адверсивний приклад на t-й ітерації;

α – розмір кроку для кожної ітерації;

$\nabla_{x_t} J(\theta, x_t, y)$– градієнт функції втрат J по відношенню до x_t, обчислений при параметрах моделі θ і мітці y;

$sign(\cdot)$ – функція, яка повертає знак градієнта;

$Proj_{B_\epsilon(x)}$– проекція на дозволений простір $B_\epsilon(x)$, що обмежує x_t в межах ϵ - сфери навколо початкового прикладу x.

Процес створення ефективного адверсивного прикладу може включати багато ітерацій модифікації вхідних даних і аналізу їх впливу на вихідні результати моделі. Таким чином, зловмисник може використовувати різні методи оптимізації для покращення якості збурення, щоб воно залишалося непомітним для ока людини, через властивості її зорової системи, але водночас викликало значні помилки в моделі. Після успішної перевірки в

контрольованому середовищі зловмисник може застосувати адверсивні приклади в реальних сценаріях, наприклад, у виробничих системах, вебсервісах або мобільних додатках для досягнення своїх цілей.

З іншого боку, застосування адверсивних атак може бути спрямоване не лише на всю модель машинного навчання в цілому, а й на конкретну "ціль" у вигляді певного класу або виходу моделі. На відміну від нецільових атак, метою яких є просто отримання будь-якої некоректної класифікації, цільові адверсивні атаки переслідують конкретну мету – змусити модель класифікувати вхідні дані як визначений заздалегідь клас або категорію.

Хоча для здійснення цільових атак можуть застосовуватися ті самі методи, що й для нецільових, проте вони використовуються з більш вузькою і специфічною метою. Внаслідок цього, зловмисник, імовірно, досягне бажаного результату набагато швидше порівняно з нецільовими атаками. Крім того, виявлення таких цілеспрямованих атак є складнішим завданням і потребує застосування методів реверсивного інжинірингу для аналізу та виправлення ситуації після факту її здійснення.

2.2 Атака на рівні моделі

Атаки на рівні моделі є різновидом адверсивних атак, спрямованих на безпосереднє маніпулювання самою моделлю МН, а не лише на вхідні дані. Такі атаки можуть відбуватися на різних етапах життєвого циклу моделі: при розробці, під час навчання, розгортання та навіть під час експлуатації.

До атак, які здійснюються на етапі розробки моделі, належать вбудовування бекдорів і вплив на процес відбору даних. Бекдор або «чорний хід» активується при введенні певних вхідних даних, що дозволяє зловмиснику маніпулювати результатами моделі. Інший спосіб маніпуляції включає використання даних, які можуть погіршити продуктивність або внести упередженість у модель.

Атаки, що впливають на оптимізацію моделі, зазвичай здійснюються на етапі її навчання. У цьому випадку модель може демонструвати поведінку, бажану для зловмисника. Наприклад, можуть бути здійснені атаки на функцію

втрат, що включають зміну ваг для надання більшої значущості певному класу або зразку даних, маніпуляцію гіперпараметрами алгоритму оптимізації або дестабілізацію процесу навчання.

Якщо модель недостатньо захищена, вона може бути викрадена або незаконно скопійована для подальшого аналізу та виявлення вразливостей. Навіть добре захищені моделі можуть бути піддані атакам, спрямованим на виявлення вразливостей, включаючи пошук бекдорів, через які зловмисники можуть отримати доступ до моделі.

За останні кілька років було зафіксовано декілька помітних прикладів адверсивних атак на рівні моделі. Наприклад, у 2020 році дослідники продемонстрували можливість змусити чат-бота генерувати невідповідні або образливі відповіді, додаючи невеликі збурення до вхідного тексту [12]. Цей тип атаки може потенційно завдати шкоди репутації компанії або призвести до втрати клієнтів.

У 2022 році група дослідників показала, що адверсивні приклади можна використовувати для обходу фільтрів спаму, дозволяючи шкідливим електронним листам уникати виявлення. Вони створили електронні листи зі спамом, додавши збурення до вмісту електронної пошти, що призвело до неправильної класифікації спам-фільтрами, які визначили ці листи як не спам [13].

2.3 Захист нейронних мереж

Перед розробкою НМ перш за все варто задуматися про забезпечення її захисту на всіх етапах життєвого циклу. Важливо врахувати заходи для забезпечення конфіденційності, цілісності та доступності моделі, а також передбачити можливі витоки даних, злами та фальсифікації. Хоча розподіл ролей під час розробки можна вирішити за допомогою затверджених стандартів комплексних систем захисту інформації, необхідно також зосередитися на забезпеченні захисту від адверсивних атак, розглянутих у попередніх підрозділах поточного розділу, які наразі є найбільшим викликом для НМ.

На сьогоднішній день існують методи протидії різним типам атак, які варіюються від конкретних засобів до загальних стратегій захисту. Оскільки FFNN є досить простими, до них можна застосовувати більш загальні стратегії захисту. Типові заходи включають наступні методології: адверсивне навчання, моніторинг, підсилення конструкцій міцності моделі та знищення структур адверсивного збудження (рис. 2.2).

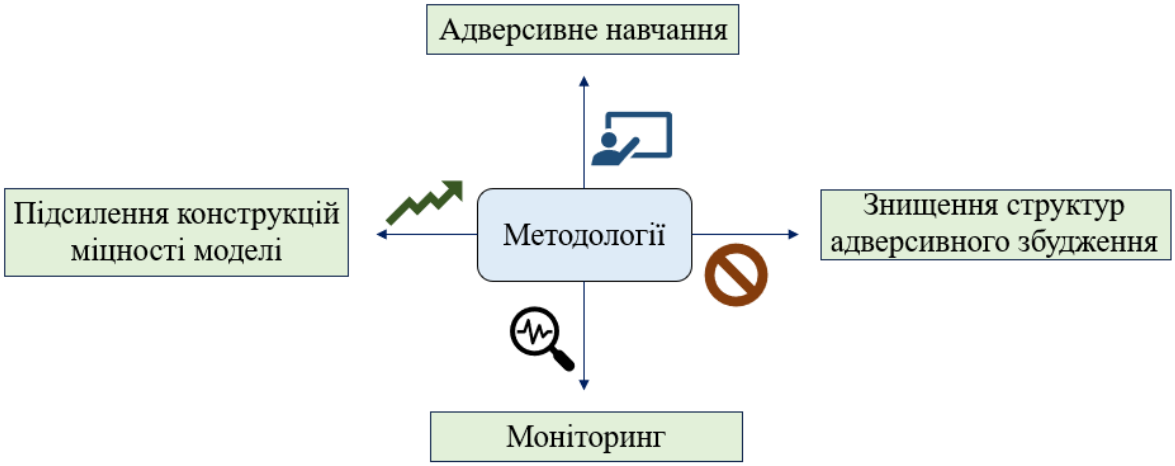


Рисунок 2.2 – Методології захисту нейронної моделі

Адверсивне навчання — це техніка глибокого навчання, спрямована на підвищення надійності моделі проти адверсивних прикладів. Ключовий метод передбачає додавання таких прикладів до процесу навчання. Постійно вивчаючи особливості вибірок, модель може краще захищатися від атак, які включають ледь помітні збурення у вхідних даних. Це підвищує точність і ефективність моделі в реальних сценаріях, де вона може стикатися з шкідливими зразками, призначеними для подальшого введення її в оману та отримання неправильних прогнозів.

Наступним важливим аспектом захисту моделей глибокого навчання є моніторинг. Стратегія виявлення адверсивних зразків полягає у модифікації вхідних дані, змушуючи модель робити неправильні прогнози. Для цього можна налаштувати спеціальні моделі в ключових точках системи, щоб ідентифікувати такі зразки й забезпечити раннє попередження про потенційні адверсивні атаки.

Такий підхід дозволяє вживати профілактичних заходів для захисту від атак і підтримувати цілісність прогнозів моделі.

Також не менш важливим компонентом є конструкція міцності моделі. Конструкція міцності передбачає використання спеціальних фільтруючих структур для підвищення стійкості до адверсивного шуму. Такий шум відноситься до тонких збурень, які додаються до вхідних даних, щоб змусити модель робити хибні прогнози. Розробка моделі з підвищеною стійкістю до адверсивного шуму дозволяє краще захищатися і підтримувати точність прогнозів.

Остання з переліку, стратегія знищення структури адверсивного збурення. Вона передбачає використання різних методів для послаблення ефекту адверсивного шуму та запобігання атакам на моделі НМ. Стратегії можуть включати використання алгоритмів фільтрації, руйнування структури шуму та покриття шуму під час обробки даних. Мета даної системи полягає в підвищенні стійкості моделі до адверсивного шуму, який додається до вхідних даних для отримання неправильних прогнозів.

Таким чином, перелічені аспекти захисту НМ є ключовими для створення надійних, безпечних і стійких систем будь-якого масштабу, особливо в тих галузях, де цілісність і точність моделі мають критичне значення.

Отже, у другому розділі було розглянуто різні типи атак на НМ та відповідні методи захисту. Зокрема, було досліджено атаки на рівні даних, такі як адверсивні атаки, включаючи FGSM та PGD, а також цільові атаки і отруєння даних. Адверсивні атаки спрямовані на слабкі місця моделей машинного навчання, використовуючи мінімальні зміни до вхідних даних, які можуть бути непомітними для людини, але значно впливають на роботу моделі. Окрім того, було висвітлено атаки на рівні моделі, такі як вбудовування бекдорів і маніпуляція процесом навчання.

Також було досліджено методи захисту, зокрема адверсивне навчання, яке підвищує стійкість моделей до атак, моніторинг для виявлення та запобігання адверсивним зразкам, підсилення міцності моделі для підвищення її стійкості до адверсивного шуму, та знищення структур адверсивного збурення, що

передбачає використання алгоритмів фільтрації для запобігання атакам. Ці заходи забезпечують надійність та безпеку НМ, особливо в критичних галузях.

3 РЕАЛІЗАЦІЯ АТАК ТА ЗАХИСТУ НЕЙРОННОЇ МЕРЕЖІ

У цьому розділі буде представлено практичну реалізацію досліджуваних методів для розпізнавання рукописних цифр на базі даних MNIST із використанням FFNN, а також аналіз адверсивних атак та методів захисту від них. На основі теоретичних аспектів, розглянутих у попередньому розділі, проведено серію експериментів для оцінки ефективності запропонованих підходів.

3.1 Технічні характеристики та інструменти для дослідження

Спершу доцільно розглянути базу даних MNIST, яка була зібрана Національним інститутом стандартів і технологій США. Датасет MNIST є стандартним набором даних для навчання і тестування алгоритмів розпізнавання рукописних цифр. Ці зображення є відсканованими зразками почерку 250 людей, половина з яких були співробітниками Бюро перепису населення США, а половина з яких - учні середньої школи. Він містить 60 000 зображень для навчання та 10 000 зображень для тестування, кожне з яких є чорно-білим із розміром 28x28 пікселів (рис.3.1). Кожне зображення представлено вектором довжиною 784, де кожен елемент вектора відповідає інтенсивності пікселя від 0 (білий) до 255 (чорний), що було зазначено раніше.

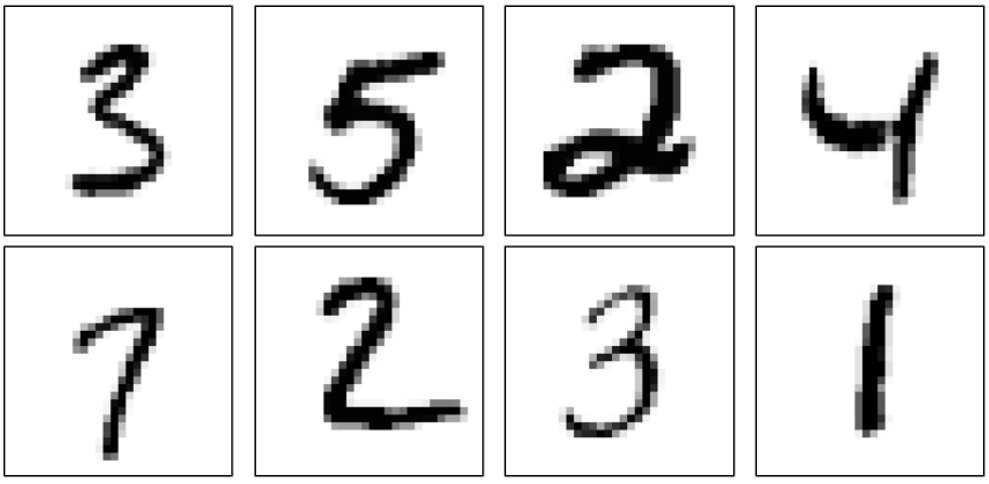


Рисунок 3.1 – Приклади рукописних цифр з датасету MNIST

Для виконання практичної частини роботи було використано пристрій з наступними характеристиками, які наведено в табл. 3.1.

Таблиця 3.1 – Технічні характеристики пристрою

Характеристика	Опис
Процесор (CPU)	Intel Core i7-4700MQ (2.64 GHz)
Оперативна пам'ять (RAM)	8 ГБ DDR3L SDRAM 1600 МГц
Графічний процесор (GPU)	NVIDIA GeForce GT 750M (2 ГБ GDDR5)
Операційна система (OS)	Windows 10x64

Процесор Intel Core i7-4700MQ з тактовою частотою 2.64 GHz забезпечує високу обчислювальну потужність, що є важливим для проведення складних обчислень, таких як тренування НМ і виконання алгоритмів МН. Завдяки можливості розгону до 3.4 GHz (в режимі Turbo Boost), процесор може справлятися з важкими завданнями і забезпечувати високу продуктивність під час виконання численних експериментів.

Оперативна пам'ять об'ємом 8 ГБ забезпечує достатній простір для зберігання і обробки великих обсягів даних, що є ключовим при роботі з базами даних, такими як MNIST. Висока швидкість пам'яті (1600 МГц) забезпечує швидкий доступ до даних, що покращує загальну продуктивність системи під час навчання моделей та проведення експериментів.

Графічний процесор NVIDIA GeForce GT 750M з 2 ГБ відеопам'яті GDDR5 забезпечує прискорення обчислень для задач, пов'язаних з обробкою зображень і тренуванням нейронних мереж. GPU значно покращує швидкість обробки даних у порівнянні з використанням лише центрального процесора, що є важливим при виконанні задач глибокого навчання.

Операційна система Windows 10x64 є стабільною і підтримує широкий спектр ПЗ, необхідного для проведення досліджень. Вона забезпечує сумісність з багатьма бібліотеками МН, а також надає зручне середовище для розробки у Visual Studio Code. Платформа Windows 10 дозволяє легко встановлювати та налаштовувати інструменти для обробки даних та розробки моделей. Зокрема розглянемо і інструменти, що доцільно використовувати для дослідження.

Python є однією з найкращих мов програмування для досліджень у сфері НМ з кількох причин. Його популярність у цій галузі обумовлена багатьма

факторами, включаючи розширені бібліотеки та активну спільноту розробників. Обрана мова програмування має простий і зрозумілий синтаксис, що дозволяє легко писати та читати код. Це знижує бар'єр для входу і дозволяє дослідникам зосередитися на алгоритмах і концепціях нейронних мереж, а не на складнощах програмування.

Visual Studio Code є зручним текстовим редактором із безліччю розширень, які полегшують роботу з Python, такі як IntelliSense для автозаповнення коду, Pylint для аналізу коду та Jupyter для інтеграції з ноутбуками Jupyter.

Отже, використання мови Python в поєднанні з Visual Studio Code дозволяє ефективно створювати, тренувати і тестувати НМ, а також реалізовувати методи адверсивних атак та захисту від них.

3.2 Атаки на нейронні мереж прямого зв’язку

Атакувальні приклади — це спеціально створені вхідні дані, які спрямовані на оману НМ, змушуючи їх робити помилкові прогнози. Вони створюються шляхом додавання малих, спеціально розрахованих збурень до вхідних даних, таких як зображення. Збурення орієнтовані на слабкі місця моделі, використовуючи градієнти функції втрат для максимізації помилок класифікації.

У нашому випадку, для генерації адверсивних прикладів потрібно вирішити задачу мінімізації. Для цього встановлюється «цільова» мітка, яка буде називатися $\overrightarrow{y_{\text{ц}}}$. Наприклад, якщо потрібно, щоб мережа визначала адверсивний приклад за цифру 8, то потрібно обрти $\overrightarrow{y_{\text{ц}}}$ у вигляді вектора з єдиною активною позицією на 8 елементів вектора рівному 1.

Наступним кроком визначається функція вартості.

$$C = \frac{1}{2}\|\vec{y} - \hat{y}(\vec{x})\|_2^2 \tag{3.1}$$

де: $\|\cdot\|_2^2$ – квадрат евклідової норми;

$\vec{x}$ – вхідне зображення мережі;

$\hat{y}$ – вихідне значення мережі, що є функцією від вхідного зображення мережі.

Основною метою виступає знаходження такого $\vec{x}$, при якому C мінімізується. Цей процес означатиме, що вихід мережі буде близьким до бажаного значення $\overrightarrow{y_{\text{ц}}}$. У повному математичному формулюванні задача оптимізації вигладає в наступному вигляді:

$$arg\ \min_{\vec{x}} C(\vec{x}) \tag{3.2}$$

Для вирішення цієї задачі використовується метод градієнтного спуску. Починаючи з випадкового вектора $\vec{x}$ і поступово оновлюючи його, робляться кроки у напрямку, протилежному градієнту функції вартості $\nabla_x C$. Для обчислення градієнту використовується зворотне поширення помилки у НМ. Проте, на відміну від стандартного навчання, де потрібно змінювати ваги та зміщення мережі, для адверсивних атак їх залишають постійними і змінюють лише вхідні дані.

У підсумку, цей процес дозволяє створювати зображення, яке мережа буде класифікувати відповідно до цільової мітки, навіть якщо це зображення насправді не відповідає цій мітці.

Для початку буде визначена, як було зазначено раніше, сигмоїдна функція і знайдена її похідна.

Лістинг 3.1 – Знаходження сигмоїди і похідної

```
def sigmoid(z):
    return 1.0/(1.0+np.exp(-z))
def sigmoid_prime(z):
    return sigmoid(z)*(1-sigmoid(z))
```

Наступним кроком реалізується функція для обчислення градієнта похідної функції вартості $\nabla_x C$ відносно $\vec{x}$ з цільовою міткою $\overrightarrow{y_{\text{ц}}}$. Функція, яка обчислює $\nabla_x C$, визначає, в якому напрямку і наскільки потрібно змінити кожен піксель вхідного зображення $\vec{x}$, щоб функція вартості зменшилась C.

Лістинг 3.2 – Обчислення градієнта функції вартості

```
def input_derivative(model, x, y):
    nabla_b = [np.zeros(b.shape) for b in model.biases]
    nabla_w = [np.zeros(w.shape) for w in model.weights]
```

Продовження лістингу 3.2.

```
    nabla_w = [np.zeros(w.shape) for w in model.weights]
    activation = x
    activations = [x]
    zs = []
    for b, w in zip(model.biases, model.weights):
        z = np.dot(w, activation)+b
        zs.append(z)
        activation = sigmoid(z)
        activations.append(activation)
    delta = model.cost_derivative(activations[-1], y) * \
        sigmoid_prime(zs[-1])
    nabla_b[-1] = delta
    nabla_w[-1] = np.dot(delta, activations[-2].transpose())
    for l in range(2, model.num_layers):
        z = zs[-l]
        sp = sigmoid_prime(z)
        delta = np.dot(model.weights[-l+1].transpose(), delta) * sp
        nabla_b[-l] = delta
        nabla_w[-l] = np.dot(delta, activations[-l-1].transpose())
```

Другим етапом генерується функція адверсивної атаки на НМ за допомогою градієнтного спуску. Вектор goal розміром 10x1 заповнений нулями. Встановлюється одиниця на позиції n, перетворюючи вектор у one-hot представлення цільової мітки. Випадкове зображення x розміром 28x28 пікселів, заповнене значеннями з нормального розподілу зі середнім значенням 0.5 і стандартним відхиленням 0.3.

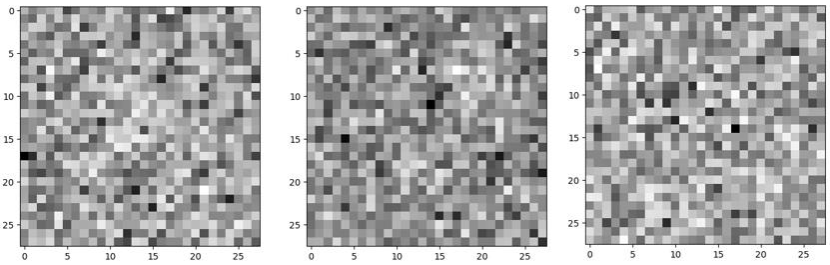
Лістинг 3.3 – Функція адверсивної атаки

```
def adversarial(model, n, steps, eta):
    goal = np.zeros((10, 1))
   goal[n] = 1
 x = np.random.normal(.5, .3, (784, 1))
goal[n] = 1
```

Продовження лістингу 3.3

```
x = np.random.normal(.5, .3, (784, 1))
for i in range(steps):
d = input_derivative(model, x, goal)
   x -= eta * d
return x
```

Перейдемо до демонстрації результатів реалізації створеного атакувального прикладу. Вихідні дані моделі завжди представлені у вигляді вектора зі значеннями від 0 до 1. Модель визначає найближче до 1 значення як той результат, який було задано (рис.3.2).



Цифра 2 Цифра 3 Цифра 4

Рисунок 3.2 – Створені зображення цифр моделлю

Отримані результати важко розрізнити людським оком і вони виглядають беззмістовними, тому наступним завданням буде згенерувати зображення, яке виглядає як одне число, але НМ буде впевнена, що воно є іншим. Для цього потрібно змінити функцію вартості. Замість того, щоб просто оптимізувати вхідне зображення $\vec{x}$ для досягнення цільової мітки на виході, необхідно також оптимізувати вхідне зображення, щоб воно виглядало як певне число $\vec{x}_{\text{ц}}$. Оновлена функція вартості виглядатиме наступним чином:

$$C = \left\|\overrightarrow{y_{\text{ц}}} - \hat{y}(\vec{x})\right\|_2^2 + \lambda\left\|\vec{x} - \overrightarrow{x_{\text{ц}}}\right\|_2^2 \qquad (3.3)$$

Другий доданок функції вартості визначає відстань між $\vec{x}$ та $\overrightarrow{x_{\text{ц}}}$ – зображення, яке буде використано для атакуючого прикладу. Оскільки задача мінімізації функції залишається актуальною, також додається вимога зменшити відстань між атакувальним прикладом і цим зображенням. Гіперпараметр λ, який

може бути налаштованим будь-яким чином, визначає пріроритетність між оптимізацією бажаного виходу та зображення, яке має виглядати як $\overrightarrow{x_{\text{ц}}}$.

Нова створена функція має єдину відмнність, що полягає в додатковому члені в оновленні градієнтного спуску для регуляризуючого компонента.

Лістинг 3.4 – Оновлена функція вартості

```
def sneaky_adversarial(model, n, x_target, steps, eta, lam=.2):
    goal = np.zeros((10, 1))
    goal[n] = 1
    x = np.random.normal(.5, .3, (784, 1))
    for i in range(steps):
        d = input_derivative(model, x, goal)

        x -= eta * (d + lam * (x - x_target))
    return x
```

Отримані результати (рис. 3.3) демонструють, що деякі цифри розпізнаються з більшою точністю, ніж інші. Зокрема, цифри 0, 2, 3, 5, 6, 8 модель визначає з високою точністю, тоді як цифри 1, 4, 7, 9 мають нижчий рівень. Це можна пояснити регулюючим компонентом, доданим до функції вартості, який допомагає моделі краще узагальнювати та уникати перенавчання. Однак, складніші форми та схожість деяких цифр між собою можуть ускладнювати їх точне розпізнавання.

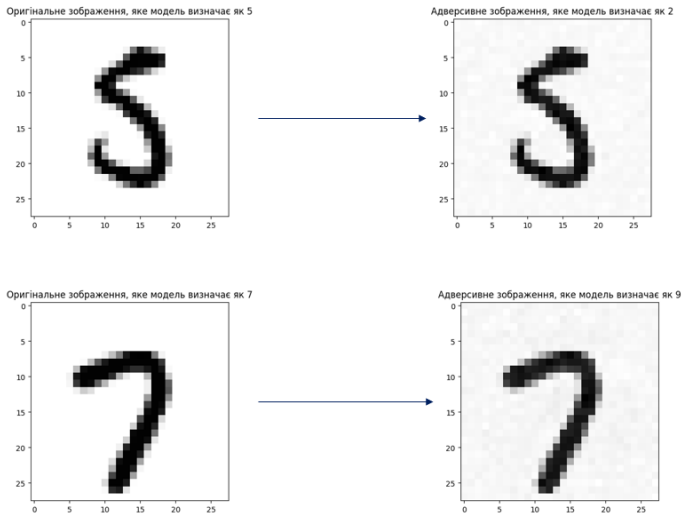


Рисунок 3.3 – Оригінальне число і адверсивний приклад

3.3 Захист нейронної мережі від атак

Захист від атакувальних прикладів у НМ є однією з ключових проблем сучасної кібербезпеки. Атакувальні приклади, які використовують вразливості в моделях МН, можуть призводити до неправильних рішень та серйозних наслідків у реальних додатках, таких як розпізнавання образів, автономне водіння та біометрична ідентифікація.

У цій роботі буде розглянуто два різні підходи до захисту від атакувальних прикладів. Один із них передбачає використання бінарного порогового перетворення. Реалізація цього методу доволі проста: кожен піксель зображення перетворюється у чорний або білий колір залежно від його початкового значення. Якщо значення пікселя менше за 0,5 – білий, більше за 0,5 – чорний.

Такий підхід дозволяє усунути будь-який шум, накладений на зображення. Навіть якщо на вхідній стадії хтось спробує додати шум до зображення, атака буде нейтралізована на етапі перетворення.

Лістинг 3.5 – Функція бінарного перетворення

```
def binary_thresholding(n, m):
    x = sneaky_generate(n, m)
    x = (x > .5).astype(float)
    plt.imshow(x.reshape(28,28), cmap="Greys")
    plt.title("Застосовано бінарне перетворення")
    plt.show()
    print("Prediction with binary thresholding: " + str(np.argmax(np.round(model.feedforward(x))))
+ '\n')
    print("Network output: ")
    print(np.round(model.feedforward(x), 2))
```

Виходячи із результатів, представлених на рис 3.4, бінаризація зображення дозволяє усунути накладений шум, що значно підвищує точність класифікації, навіть якщо вхідні дані були змінені.

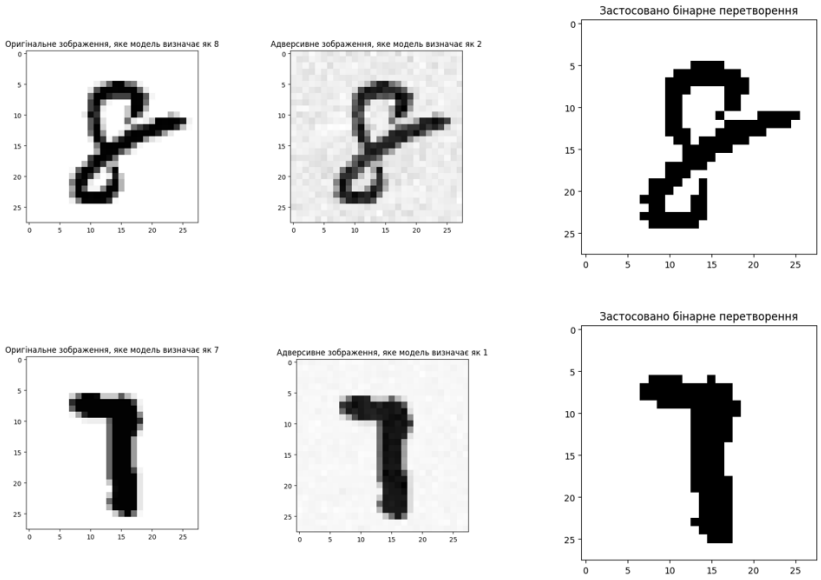


Рисунок 3.4 – Застосування бінарного порового перетворення

Хоча бінарне порогове перетворення є ефективним методом для видалення шуму з атакувальних прикладів і підвищує точності класифікації НМ, воно не є універсальним рішенням і не завжди буде працювати. Одна з головних причин полягає в тому, що цей підхід не буде працювати на CNN моделях, які навчаються на великій кількості кольорових зображень, наприклад ImageNet.

Бінаризація може призвести до втрати суттєвої інформації, особливо у випадках, коли важливі деталі зображення містяться в проміжних відтінках. Крім того, у складніших атаках зловмисники можуть адаптувати свої методи для створення атакувальних прикладів, які залишаються ефективними навіть після бінаризації. Зокрема, вони можуть використовувати структури та патерни, які витримують процес порогового перетворення, або використовувати методи, що маніпулюють самими бінарними значеннями.

Виходячи з цього, необхідно розробити більш універсальний підхід, який зможе захистити майже будь-яку нейронну мережу. Якщо розробник має доступ до атакувальних прикладів, як у нашому випадку, можна створити велику кількість таких прикладів і додати їх до навчального набору даних мережі. Це дозволить моделі навчитися на зашумлених даних і ігнорувати шум на зображеннях, підвищуючи її стійкість до атакувальних прикладів.

Лістинг 3.6 – Створення 10000 адверсивних прикладів

```
def augment_data(n, data, steps):
```

Продовження лістингу 3.6

```
augmented = []
    for i in range(n):
        # Progress "bar"
        if i % 500 == 0:
        print("Generated digits: " + str(i))
        rnd_actual_digit = np.random.randint(10)
        rnd_actual_idx = np.random.randint(len(data))
        while np.argmax(data[rnd_actual_idx][1]) != rnd_actual_digit:
        rnd_actual_idx = np.random.randint(len(data))
        x_target = data[rnd_actual_idx][0]
        rnd_fake_digit = np.random.randint(10)
        x_adversarial = sneaky_adversarial(model, rnd_fake_digit, x_target, steps, 1)
        y_actual = data[rnd_actual_idx][1]
        augmented.append((x_adversarial, y_actual))
    return augmented
```

Випадково обирається зображення з датасету, генерується для нього атакувальний приклад із подальшим додаванням його разом із правильною міткою до нового набору даних. Оптимальною кількістю є 10000 нових зображень, оскільки додавання 1000 зображень покращить ефективність лише на декілька відсотків, тоді як додавання 100000 зображень не дає суттєвого приросту у продуктивності порівняно з вибраним набором, але значно ефективніше з точки зору використання ресурсів. Для створення набору з 10000 нових елементів на пристрої з заданими технічними характеристиками потрібно близько 10 хвилин, тоді як створення набору з 100000 елементів займає близько 140 хвилин.

Наступним етапом після створення атакувальних прикладів є навчання моделі на розпізнавання кожного елемента відповідно до заданої мітки. Табл. 3.2 і рис 3.5 демонструє результати серії навчань моделі, де НМ правильно класифікує елементи за їхніми лейблами.

Таблиця 3.2 – Тренування моделі

Серія тренувань	Кількість результатів, коли модель дала правильну відповідь
1	7441
2	8327
3	8463
4	8526
5	9411
6	9455
7	9467
8	9469
9	9481
10	9489

При збільшенні кількості кроків від 1 до 2, кількість правильних відповідей суттєво зростає. Такий результат свідчить про те, що на початкових етапах збільшення кількості кроків позитивно впливає на точність моделі. Після досягнення приблизно 7 кроків, точність стабілізується, і подальше збільшення кількості кроків не призводить до значного покращення. Отже, для ефективного навчання моделі на атакувальних прикладах достатньо використовувати близько 7 кроків.

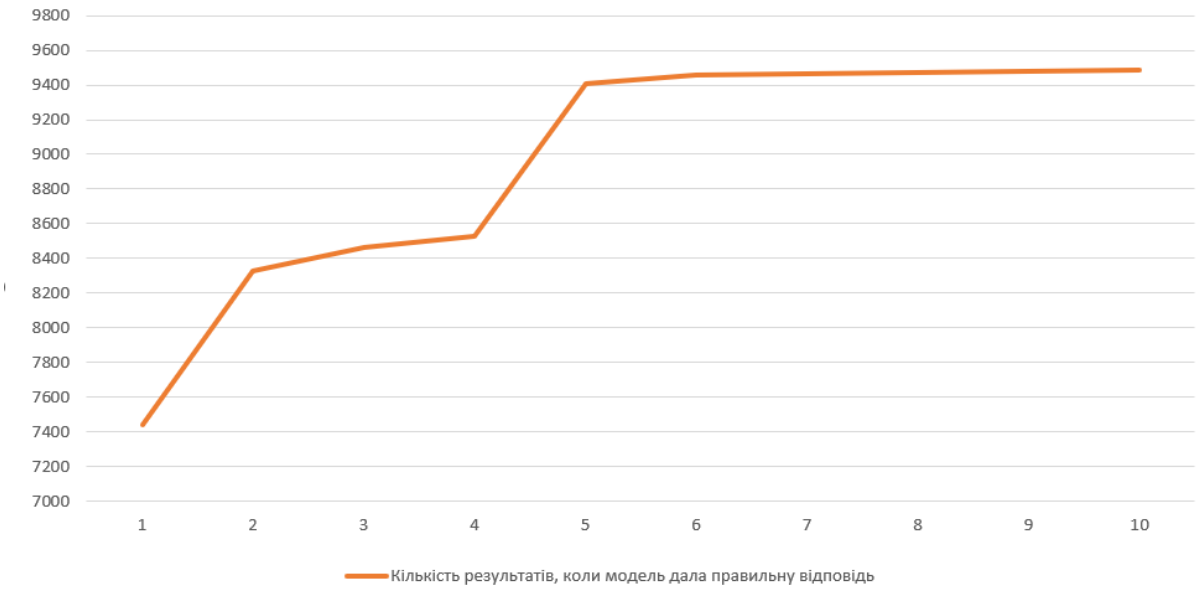


Рисунок 3.5 – Діаграма навчання моделі

За допомогою мережі, навченої на 10000 атакувальних прикладах, доданих до 50000 прикладів вихідного навчального набору, отримана точність результатів дорівнює 94,89%.

Лістинг 3.6 – Функція обчислення точності моделі

```
def accuracy(model, test_data):
    tot = float(len(test_data))
    correct = 0
    for i in range(len(test_data)):
        correct += int(np.argmax(model.feedforward(test_data[i][0])) == np.argmax(test_data[i][1]))
    return correct / tot
print('Точність: ' + str(accuracy(net2, adversarial_test_set)))
```

Досягнута точність варіюється від 90% до 95% на атакувальних прикладах, що значно перевищує результати в 50%, отримані з вихідною мережею. Цікаво, що збільшення кількості кроків під час генерації атакувальних прикладів не знижує точності, тоді як значне зменшення кількості кроків призводить до її зниження. Це пов'язано з тим, що при використанні заданого розміру кроку атакувальні приклади досягають повної конвергенції. При використанні меншої кількості кроків, атакувальні приклади сходяться лише частково.

Останнім етапом дослідження є порівняння результатів роботи оригінальної моделі та моделі, навченої на атакувальних прикладах, у задачі розпізнавання зашумлених зображень (рис. 3.6).

Результати показують, що модель, навчена на атакувальних прикладах, демонструє значно вищу стійкість до шуму. Точність розпізнавання зашумлених зображень у цієї моделі є стабільно високою навіть при високому рівні шуму, тоді як оригінальна модель значно втрачає в точності.

Крім того, аналіз результатів виявив, що навчання на атакувальних прикладах допомагає моделі краще узагальнювати інформацію і зменшує її вразливість до спотворених даних.

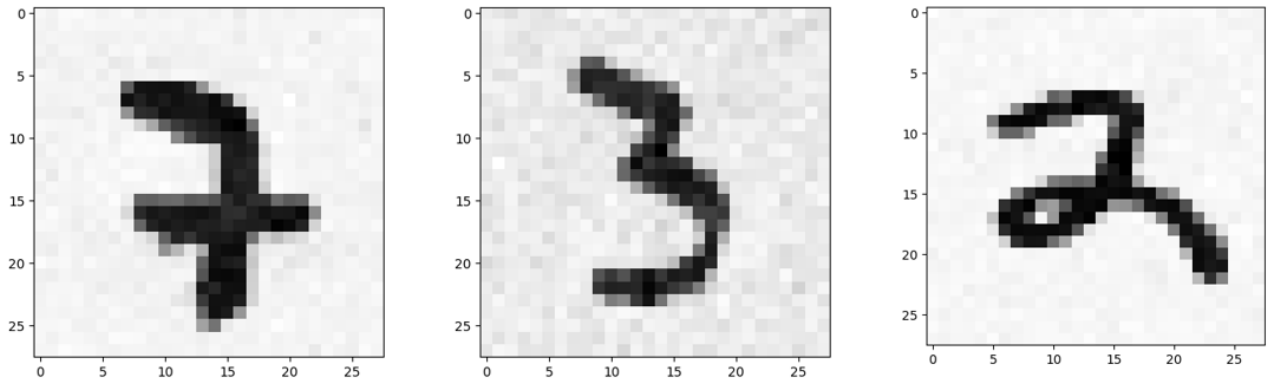


Рисунок 3.6 – Порівняння результатів

У даному розділі було проведено детальне дослідження щодо створення та використання адверсивних атак на НМ, а також захисту від таких атак. Було розроблено та реалізовано методи генерації атакувальних прикладів шляхом мінімізації функції вартості за допомогою градієнтного спуску. Окрім цього, було досліджено ефективність бінарного порогового перетворення як методу захисту від адверсивних атак та створено набір нових прикладів для подальшого навчання моделі.

Результати показали, що навчання на атакувальних прикладах допомагає моделі краще узагальнювати інформацію і зменшує її вразливість до спотворених даних.

ВИСНОВКИ

У першому розділі роботи було розглянуто широке коло питань, пов'язаних із системами штучного інтелекту та їх застосуванням. Зокрема, було проведено огляд основних типів систем ШІ, таких як машинне навчання, обробка природної мови, комп'ютерний зір та робототехніка. Окрему увагу приділено хронології розвитку цих систем та їх значущості в сучасному світі. Крім того, було детально описано архітектури штучних нейронних мереж, включаючи нейронні мережі прямого зв'язку, рекурентні нейронні мережі, згорткові нейронні мережі та генеративні змагальні мережі.

Дослідження виявило, що кожен тип системи ШІ має свої унікальні особливості та області застосування. Машинне навчання є основою для багатьох сучасних технологій ШІ, що використовуються для аналізу великих даних і самонавчання моделей. Системи обробки природної мови значно покращили взаємодію між людиною і машиною, зробивши технології більш доступними. Комп'ютерний зір і робототехніка демонструють значний прогрес у багатьох галузях, від медицини до безпеки. Різні архітектури ШНМ дозволяють вирішувати складні завдання, пов'язані з розпізнаванням образів, обробкою послідовностей і генерацією нових даних.

У другому розділі розглядався поточний стан кіберзлочинності та її зростання, особливо у галузях охорони здоров'я та урядових організаціях. Детально досліджено типи атак на нейронні мережі на рівні даних і на рівні моделі, включаючи адверсивні атаки, такі як FGSM і PGD, цільові атаки, атаки з використанням вразливостей, а також маніпуляції під час навчання моделей. Окрім цього, було висвітлено різні методи захисту НМ від цих атак.

Адверсивні атаки виявилися дуже ефективними у введенні моделей в оману шляхом додавання мінімальних збурень до вхідних даних. Використання градієнтних методів, таких як FGSM і PGD, дозволяє зловмисникам створювати атакувальні приклади, які важко виявити. Атаки на рівні моделі, такі як вбудовування бекдорів і маніпуляція процесом навчання, можуть суттєво вплинути на роботу моделей.

Проте, впровадження захисних механізмів може значно підвищити стійкість нейронних мереж до атак. Для цього рекомендується використовувати адверсивне навчання, що включає атакувальні приклади в процес навчання моделі, тим самим підвищуючи її стійкість. Ефективними є також методи моніторингу та алгоритмів фільтрації зловмисних змін на всіх етапах життєвого циклу моделі, що допомагають запобігати атакам.

Третій розділ дипломної роботи був присвячений практичній реалізації методів для розпізнавання рукописних цифр на основі даних MNIST з використанням FFNN. Також на основі теоретичних аспектів, розглянутих у попередньому розділі, створено адверсивну атаку та розроблено методи захисту від неї.

Для генерації адверсивних прикладів було розроблено методи мінімізації функції вартості за допомогою градієнтного спуску. Зокрема, використовувалися сигмоїдна функція і її похідна для обчислення градієнта похідної функції. Реалізовано функції, що обчислюють градієнти та створюють адверсивні приклади, які зловмисники можуть використовувати для введення моделей в оману. Ці приклади дозволили вивчити, як мінімальні збурення можуть вплинути на роботу НМ, спричиняючи помилкові прогнози.

Для захисту від адверсивних атак було застосовано два підходи: бінарне порогове перетворення та адверсивне навчання.

Реалізація бінаризації полягала в перетворенні кожного пікселя зображення на чорний або білий колір залежно від його початкового значення. Даний підхід дозволяє усунути шум, накладений на зображення, і забезпечує високий рівень захисту, особливо на етапі перетворення.

Для підвищення стійкості моделі було використано адверсивне навчання, яке включало додавання атакувальних прикладів до навчального набору даних. Таким чином, це дозволило моделі навчитися на зашумлених даних і ігнорувати шум, підвищуючи її стійкість до атакувальних прикладів. Було створено набір з 10000 нових зображень, що оптимально з точки зору ефективності використання ресурсів та покращення точності моделі.

Розроблені методи захисту можуть бути корисні в різних сферах, де використання НМ є критично важливим і потребують постійного вдосконалення процесів. Наприклад, у системах розпізнавання образів, автономному водінні, біометричній ідентифікації та інших додатках, де точність і безпека моделей мають вирішальне значення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. S. Herping. Securing Artificial Intelligence – Part I [Electronic resource]. 2019. Режим доступу: https://www.stiftung-nv.de/sites/default/files/securing_artificial_intelligence.pdf(27.06.2022)

2. Присяжнюк, А. (2019, February 5). Як працює machine learning та його застосування на практиці | Na chasi. Na Chasi. Режим доступу: https://nachasi.com/tech/2019/01/31/yak-pratsyuye-machine-learning/

3. Foote, K. D. (2023, May 4). A brief history of machine learning - DATAVERSITY. Режим доступу: https://www.dataversity.net/a-brief-history-of-machine-learning/

4. What is natural language processing? | IBM. Режим доступу: https://www.ibm.com/topics/natural-language-processing

5. Eliza, a chatbot therapist. Режим доступу: https://web.njit.edu/~ronkowit/eliza.html

6. Heiman, A. (2023, June 24). The perceptron explained - Alice Heiman - medium. Medium. Режим доступу: https://medium.com/@aliceheimanxyz/the-perceptron-explained-9bf9183c4999

7. Koetsier, T. (2019). The ascent of GIM, the Global Intelligent Machine. History of mechanism and machine science/History of mechanism and machine sciense. Режим доступу: https://doi.org/10.1007/978-3-319-96547-5

8. Staff, C. (2024, April 10). 4 Types of neural network architecture. from Режим доступу: https://www.coursera.org/articles/neural-network-architecture

9. MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. Режим доступу: http://yann.lecun.com/exdb/mnist/

10. Passeri, P. (2024, Березень 26). 2024 Cyber attacks statistics. Режим доступу: https://www.hackmageddon.com/2024/03/26/2024-cyber-attacks-statistics/

11. N. Carlini and H. Farid, “Evading Deepfake – Image detectors with white- and black-box attacks,” in Proc. IEEE Conf. Comput. Vis.Pattern Recognit. (CVPR’20), Seattle, WA, USA, Липень 14-19, 2020, ст. 2804–2813.

12. Feine, S. Morana, and A. Maedche, “A chatbot response genera tion system,” in Proc. Mensch und Comput. (MuC’20), Tagungsband, Magdebug, Germany, Вересень 6-9, 2020, ст. 333–341.

13. S. A. Salloum, T. Gaber, S. Vadera, and K. Shaalan, “A systematic literature review on phishing email detection using natural language processing techniques,” IEEE Access, vol. 10, ст. 65 703–65 727, 2022

ДОДАТОК А

SYSTEMS AND METHODS OF INFORMATION PROTECTION
СИСТЕМИ І МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ

УДК 004.056.5 **DOI:10.30837/rt.2023.4.215.01**

YURIY GOLIKOV, MARYNA YESINA, Ph.D. in technical sciences, OLENA KOBYLIANSKA

COMPARATIVE ANALYSIS OF ARTIFICIAL INTELLIGENCE BASED ON EXISTING CHATBOTS

Introduction

Today, artificial intelligence (AI) is rapidly gaining popularity in a variety of sectors, including the corporate world, the business community, and people's daily lives. The use of AI in areas such as medicine, banking, and government is becoming more frequent. AI facilitates data processing because it occurs without the intervention of human labor and usually ensures the accuracy of the tasks performed. According to statistics, in 2023, 35 % of companies used AI in their operations, and 90 % of organizations consider AI important to achieve competitive advantage [1].

Artificial intelligence systems also affect human everyday life, simplifying the following aspects of their activities: planning and organizing daily activities, using efficiency tools in finance, education and health spheres, etc. Thanks to it, society can use its time more efficiently by accessing fast and accurate information.

This article focuses on the analysis of the features of two leading artificial intelligence systems – Bard and ChatGPT. It includes a practical comparison of the same parameters of both systems, as well as identifying the advantages and disadvantages of each of them.

1. Overview of the ChatGPT Language Model

ChatGPT, created by OpenAI, is a text generation system that belongs to the GPT (Generative Pretrained Transformer) series. Based on a transformer architecture, this model is trained on large amounts of text data to generate data similar in writing style to human-generated text. Designed to respond to user requests, ChatGPT is suitable for use in conversational applications such as chatbots, customer service, and virtual assistants. This model has been trained on data from a variety of sources, such as online resources, books, and social media, allowing it to generate coherent and contextual text responses. To use ChatGPT, the user submits a prompt, such as a question or comment, and the model generates an answer based on the data it receives and its previous learning. One of the main advantages of ChatGPT is its ability to produce contextually relevant text. For example, when asking about fashion, the model may provide information that includes the following words: style, outfit, cut. ChatGPT can also continue the dialogue using the previous conversation as context. ChatGPT is also used for other tasks, such as answering questions, summarizing and classifying text, thanks to refinements for specific purposes. This model is part of a broader trend of using large language models for applications, which has the potential to transform the way we interact with technology and communicate with devices into a more natural and intuitive way [2].

Above, a general overview of the ChatGPT model was presented. Next, we will focus on comparing two versions of this model: ChatGPT-3, which appeared in 2020, and ChatGPT-4, released in 2023. This will allow us to determine which of these models is better suited for benchmarking with the Bard model.

ChatGPT-3 stands out for its high ability to understand and generate texts. It is trained on a wide range of internet data, which provides it with extensive knowledge. This model effectively performs many tasks, by creating original texts. However, it can give inaccurate answers and tends to be biased, especially in complex scenarios (it can "hallucinate").

ChatGPT-4, on the other hand, has improved its ability to distinguish and answer more complex questions thanks to its improved transformer architecture. The model received more training data and reduced the error rate compared to previous versions. ChatGPT-4 solves complex problems more accurately and reliably, showing a better understanding of context. Also, the following functionality was added: processing and generation of graphic images, additional utilities for processing files of more than 50 pages. However, despite the improvements, it is still prone to some bugs, and its complexity may require more resources. Table. Figure 1 shows a comparative characteristic of the presented models.

Table 1

Comparative characteristics of GPT-3 and GPT-4

Characteristics	GPT-3	GPT-4
Options	175 billion	Currently unknown
Modality	text	Text & Images
Performance	weak in solving complex problems	on the same level as a human being
Hallucinations	tendency to bias and mistakes	less biased and more stable

Let's decipher some concepts from Table. 1 Regarding this study:

1) In the context of language systems, "parameters" refer to configured internal variables or settings. A higher number of parameters indicates that the model is better suited to studying and generalizing patterns based on the data it has been trained on. GPT-3 was released with 175 billion parameters, making it one of the largest large language models (LLMs). The parameters of GPT-4 have not been officially announced, but it is safe to say that their number is well above 175 billion.

2) GPT-3 is unimodal, meaning it can only accept textual data. It can process and generate various text forms, but it cannot process images or other types of data. GPT-4, on the other hand, is multimodal. It can receive and create textual and graphical inputs and outputs, making it much more diverse. It can also perform more complex tasks that require a combination of textual and graphic modalities, such as captions, summarizing, or translating images.

3) The performance of a system is determined by its ability to respond adequately to incoming requests. This reflects how well the model captures the essence of the language and provides meaningful responses. Such performance is usually measured by criteria such as embarrassment, accuracy, and smoothness. With an increased number of parameters and advanced multimodal capabilities, GPT-4 is ahead of GPT-3 in terms of performance.

4) Hallucinations in a model are responses that make no sense or are irrelevant to the inputs received. This is because the model relies on its primary training data or knowledge to generate responses based on learned patterns. [3] notes that the probability of hallucinations in GPT-3 is between 15% and 20%. While it's currently unknown how prone GPT-4 is to hallucinations, OpenAI CEO Sam Altman says that "it hallucinates significantly less."

Considering all the arguments, we come to the conclusion: GPT-4 is superior to GPT-3 in efficiency, which is logical, given that each new generation of the model improves, correcting shortcomings and making significant improvements. For comparison with Bard, we choose the GPT-4 model because it has fewer errors in responses, has higher accuracy, and supports multimodal functions.

2. Overview of the Bard Language Model

Google's Bard API is a tool that allows developers to access and use data from a variety of sources. It uses Natural Language Processing (NLP) to extract information from various types of documents, such as websites, PDFs, and other text formats. In addition to complementing Google search, Bard can be integrated into websites, messaging platforms, or apps to provide realistic natural language answers to users' questions.

In December 2023, Google Bard was updated with the latest Gemini language model. This model, along with predecessors such as the Pathways Language Model 2 (PaLM 2) and Google's Language Model for Dialogue Applications (LaMDA), is based on the Transformers architecture developed by Google in 2017. Thanks to Transformer's open-source code, this architecture has formed the basis of numerous other generative AI tools, including the GPT-3 language model used in ChatGPT.

Bard focuses on search capabilities, trying to provide a more natural use of language queries instead of standard keywords. Its AI learns from real-world dialogues, offering not just answers but contextualized information. Bard is also designed to handle additional questions, which is a novelty in the field of search. It has features for collaboration and double-checking of results, assisting users in verifying the information received. It is also integrated with various Google apps and services,

including YouTube, Maps, Hotels, Flights, Gmail, Docs, and Drive, allowing users to use it to work with personal content.

Google Bard, with its advanced AI capabilities, offers users a number of unique features. Here are some of the key ones:

1. Integration with Google Lens to read images. Now it is possible to analyze the image, expanding its capabilities in working with dialogue text.

2. Image generation. The developers have added an image creation feature, improving the visual experience.

3. Visual information for answers. Bard is able to augment text responses with visual information for deeper understanding.

4. Extensive integration with Google services. Effective integration with Google services such as Maps, Docs, and others.

5. Plugin support. Plugins to extend its functionality, including integration with other websites and companies.

6. Saving drafts.

7. One-click chat export. The Bard Responses Export feature allows users to easily save their responses for later use [4].

3. Comparative Analysis of ChatGPT-4 and Bard

Let's look at a practical comparison of two advanced language models – OpenAI's ChatGPT-4 and Google's Bard. Both of these systems are based on the latest advances in artificial intelligence and neural networks, but at the same time offer unique features and functionality. We will focus on comparing their features, ability to perform various tasks, as well as consider their strengths and weaknesses in the context of different use cases. This comparison will help users understand which of these models is better suited to their specific needs.

The first test will involve solving a puzzle designed for children. The task is formulated as follows: "There is only one elevator in a 12-storey building. There are 2 people living on the ground floor, and the number of residents doubles on each subsequent floor. On which floor of this building is the elevator call button most often used?" Logically, most of the elevator calls occur on the ground floor. The responses of the models can be viewed in Fig. 1. From the test results, it is clear that none of the models provided the correct answer. However, in favor of ChatGPT-4, it correctly determined the number of residents on the 12th floor, while Bard settled on the calculations for the 4th floor and mistakenly listed the 5th floor in its response.

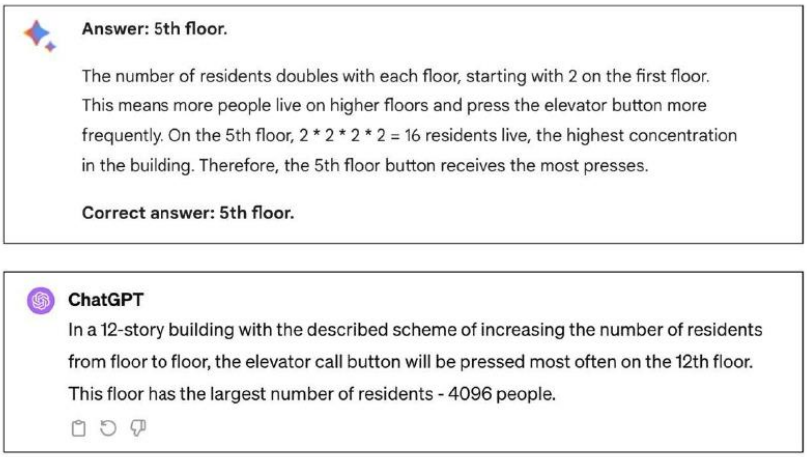


Fig. 1. Model Responses to a Logic Problem

The second test consists of creating an essay on the topic "Protection of personal data in the information space" with a limit of 2000 characters. The results of this task will be presented in Comparative Table 2. Analyzing the table, it can be noted that ChatGPT-4 complied with the set limit with a deviation of 14%, while Bard exceeded the specified volume by almost 75%. The ratio of the number of unique words to the total number is 72% for ChatGPT-4 and 57% for Bard. The average number of words per sentence for each model is 10% of the total. About a quarter of the essays created by ChatGPT-4 contain unimportant information, while in Bard this figure is a fifth of the entire text.

Table 2

Comparison of ChatGPT-4 and Bard Essays Created

	ChatGPT-4	**Bard**
Number of characters	1726	3496
Word Count	198	364
Unique words	142	206
Number of sentences	19	36
Time to read	1 min.	2 min.
"Water"	26%	19%

The third task was to improve the website of V. N. Karazin Kharkiv National University, focusing not just on the description of the sections, but on improving it for users. ChatGPT-4 offered the following solutions: updating the design of the site, developing a mobile version, creating a forum or chat for the exchange of information between entrants and students, as well as bringing the site in line with modern web security standards. Bard, on the other hand, put forward other ideas: adding sections on the history and traditions of the university, modern life at the university, famous alumni, scientific achievements, improving accessibility for people with disabilities, including enlargement of the font and adaptation for the visually impaired, background sounds, as well as adding sections with reviews, news and events, and a map of the university. Thus, the models put forward different priorities as to what is more important for the site: ensuring its security or user-friendliness and informativeness.

In the fourth test, the models were engaged in the development of HTML and CSS code for the main page of the online eyewear store. Evaluating the results, which are presented in Fig. 2 and 3, it can be noted that Bard proved to be more efficient in completing the task. On the homepage created by Bard, there were not only basic links to the products, contacts and return to the main page, but also organized product categories such as men's and women's eyewear, vision correction glasses, as well as special offers and discounts.

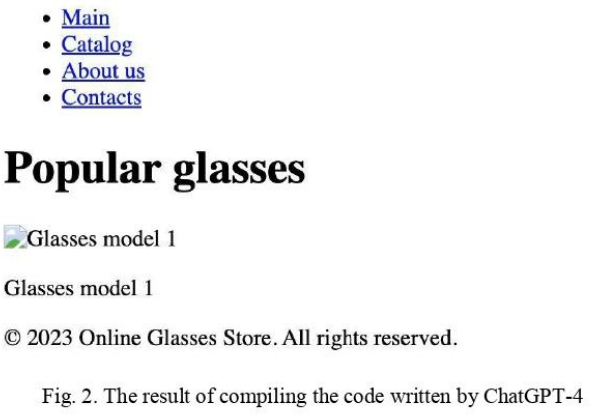


Fig. 2. The result of compiling the code written by ChatGPT-4

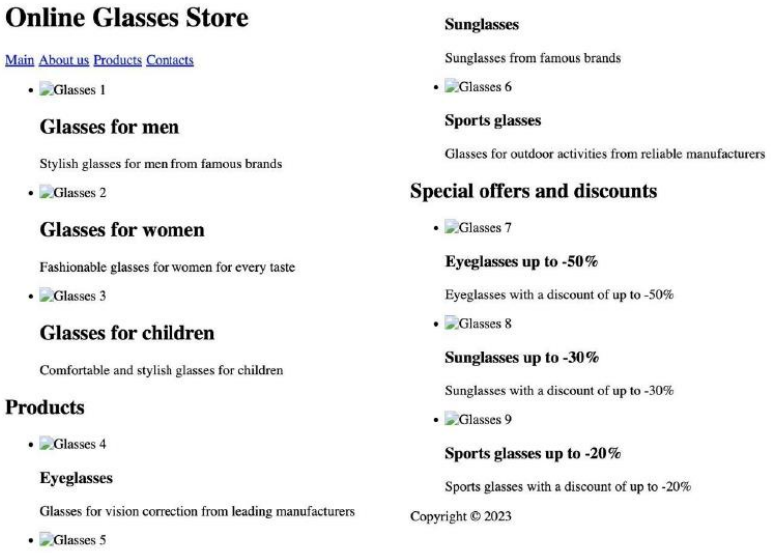


Fig. 3. The result of compiling code written by Bard

So, summing up all the test tasks performed by both models, it should be noted that the choice between them depends on the specific needs of users. Under the same conditions, both models showed different results, sometimes even deviating from their usual strengths. For example, while ChatGPT-4 is often recommended for programming, it performed less impressively than Bard in this benchmarking analysis. At the same time, Bard was unable to effectively solve a simple logical problem.

Conclusions

The article conducted a comparative analysis of the two leading models of artificial intelligence – ChatGPT-4 and Bard. As a result, it was found that the choice between the models depends on the specific needs of the user, as each of them showed different results. The advantages of ChatGPT-4, according to our research, include accurate mathematical calculations, performing tasks with minimal deviations from the conditions, as well as specific tips for improving the website. In contrast, Bard has shown a broader approach to tasks, going beyond the given conditions and offering more relevant solutions. As for the disadvantages, both models show weaknesses in logical thinking. Also, testing for image generation was not carried out due to the limitations of one of the models. Though, both systems continue to evolve and learn, so it is likely that current problems and disadvantages will be handled in the near future.

References:

1. Webster M. (October 6, 2023) 149 AI Statictics: The Present And Future Of All At Your Fingerprints. [Electronic resource]. Retrieved from https://www.authorityhacker.com/aistatistics/#:~:text=Top%20AI%20Statistics%20(Editor's%20Pick)&text=35%25%20of%20businesses%20have%20adopted,the%20global%20economy%20by%202030.

2. Md Sakibul Islam Sakib (February 2023) What is ChatGPT? [Electronic resource]. Retrieved from https://www.researchgate.net/publication/367794587_What_is_ChatGPT.

3. Ayush Kudesia (March 28, 2023) GPT 3 vs. 4: Know The Difference. [Electronic resource]. Retrieved from https://fireflies.ai/blog/gpt3-vs-4.

4. What is Bard (Google AI)? [Electronic resource]. Access mode: Everything you need to know https://instagantt.com/project-management/what-is-bard-google-ai.

Received 07.09.2023

Information about the authors:

Golikov Yuriy – Master's degree in Computer Science, CEO and Founder of DevBrother tech company (Ukraine, Poland and USA), e-mail: yuriy@devbrother.com

Yesina Maryna – Ph.D. in technical sciences, Associate Professor, Department of Security of Information Systems and Technologies, V. N. Karazin Kharkiv National University, researcher-consultant is JSC "Institute of Information Technologies" Kharkiv, Ukraine, e-mail: m.v.yesina@karazin.ua, ORCID: https://orcid.org/0000-0002-1252-7606

Kobylianska Olena – student of the Faculty of Computer Sciences of V. N. Karazin Kharkiv National University, Ukraine, e-mail: kobol1801@gmail.com

ДОДАТОК А

ISSN 2519-2310 Computer Science &Cyber Security, Issue 2(24) 2023

DOI: 10.26565/2519-2310-2023-2-03
УДК 004.056.5

ПОРІВНЯЛЬНИЙ АНАЛІЗ ШТУЧНОГО ІНТЕЛЕКТУ НА ОСНОВІ ІСНУЮЧИХ ЧАТ-БОТІВ

Кобилянська Олена[1], Єсіна Марина[1,2],Горбенко Юрій[2]
[1]Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, 61022, Україна
e-mail: kobol1801@gmail.com, ORCID:https://orcid.org/0000-0003-3405-3429 ,
e-mail: m.v.yesina@karazin.ua, ORCID: https://orcid.org/0000-0002-1252-7606
[2]АТ «ІІТ», вулиця Коломенська, 15, Харків, 61166, Україна
jsciitua@gmail.com
Надійшла до редакції 1 листопада 2023 р. Переглянута 2 грудня 2023 р. Прийнята 25 грудня 2023 р.

Анотація: У даній роботі представлено комплексний аналіз двох провідних систем штучного інтелекту (ШІ) – ChatGPT-4 від OpenAI та Bard від Google AI. Також наводиться огляд розвитку штучного інтелекту в різних галузях та його впливу на повсякденне життя людини, особливо в таких сферах, як медицина, фінанси, державне управління тощо. Проводиться заглиблення в детальне порівняння різних версій ChatGPT (GPT-3 та GPT-4), шляхом обговорення та аналізу їхніх можливостей, вдосконалення та обмежень. У статті також розглядається інтеграція системи Bard із сервісами Google, її унікальні функціональні можливості та останні оновлення. Мета дослідження полягає в порівнянні можливостей систем штучного інтелекту ChatGP-4T та Bard, висвітленні їхніх сильних і слабких сторін, а також їх практичного застосування. Проведено порівняльне тестування для оцінки продуктивності кожної моделі (системи) в різних завданнях, включаючи розв'язання логічного завдання, написання есе, аналіз із подальшим внесення пропозицій щодо покращення веб-сайту та написання коду HTML/CSS для веб-сторінки. Результати підкреслюють той факт, що, незважаючи на визнані переваги цих моделей, їхні функціональні характеристики іноді можуть бути обмежені або не відповідати очікуванням при виконанні специфічних завдань, а вибір системи (моделі) буде коригуватися у залежності від потреб користувачів.
Ключові слова: *ChatGPT-4,Bard,OpenAI,GoogleAI,штучний інтелект.*

1.Вступ

На сьогоднішній день, штучний інтелект (ШІ) швидко набуває популярності у різних секторах, включаючи корпоративний світ, бізнес-кола та повсякденне життя людей. Застосування ШІ в областях, як-от медицина, банківська сфера та урядові структури, стає все частішим. ШІ полегшує обробку даних, оскільки вона відбувається без втручання людської праці та зазвичай забезпечує точність виконаних завдань. Згідно зі статистикою, у 2023 році 35% компаній використовували ШІ у своїй діяльності, а 90% організацій вважають ШІ важливою для досягнення конкурентних переваг [1].

Системи штучного інтелекту впливають і на людське повсякдення, спрощуючи наступні аспекти їх діяльності: планування та організація денних справ, використання засобів ефективності у фінансах, навчанні та здоров'ї тощо. Завдяки йому, суспільство може ефективніше використовувати свій час, отримуючи доступ до швидкої та точної інформації.

Дана стаття зосереджена на аналізі особливостей двох провідних систем штучного інтелекту –*Bard* та *ChatGPT*. Вона включає в себе практичне порівняння однакових параметрів обох систем, а також виявлення переваг та недоліків кожної з них.

2. Огляд мовної моделі ChatGPT

ChatGPT, створена OpenAI, є системою генерації тексту, яка належить до серії *GPT* (*Generative Pretrained Transformer*). Базуючись на трансформерній архітектурі, ця модель навчена на великих масивах текстових даних для генерації даних подібних за стилем написання до тексту, створеним людиною. Розроблена для реагування на запити користувачів, *ChatGPT* підходить для використання у діалогових програмах, таких як чат-боти, обслуговування клієнтів та віртуальні асистенти. Ця модель була тренована на даних з різних джерел, таких як

ISSN 2519-2310 Computer Science &Cyber Security, Issue 2(24) 2023

Інтернет-ресурси, книги та соцмережі, що дозволяє їй створювати зв'язні та контекстуальні текстові відповіді. Щоб використовувати *ChatGPT*, користувач подає підказку, таку як питання або коментар, і модель генерує відповідь, враховуючи отримані дані та своє попереднє навчання. Однією з головних переваг *ChatGPT* є її здатність до контекстуально релевантного тексту. Наприклад, при запитанні про моду, модель може надати інформацію, що включає наступні слова: стиль, вбрання, крій. *ChatGPT* також може продовжувати діалог, використовуючи попередню розмову як контекст. *ChatGPT* також застосовується для інших задач, таких як відповіді на питання, узагальнення та класифікація тексту, завдяки доопрацюванням під конкретні цілі. Ця модель є частиною більш широкої тенденції використання великих мовних моделей для застосунків, що має потенціал перетворити спосіб взаємодії з технологіями та спілкування з пристроями на більш природній і інтуїтивно зрозумілий[2].

Вище було представлено загальний огляд моделі *ChatGPT*. Далі ми зосередимося на порівнянні двох версій цієї моделі: *ChatGPT-3*, що з'явилася у 2020 році, та *ChatGPT-4*, випущеної у 2023 році. Це дозволить нам визначити, яка з цих моделей краще підходить для порівняльного аналізу з моделлю *Bard*.

ChatGPT-3 вирізняється своєю високою здатністю до розуміння та створення текстів. Він навчений на обширному спектрі Інтернет-даних, що надає йому широкі знання. Ця модель ефективно виконує багато завдань, створюючи оригінальні тексти. Однак, вона може давати неточні відповіді та має тенденцію до упередженості, особливо у складних сценаріях (тобто, умовно кажучи - *може «галюцинувати»*).

ChatGPT-4, з іншого боку, покращив здатність розрізняти та відповідати на більш складні питання завдяки удосконаленій трансформерній архітектурі. Модель отримала більше навчальних даних і зменшила частоту помилок порівняно з попередніми версіями. *ChatGPT-4* вирішує складні завдання точніше та надійніше, показуючи краще розуміння контексту. Також, до функціоналу системи був доданий наступний функціонал: обробка та генерація графічних зображень, додаткові утиліти на обробку файлів обсягом більш, ніж 50 сторінок. Однак, попри поліпшення, вона все ще схильна до деяких помилок, і її складність може потребувати більше ресурсів. У табл. 1 наведена порівняльна характеристика поданих моделей.

Таблиця 1 – Порівняльна характеристика *GPT-3*та*GPT-4*
Table 1 – Comparative characteristic*GPT-3*&*GPT-4*

Характеристики	**GPT-3**	**GPT-4**
Параметри	*175 млрд*	*наразі невідомо*
Модальність	*текст*	*текст і зображення*
Продуктивність	*слабка у вирішенні складних задач*	*на одному рівні із людиною*
Галюцинації	*схильність до упередженості та помилок*	*менш упереджена та більш стабільна*

Пояснимо деякі поняття із табл. 1 відносно даного дослідження:

1. У контексті мовних систем, категорія «параметри» відносяться до налаштованих внутрішніх змінних або інших налаштувань. Більша кількість параметрів вказує на те, що ця модель краще пристосована до вивчення та узагальнення закономірностей на основі даних, на яких вона «навчалася». *GPT-3* була випущена з 175 мільярдами параметрів, що робить її

ISSN 2519-2310 Computer Science &Cyber Security, Issue 2(24) 2023

однією з найбільших великих моделей (*LargeLM*). Про параметри *GPT-4* офіційно не повідомлялося, але можна з упевненістю казати, що їх кількість значно перевищує 175 млрд.

2. *GPT-3* є унімодальною, тобто може приймати лише текстові дані. Вона може обробляти і генерувати різні текстові форми, але не може обробляти зображення або інші типи даних. *GPT-4* є мультимодальною, вона може приймати і створювати текстові і графічні вхідні та вихідні дані, що робить її набагато різноманітнішою. Вона, також, може виконувати більш складні завдання, які вимагають поєднання текстової та графічної вихідної інформації, такі як підписи, підбиття підсумків або переклад зображень.

3. Продуктивність системи визначається її здатністю адекватно реагувати на вхідні запити. Це відображає, наскільки успішно модель вловлює суть мови та надає значущі відповіді. Таку ефективність зазвичай вимірюють за критеріями, як: «збентеженість», «точність» і «плавність». Завдяки збільшеній кількості параметрів та розширеним мультимодальним можливостям, *GPT-4* випереджає *GPT-3* у термінах її продуктивності.

4. Галюцинації в моделі – це «відповіді», які не мають сенсу або не мають відношення до отриманих вихідних даних. Це відбувається тому, що модель покладається на свої первинні навчальні дані або знання, щоб генерувати наступні відповіді на основі вивчених шаблонів. У роботі [3]зазначається, що ймовірність галюцинацій у *GPT-3* становить від 15% до 20%. Хоча наразі невідомо, наскільки *GPT-4* схильна до галюцинацій, генеральний директор комп. *OpenAI* Сем Альтман каже, що *«вона галюцинує значно менше...»*.

Зважаючи на усі аргументи, доходимо висновку: - *GPT-4* перевершує *GPT-3* у ефективності, що є логічним, враховуючи, що кожне нове покоління моделі покращується, виправляючи недоліки та вносячи значні удосконалення. Тому, для порівняння із *Bard*, обираємо модель *GPT-4*, оскільки вона виявляє менше помилок у відповідях, має вищу точність та підтримує мультимодальні функції.

3. Огляд мовної моделі Bard

Bard API від Google– це інструмент, який дозволяє розробникам отримувати доступ до даних з різних джерел і використовувати їх. Він використовує обробку природної мови (*NLP*) для вилучення інформації з різних типів документів, таких як веб-сайти, PDF-файли та інші текстові формати. Окрім доповнення пошуку Google, *Bard* може бути інтегрований у вебсайти, платформи обміну повідомленнями або додатки для надання реалістичних відповідей природною мовою на запитання користувачів.

У грудні 2023 року Google Bard був оновлений за допомогою новітньої мовної моделі *Gemini*. Ця модель, разом із такими попередниками, як *Pathways Language Model 2 (PaLM 2)* та *Google's Language Model for Dialogue Applications (LaMDA)*, створена на основі архітектури *Transformers*, розробленої Google в 2017 році. Завдяки відкритому вихідному коду Transformer, ця архітектура лягла в основу численних інших генеративних інструментів штучного інтелекту, в тому числі мовної моделі *GPT-3*, яка використовується в *ChatGPT*.

Bard зосереджений на пошукових можливостях, намагаючись забезпечити більш природне використання мовних запитів замість стандартних ключових слів. Його штучний інтелект навчається на основі реальних діалогів, пропонуючи не просто відповіді, а контекстуалізовану інформацію. *Bard* розроблено також для обробки додаткових запитань, що є новинкою у сфері пошуку. Має функції для спільної роботи та подвійної перевірки результатів, допомагаючи користувачам у перевірці отриманої інформації. Він також інтегрований з різними додатками та сервісами Google, включаючи *YouTube, Maps, Hotels, Flights, Gmail, Docs* та *Drive*, дозволяючи користувачам використовувати його для роботи з особистим контентом.

ISSN 2519-2310 Computer Science &Cyber Security, Issue 2(24) 2023

GoogleBard, з його розширеними можливостями штучного інтелекту, пропонує користувачам ряд унікальних функцій. Ось деякі з ключових:

1. Інтеграція з *Google Lence* для читання зображень. Тепер став можливий аналіз зображення, розширюючи свої можливості у роботі з діалоговим текстом.

2. Генерація зображень. Розробники додали функцію створення зображень, покращуючи візуальні можливості.

3. Візуальна інформація до відповідей. *Bard* здатен доповнювати текстові відповіді візуальною інформацією для глибшого розуміння.

4. Широка інтеграція з сервісами Google. Ефективне інтегрування з Google сервісами, такими як карти, документи, таблиці та інші.

5. Підтримка плагінів. Плагіни для розширення своєї функціональності, включаючи інтеграцію з іншими веб-сайтами і компаніями.

6. Збереження чернеток. Вирішена проблема втрати відповідей при оновленні вкладки.

7. Експорт чату одним кліком. Функція експорту відповідей у *Bard* дозволяє користувачам легко зберігати свої відповіді для подальшого використання [4].

4. Порівняльний аналіз ChatGPT-4 та Bard

Розглядаємо практичне порівняння двох передових мовних моделей *–ChatGPT-4* від *OpenAI* і *Bard* від Google. Обидві ці системи базуються на найновіших досягненнях у галузі штучного інтелекту та нейронних мереж, але водночас пропонують унікальні характеристики та функціональності. Зосередимося на порівнянні їхніх особливостей, здатності до виконання різноманітних завдань, а також розглянемо їхні сильні та слабкі сторони в контексті різних сценаріїв використання. Це порівняння допоможе користувачам зрозуміти, яка з цих моделей краще підходить для їхніх конкретних потреб.

Перший тест включатиме вирішення головоломки для підлітків. Задача формулюється так: «*У 12-поверховій будівлі є лише один ліфт. На нижньому поверсі проживає 2 людини, і на кожному наступному поверсі кількість мешканців подвоюється. На якому поверсі цього будинку найчастіше використовують кнопку виклику ліфта?*». Логічно, що найбільше викликів ліфта відбувається на 1-му поверсі. Відповіді цих моделей, наведено на рис. 1.

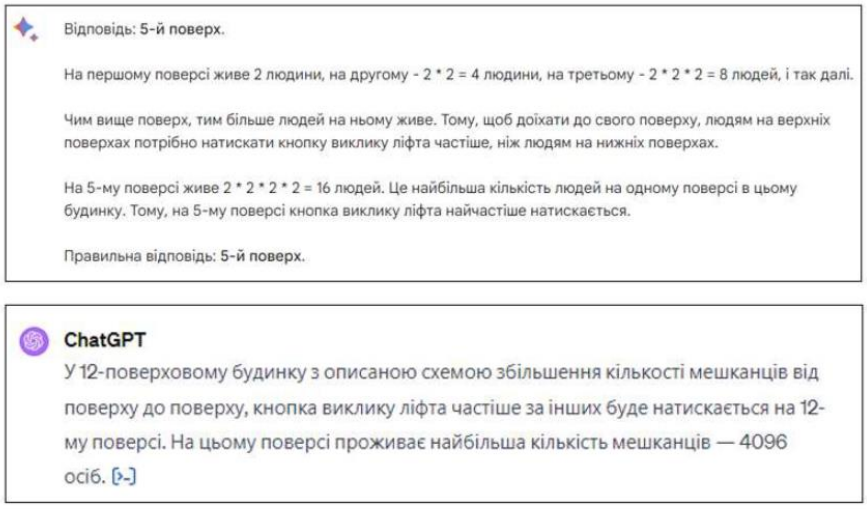


Рис. 1 – Відповіді моделей на логічну задачу
Fig.1 – Answers of models to a logical problem

ISSN 2519-2310 Computer Science &Cyber Security, Issue 2(24) 2023

З результатів тестування слід, що жодна з моделей не надала правильної відповіді. Проте, на користь *ChatGPT-4*, він правильно визначив кількість жителів на 12-му поверсі, тоді як *Bard* зупинився на обчисленнях для 4-го поверху та помилково зазначив 5-й поверх у своїй відповіді.

Другий тест полягатиме у створенні есе на тему «Захист особистих даних в інформаційному просторі» з обмеженням у 2000 символів. Результати цього завдання будуть представлені у порівняльній табл.2. Аналізуючи таблицю, слід відзначити, що *ChatGPT-4* дотримався встановленого ліміту з відхиленням у 14%, в той час як *Bard* перевищив заданий обсяг майже на 75%. Відношення кількості унікальних слів до загальної кількості складає 72% для *ChatGPT*-4 і 57% для *Bard*. Середня кількість слів на речення у кожної моделі становить 10% від загального обсягу. Приблизно чверть, створеного *ChatGPT-4*, містить неважливу інформацію, в той час як у *Bard* цей показник становить п'яту частину від усього тексту.

Таблиця 2 – Порівняння створених есе *ChatGPT-4*та*Bard*
Table 2 – Comparison of generated essays *ChatGPT-4*&*Bard*

Параметр	ChatGPT-4	Bard
Кількість символів	*1726*	*3496*
Кількість слів	*198*	*364*
Унікальних слів	*142*	*206*
Кількість речень	*19*	*36*
Час на прочитання	*1 хв.*	*2 хв.*
Неважлива інформація	*26%*	*19%*

Сутність 3-го завдання полягала у вдосконаленні веб-сайту Харківського національного університету імені В. Н. Каразіна, зосереджуючись не просто на описі розділів, а на його поліпшенні для користувачів. В результаті *ChatGPT-4* запропонував такі рішення: - оновлення дизайну сайту;- розробка мобільної версії; - створення форуму чи чату для обміну інформацією між абітурієнтами й студентами;- приведення сайту у відповідність з сучасними стандартами веб-безпеки. *Bard* же висунув інші «ідеї»: - додавання розділів про історію і традиції університету, сучасне життя в університеті, відомих випускників та наукові досягнення; - покращення доступності для людей з обмеженими можливостями, включаючи збільшення шрифту і адаптацію для людей із вадами зору; - фонові звуки; - додавання розділів з відгуками, новинами та подіями й картою університету. Таким чином, моделі висунули помітно різні пріоритети, стосовно того, що важливіше для сайту: - забезпечення його безпеки або зручність та інформативність для його користувачів.

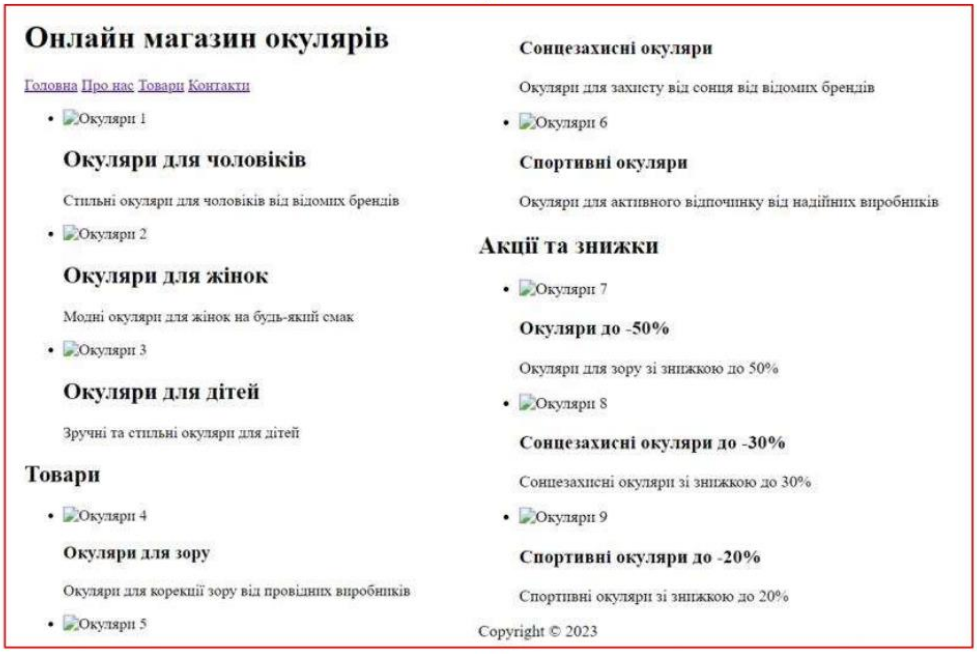
У 4-му тесті, обрані моделі займалися розробкою HTML та CSS коду для головної сторінки умовного Інтернет магазину окулярів. Оцінюючи результати, які представлені на рис. 2, можна відзначити, що *Bard* виявився більш ефективним у виконанні завдання. На головній сторінці, створеній саме *Bard*, були не тільки основні посилання на асортимент, контакти та повернення на головну сторінку, але й впорядковані категорії товарів, такі як чоловічі й жіночі окуляри, окуляри для корекції зору, а також спеціальні пропозиції і знижки.

Отже, підбиваючи підсумки всіх тестових завдань, виконаних обома моделями, слід зазначити, що вибір між ними залежатиме від специфічних потреб користувачів. Так, за однакових умов обидві моделі показали різні результати, іноді навіть відступаючи від своїх звичайних «сильних» сторін. Наприклад, хоча *ChatGPT-4* часто рекомендується для виконання

ISSN 2519-2310 Computer Science &Cyber Security, Issue 2(24) 2023

завдань програмування, у цьому порівняльному аналізі виконання тестових завдань, він показав менш значущі результати, ніж *Bard*. Водночас, *Bard* не зміг ефективно впоратися із простими завданнями на розв'язання звичайної логічної задачі.

a)



b)

Рис.2– Результати компіляції коду,створеного *ChatGPT-4 (a)* та *Bard (b)*
Fig. 2 – Results of compiling the code generated by*ChatGPT-4 (a)*&*Bard (b)*

5.Висновки

У роботі представлений порівняльний аналіз роботи двох провідних моделей штучного інтелекту – *ChatGPT-4*та*Bard*. В результаті виконання низки тестових завдань було підтверджено, що вибір між необхідною моделлю, залежить від конкретних потреб її користувачів, оскільки кожна з них демонструє помітно різні результати.

До переваг *ChatGPT-4* (за проведеними дослідженнями) слід віднести точні математичні розрахунки, виконання задач з мінімальними відхиленнями від умов, а також конкретні поради для поліпшення веб-сайту. На відміну від нього, *Bard* підтвердив більш широкий підхід до завдань, виходячи за рамки заданих умов та пропонуючи користувачам більш актуа-

льні (варіативні) рішення. Щодо недоліків, то обидві моделі демонструють певні «слабкості» в алгоритмах «логічного мислення». Також, тестування на генерацію зображень не проводилося через обмеження однієї з моделей, проте обидві системи продовжують безперервно розвиватися й навчатися, що скоріш за все, буде реалізовано в найближчому майбутньому.

References

[1] Webster M. (October 6, 2023) 149 AI Statictics: The Present And Future Of All At Your Fingerprints. authorityhacker.com/ai-statistics/

[2] Md Sakibul Islam Sakib (February 2023) What is ChatGPT? http://surl.li/pqywz

[3] Ayush Kudesia (March 28, 2023) GPT 3 vs. 4: Know The Difference https://fireflies.ai/blog/gpt3-vs-4.

[4] What is Bard (Google AI)? Everything you need to know https://instagantt.com/project-management/what-is-bard-google-ai.

Submitted November 1, 2023; Revised December 2, 2023; Accepted December 25, 2023

Authors:

Kobylianska Olena, CSD Student, Department of Security of Information Systems and Technologies, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: kobol1801@gmail.com
ORCID: https://orcid.org/0000-0003-3405-3429

Yesina Maryna, Ph.D., Associate Professor, Department of Security of Information Systems and Technologies, V. N. Karazin Kharkiv National University, Ukraine.
E-mail:m.v.yesina@karazin.ua
ORCID: https://orcid.org/0000-0002-1252-7606

Yurii Gorbenko, Ph.D.,firstdeputychiefdesignerof JSC"IIT", Kharkiv, Ukraine.
E-mail:jsciitua@gmail.com

Comparative analysis of artificial intelligence based on existing ChatBots.

Abstract.This paper presents a comprehensive analysis of two leading artificial intelligence (*AI*) systems – *ChatGPT-4* from *OpenAI* and *Bard* from *Google AI*. It also provides an overview of the development of artificial intelligence in various fields and its impact on human daily life, especially in areas such as medicine, finance, public administration, etc. A detailed comparison of different versions of *ChatGPT* (GPT-3 and *GPT-4*) is carried out by discussing and analyzing their capabilities, improvements, and limitations. The article also discusses the integration of the *Bard* system with Google services, its unique functionality, and the latest updates. The purpose of the study is to compare the capabilities of *ChatGP-4T* and *Bard AI* systems, highlight their strengths and weaknesses, as well as their practical application. Comparative testing was conducted to evaluate the performance of each model (*system*) in various tasks, including solving a logical problem, writing an essay, analyzing followed by making suggestions for improving the website and writing *HTML/CSS* code for a web page. The results highlight the fact that, despite the recognized advantages of these models, their functional characteristics may sometimes be limited or not meet expectations when performing specific tasks, and the choice of system (*model*) will be adjusted depending on the needs of users.

Keywords: *ChatGPT-4,Bard, OpenAI, GoogleAI, Artificial Intelligence.*