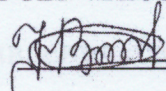


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

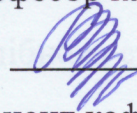
до випускної кваліфікаційної роботи магістра
на тему: «Методи та технології запобігання
фішинговим атакам електронною поштою на основі
штучного інтелекту»

Виконав: студентка 2-го курсу групи ЗКБм-24-1
спеціальності 125 «Кібербезпека та захист інформації»



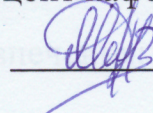
Наталія БОБЕР

Керівник: професор кафедри КІ та КБ, д.т.н., доцент



Володимир ВОРОТНИКОВ

Рецензент: доцент кафедри КІ та КБ, к.т.н., доцент



Олександр МАЄВСЬКИЙ

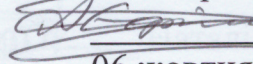
Житомир - 2025

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека та захист інформації»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій ЄФІМЕНКО

06 жовтня 2025 р.

ЗАВДАННЯ

на випускну кваліфікаційну роботу магістра

Студент: Бобер Наталія Вікторівна

Тема роботи: **«Методи та технології запобігання фішинговим атакам
електронною поштою на основі штучного інтелекту»**

затверджена Наказом університету від 06 жовтня 2025 р. №566/с

Термін здачі студентом закінченої роботи 22 грудня 2025 р.

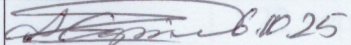
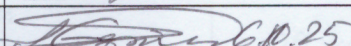
Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

1. Методи та технології виявлення електронних листів, написаних штучним інтелектом
2. Процес виявлення AI-подібного стилю у електронних листах

Обладнання та програмне забезпечення:

1. ПК з операційною системою Windows
2. Python
3. Бібліотеки Python
4. Google Gmail API

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Андрій ЄФІМЕНКО	 6.10.25	 6.10.25
2	Андрій ЄФІМЕНКО	 6.10.25	 6.10.25
3	Андрій ЄФІМЕНКО	 6.10.25	 6.10.25

Студент  Наталія БОБЕР
 Керівник  Володимир ВОРОТНИКОВ

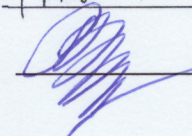
3	Джерел		Виконано
4	Дослідження соціальної та фінансової	11 листопада 2025	Виконано
5	Дослідження штучного інтелекту та його методів для написання фінансових листів	12 листопада 2025 – 19 листопада 2025	Виконано
6	Огляд методів виявлення тексту, згенерованого за допомогою штучного інтелекту	20 листопада 2025 – 27 листопада 2025	Виконано
7	Розробка програмної реалізації для виявлення AI-подібного стилю у електронних листах	28 листопада 2025 – 12 грудня 2025	Виконано
8	Аналіз отриманих результатів	13 грудня 2025	Виконано
9	Підготовка та оформлення пояснювальної записки відповідно до вимог	14 грудня 2025 – 18 грудня 2025	Виконано
10	Подання роботи на рецензування	22 грудня 2025	Виконано

Студент  Наталія БОБЕР
 Керівник  Володимир ВОРОТНИКОВ

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	19 вересня 2025	Виконано
2	Розробка плану роботи	22 вересня 2025 – 20 жовтня 2025	Виконано
3	Опрацювання літературних джерел	20 жовтня 2025 – 31 жовтня 2025	Виконано
4	Дослідження соціальної інженерії та фішингу	31 жовтня 2025 – 11 листопада 2025	Виконано
5	Дослідження штучного інтелекту та його методів для написання фішингових листів	12 листопада 2025 – 19 листопада 2025	Виконано
6	Огляд методів виявлення тексту, згенерованого за допомогою штучного інтелекту	20 листопада 2025 – 27 листопада 2025	Виконано
7	Розробка програмної реалізації для виявлення AI-подібного стилю у електронних листах	28 листопада 2025 – 12 грудня 2025	Виконано
8	Аналіз отриманих результатів	13 грудня 2025	Виконано
9	Підготовка та оформлення пояснювальної записки відповідно до вимог	14 грудня 2025 – 18 грудня 2025	Виконано
10	Подання роботи на рецензування	22 грудня 2025	Виконано

Студент  Наталія БОБЕР

Керівник  Володимир ВОРОТНІКОВ

РЕФЕРАТ

Випускна кваліфікаційна робота магістра містить 75 сторінок, 24 рисунки, список використаних джерел із 39 найменувань.

Метою випускної кваліфікаційної роботи є розробка програмної реалізації, яка допоможе виявляти AI-подібний стиль у фішингових електронних листах.

В ході дослідження було проведено навчання та тестування моделі машинного навчання на основі двох наборів фішингових електронних листів – згенерованих штучним інтелектом та написаних людиною, для автоматичного виявлення AI-подібного стилю у фішингових електронних листах.

Ключові слова: ФІШИНГ, ШТУЧНИЙ ІНТЕЛЕКТ, ЕЛЕКТРОННА ПОШТА, МАШИННЕ НАВЧАННЯ.

ABSTRACT

The master's thesis consists of 75 pages, 24 figures, and a list of 39 references.

The purpose of the thesis is to develop software that will help detect AI-like style in phishing emails.

During the research, a machine learning model was trained and tested based on two sets of phishing emails—generated by artificial intelligence and written by humans—to automatically detect AI-like style in phishing emails.

Keywords: PHISHING, ARTIFICIAL INTELLIGENCE, E-MAIL, MACHINE LEARNING.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. СОЦІАЛЬНА ІНЖЕНЕРІЯ ТА ФІШИНГОВІ АТАКИ В ЕПОХУ ШТУЧНОГО ІНТЕЛЕКТУ	12
1.1 Поняття та сутність соціальної інженерії	12
1.2 Етапи проведення атак соціальної інженерії	15
1.3 Методи соціальної інженерії	17
1.4 Електронна пошта як один із основних каналів соціально-інженерних атак	22
1.5 Фішинг – найпоширеніший метод соціальної інженерії	23
1.6 Вплив штучного інтелекту на розвиток фішингу	26
Висновки до розділу 1	30
РОЗДІЛ 2. ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У ФІШИНГОВИХ АТАКАХ ТА МЕТОДИ ЇХ ВИЯВЛЕННЯ	32
2.1 Сутність та основні риси штучного інтелекту	32
2.2 Основні характеристики штучного інтелекту [17]:	34
2.3 Створення фішингових листів за допомогою штучного інтелекту	35
2.3.1 ChatGPT	36
2.3.3 DeepAI	39
2.4 Лінгвістичний аналіз та обробка природної мови	40
2.5 Моделі машинного та глибокого навчання	42
2.6 Часовий та мережевий аналіз	43
Висновок до розділу 2	43
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВИЯВЛЕННЯ ЙМОВІРНОСТІ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ЕЛЕКТРОННИХ ЛИСТАХ	45
3.1 Основне завдання програмної реалізації	45
3.2 Середовище розробки	46
3.3 Структура програмної реалізації та її функціональне призначення	47
3.4 Використані програмні бібліотеки та модулі Python	48
3.5 Набори даних для машинного навчання	49
3.6 Тестування програмної реалізації	51
Висновок до розділу 3	56
ВИСНОВКИ	58

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	60
ДОДАТКИ.....	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІІІ – штучний інтелект

ПЗ – програмне забезпечення

ML – machine learning

AI – artificial intelligence

ВСТУП

Фішингові атаки є одними з найпоширеніших, найдорожчих та найшкідливіших загроз у сфері кібербезпеки, з якими стикаються організації та окремі особи у всьому світі. Найпоширенішим методом, який використовують шахраї надсилаючи фальшиві повідомлення для збору даних є фішингові електронні листи. З розвитком технологій та появою штучного інтелекту складність фішингових атак зростає, оскільки вони можуть базуватися на автоматично згенерованих листах, що ускладнює їх виявлення. Тобто, замість одного формату електронного листа, що надсилається великій кількості одержувачів, генеративний штучний інтелект може бути використаний для надсилання кожній потенційній жертві окремого електронного листа, що ускладнює для систем кібербезпеки ідентифікацію електронного листа, перш ніж він досягне одержувача.

Актуальність теми полягає в тому, що стрімкий розвиток технологій штучного інтелекту відкриває нові можливості для автоматизації різних процесів, зокрема в сфері кібербезпеки. Однак, одночасно з перевагами виникають нові загрози. Однією з найбільш небезпечних проблем є використання ШІ для здійснення атак соціальної інженерії, таких як фішинг, вішинг та інші. Враховуючи швидкий розвиток технологій ШІ, зростаючу кількість таких атак та складність їх виявлення, вивчення цієї теми є надзвичайно важливим для розуміння нових підходів до кіберзагроз і формування ефективних стратегій захисту. Тож дослідження методів автоматичного аналізу фішингових повідомлень згенерованих штучним інтелектом є надзвичайно актуальним та має важливе практичне значення для підвищення рівня інформаційної безпеки.

Метою кваліфікаційної роботи є розробка програмної реалізації, для автоматичного аналізу електронної пошти з метою виявлення AI-подібного стилю у фішингових повідомлення на основі методу машинного навчання.

Об'єктом дослідження є процес розробки програмних засобів автоматичного аналізу фішингових електронних повідомлень, зокрема тих, що згенеровані з використанням штучного інтелекту.

Предметом дослідження є методи машинного навчання та лінгвістичні ознаки, що використовуються для автоматичного виявлення AI-подібного стилю у фішингових електронних листах.

Визначена мета обумовлює наступні завдання:

- Проаналізувати сучасні підходи до виявлення фішингових атак та особливості використання штучного інтелекту в соціальній інженерії.
- Підготувати набори даних фішингових електронних листів, згенерованих штучним інтелектом та написаних людиною.
- Розробити програмну реалізацію для автоматичного аналізу електронних повідомлень з використанням методу машинного навчання.
- Провести експериментальне навчання та тестування моделі машинного навчання з метою оцінювання ефективності виявлення AI-подібного стилю у фішингових листах.

У процесі виконання кваліфікаційної роботи було розроблено програмний проект для автоматичного виявлення AI-подібного стилю у фішингових електронних листах з використанням методу машинного навчання, який можна застосовувати для аналізу вхідних повідомлень електронної пошти.

РОЗДІЛ 1. СОЦІАЛЬНА ІНЖЕНЕРІЯ ТА ФІШИНГОВІ АТАКИ В ЕПОХУ ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Поняття та сутність соціальної інженерії

Соціальна інженерія — це техніка маніпуляції, яку використовують кіберзлочинці, щоб обманом змусити людей розголошувати конфіденційну інформацію або виконувати дії, що ставлять під загрозу безпеку. Вона використовує людську психологію, а не технічні вразливості, часто використовуючи фішинг, приманювання під прикриттям або тактику цькування для отримання несанкціонованого доступу до систем, даних або фізичних місць розташування. [1]

Соціальна інженерія використовує ключові аспекти людської психології, впливаючи на такі риси, як довіра, страх, цікавість та терміновість. Вона маніпулює людьми, змушуючи їх розкривати конфіденційну інформацію або вживати заходів, які можуть поставити під загрозу їхню безпеку. Зловмисники створюють ситуації, які затьмарюють судження та перешкоджають раціональному прийняттю рішень, розуміючи, як люди реагують під тиском або коли стикаються зі спокусливими можливостями.

Наприклад, соціальний інженер може створити відчуття терміновості, видаючи себе за довірену авторитетну фігуру та представляючи сценарій, який викликає страх або занепокоєння. Це може спонукати жертву діяти швидко, не перевіряючи повністю ситуацію. Така психологічна маніпуляція використовує природну людську схильність допомагати, підкорятися або реагувати.

Соціальна інженерія кардинально змінилася з різних хитрощів, які використовували шахраї та аферисти. З новими технологіями ці методи стали складнішими та поширенішими. Спочатку соціальна інженерія спиралася на особисту взаємодію, щоб обманом виманити у людей гроші чи інформацію.

Тепер, з розвитком Інтернету, ця тактика перемістилася в онлайн. Цей зсув дозволив злочинцям використовувати анонімність та широкий охоплення цифрового спілкування.

Зростання соціальної інженерії показує, наскільки просунулися технології та наскільки люди й організації залежать від цифрових інформаційних систем. Вкрай важливо розуміти та передбачати нові стратегії, які можуть використовувати зловмисники. Оскільки тактика соціальної інженерії стає все більш витонченою, вона становить постійну загрозу, яка вимагає постійної обізнаності та захисних заходів.

Отже, соціальна інженерія — це метод психологічної маніпуляції, який використовує людську природу та моделі поведінки для отримання несанкціонованого доступу до систем, даних або ресурсів.

Тактики та методи соціальної інженерії ґрунтуються на науці про людську мотивацію. Вони маніпулюють емоціями та інстинктами жертв таким чином, що, як доведено, люди вдаються до дій, які не відповідають їхнім інтересам.

Більшість атак соціальної інженерії використовують одну або декілька з наступних тактик [1]:

- Ключові психологічні тригери: Авторитет — зловмисники видають себе за авторитетних осіб, таких як керівники чи ІТ-персонал, щоб змусити жертв виконати запити. Наприклад, зловмисник може видати себе за генерального директора, який запитує термінове схвалення банківського переказу.
- Терміновість: створення штучного тиску в часі змушує приймати швидкі, непродумані рішення. Зловмисник може заявити: «Ваш обліковий запис буде видалено через годину, якщо ви не підтвердите свої облікові дані зараз».

- **Страх/Залякування:** Погрози негативними наслідками маніпулюють жертвами, змушуючи їх вживати небезпечних дій. Наприклад, зловмисник може заявити: «Ваша система заражена — негайно натисніть тут, або ризикуєте втратити дані».
- **Довіра:** Побудова взаєморозуміння та створення враження легітимності допомагає обійти звичайний скептицизм щодо безпеки.

Методи психологічної маніпуляції [1]:

- **Взаємність:** Коли зловмисники спочатку надають щось цінне, жертви відчують себе зобов'язаними відповісти тим самим. Наприклад, хакер може надіслати «безкоштовний інструмент аудиту безпеки», який є шкідливим програмним забезпеченням, розраховуючи на те, що одержувач буде змушений його використовувати, оскільки отримав щось «корисне».
- **Соціальний доказ:** Люди стежать за діями інших, особливо в невизначених ситуаціях. Зловмисники використовують це, створюючи фальшиві сценарії, які показують, що інші виконують їхні прохання. Вони можуть надсилати фішингові електронні листи, в яких стверджується, що «90% ваших колег вже оновили свої паролі», щоб змусити ціль наслідувати їхній приклад.
- **Дефіцит:** Створення штучних обмежень призводить до термінових, емоційних реакцій, а не до логічних. Зловмисник може заявити: «Залишилося лише 2 місця для цього оновлення безпеки» або «Термін дії цього спеціального доступу закінчується через 24 години», щоб змусити приймати поспішні рішення.
- **Відданість та послідовність:** Щойно люди здійснюють невелику дію, вони з більшою ймовірністю продовжуватимуть таку поведінку, щоб вона виглядала послідовною. Зловмисники починають з незначних запитів, перш ніж переходити до більш конфіденційних. Спочатку вони можуть

запитувати публічну інформацію про компанію, а потім поступово переходити до запити конфіденційних даних.

- Авторитет: Окрім простого видавання себе за авторитетних осіб, зломисники використовують певні методи, такі як: використання офіційних доменів електронної пошти та підписів; посилення на внутрішні процедури або системи; відмова від імен фактичних керівників або відділів; створення штучних ієрархій або ланцюгів схвалення.
- Симпатія: Зломисники будують взаєморозуміння через: віддзеркалення стилів та уподобань спілкування; пошук або вигадкування спільних інтересів та походження; стратегічне використання лестощів та визнання; представляючи себе як корисних вирішувачів проблем.
- Маніпуляція шляхом відволікання уваги: люди приймають гірші рішення щодо безпеки, коли перебувають у стресі або когнітивному навантаженні. Зломисники можуть: створювати штучні кризи, що вимагають негайних дій; перевантажують цільову аудиторію технічним жаргоном або складними інструкціями; атакують у періоди підвищеного навантаження, такі як кінець кварталу.

Ці методи особливо ефективні, оскільки вони використовують фундаментальні психологічні моделі людини, які зберігаються навіть тоді, коли люди про них усвідомлюють.

1.2 Етапи проведення атак соціальної інженерії

Атаки соціальної інженерії рідко бувають спонтанними. Це добре продумані операції, які зазвичай розгортаються у кілька фаз, кожен з яких є критично важливим та вирішальним для успіху зломисника [2]:

1. Перша фаза: Дослідження та розвідка – На початковій стадії зломисник проводить детальне вивчення (збирання інформації) про свою потенційну мішень – особу або установу. Його завдання зібрати якомога більше

відомостей: від внутрішньої структури компанії та службових обов'язків персоналу до їхніх особистих зацікавлень, характеру поведінки, активності у соцмережах. Для цього залучають відкриті публічні джерела (OSINT). Чим ширший обсяг подробиць вдасться накопичити, тим більш правдоподібною, вірогіднішою та персоналізованою буде наступна атака соціальної інженерії.

2. Друга фаза: Планування сценарію нападу – Після накопичення всіх необхідних даних, зломисник розробляє детальний план нападу. Він обирає найбільш слушний метод і готує конкретні засоби впливу, тобто створює конкретні повідомлення, телефонні дзвінки або навіть елементи фізичної взаємодії, націлені на використання слабких місць цільових осіб. Наприклад, якщо ціллю нападу є відділ фінансів, може бути створено фейковий “рахунок-фактуру” від відомого партнера, який вимагає негайної оплати.
3. Третя фаза: Гачок та маніпуляція – Застосовуючи психологічні тригери, зломисник формує проблему, яка змушує ціль здійснити певні дії, як-от надання облікових даних або переказ коштів.
4. Четверта фаза: Реалізація та експлуатація – На цьому етапі відбувається безпосередня реалізація атаки. Зломисник ініціює контакт із жертвою підготовленими шляхами – електронною поштою, через месенджери, SMS або телефонним дзвінком. У деяких випадках фішингових атак, процес виконання може бути частково механізований: жертва переходить за шкідливим посиланням, що перенаправляє на фальшивий вебсайт, або активує заражений файл зі шкідливим кодом. У складніших сценаріях, зломисники можуть активно спілкуватися зі своєю жертвою особисто, підтримуючи свою вигадану історію та поступово зміцнюючи довіру для досягнення бажаного результату, аби отримати необхідні відомості – паролі чи доступ до системи.

1.3 Методи соціальної інженерії

Зі зростання цифрової ери інформаційних технологій розвиваються різноманітні види соціальної інженерії, наступні методи є одними з найбільш уживаних в атаках соціальної інженерії, за даними Інститут InfoSec та аналітики останніх років [2]:

- a) Фішинг (Phishing)
- b) Водопій (Watering Hole)
- c) Атака китобоя (Whaling Attack)
- d) Під приводом (Pretexting)
- e) Атаки приманка та послуга за послугу (Baiting and Quid Pro Quo)

Фішинг – це один із найрозповсюдженіших методів соціально-інженерних атак. Сутність цього методу полягає у розсиланні фальшивих повідомлень, переважно електронних листів, а також SMS, повідомлень у месенджерах чи соцмережах, що маскуються під справжні й надходять нібито від надійних джерел. Метою таких повідомлень є отримання конфіденційної інформації, зокрема паролів, банківських реквізитів або спонукання користувача перейти за шкідливим посиланням чи завантажити шкідливий файл. Щоб підвищити ефективність атаки, зловмисники часто грають на емоціях жертви: створюють відчуття терміновості – «ваш акаунт/рахунок заблоковано», провокують цікавість – «вітаємо, ви виграли крутий приз», просять про допомогу викликаючи співчуття «терміново потрібна ваша допомога». Для правдоподібності такі повідомлення ретельно стилізовані: використовуються підроблені логотипи, імітується фірмовий стиль та адреси відправників, для видачі себе за відому компанію, банк, державну установу або навіть колег. Згідно статистики, фішинг є одним із ключових чинників успішних кібератак.

Атака типу «водопій» є прикладом непрямой соціально-інженерної атаки, яка передбачає компрометацію ресурсу, що регулярно відвідує їхня потенційна

жертва. Наприклад, якщо якщо мішенню є співробітники ІТ-компанії, зловмисники можуть скомпрометувати популярний технічний форум розробників. Після успішного взламу сторінка зазвичай встановлює шкідливий код, троян чи бекдор на пристрої відвідувачів, що надає зловмисникові змогу отримати віддалений несанкціонований доступ до системи. Такий тип атак часто виконують досвідчені хакери, які використовують експлоїт нульового дня, що дозволяє залишати таку атаку непоміченою протягом тривалого часу.

Китобійна атака являє собою цілеспрямований різновид фішингу. Її мішенню є так звана «велика риба» : керівники компаній, фінансові директори, топ-менеджери, мережеві адміністратори, тобто ті, хто має привілейований доступ до критично важливих ресурсів або доступ до конфіденційної інформації. Особливістю таких атак є ретельна підготовка та досконале вивчення жертви, аби створити максимально персоналізовані та правдоподібні повідомлення. Такі листи, як правило, часто маскуються під звернення від колеги чи керівника, вони містять прохання про негайну дію, що створює відчуття терміновості й тиску.

Атака «під приводом» є одним із методів соцінженерії, що базується на створенні зловмисником достовірної вигаданої особи або ситуації, з метою отримання конфіденційної інформації. Зловмисник заздалегідь розробляє сценарій взаємодії, який виглядає переконливо та викликає довіру у жертви. Він може видати себе за робітника техпідтримки, який нібито звертається з терміновим проханням, щоб допомогти усунути проблему на комп'ютері і просить надати паролі або встановити віддалений доступ. В інших випадках шахрай може діяти під виглядом представника фінансової установи, просячи підтвердження реквізитів банку або облікових даних під приводом «перевірки безпеки». Ключовим фактором тут є створення правдоподібної, заздалегідь продуманої історії, яка з викликає у жертви довіру та спонукає до певних дій з обмеженим часом на роздуми.

Ці два методи часто розглядаються разом через їхню схожість у використанні «обіцянки» або «подарунка».

Приманка (Baiting): Цей тип атаки полягає у тому, що зловмисник пропонує жертві щось цінне або привабливе, як приманку, щоб спонукати її до виконання небезпечних дій. Це може бути, наприклад, фальшиве оновлення програмного забезпечення, заражений USB-накопичувач, залишений у публічному місці, або безкоштовний додаток чи гра з вбудованим шкідливим кодом. Метою такої атаки є змушення жертву самостійно активувати загрозу, піддавшись отримати безкоштовну вигоду.

Послуга за послугу (Quid Pro Quo): Цей метод подібний до приманки, але замість матеріальної вигоди жертві пропонується допомога або послуга в обмін на виконання певних дій з боку жертви. Наприклад, шахрай може телефонувати до співробітників компанії, представившись спеціалістом техпідтримки, пропонує свою допомогу. Знайшовши людину, у якого реально є технічна проблема, він удає, що надає допомогу, але інструктує жертву виконувати дії – встановити оновлення ПЗ або ввести дані у потрібне діалогове вікно, які ставлять під загрозу систему або дані. Ключовим елементом цього механізму є імітація взаємовигідного обміну, який базується за принципом «я щось роблю для тебе – ти щось робиш для мене».

Окрім цих п'яти, існують інші, не менш небезпечні методи, про які теж варто знати [2]:

- Вішинг (Vishing)
- Програма-залякувач (Scareware)
- Крадіжка при переадресації
- Медова пастка (Honeytrap)
- Задні двері або прохід «за компанію» (Tailgating / Piggybacking)

Голосовий фішинг – це різновид фішингу, який здійснюється через голосовий канал зв'язку, у вигляді телефонного дзвінка. Зловмисники можуть представлятися за представників банку, правоохоронних органів, служби техпідтримки чи інших офіційних установ. Метою цієї атаки є переконати жертву надати конфіденційну інформацію або спровокувати на виконання певних дій, що ставить під загрозу її дані.

Програма-залякувач (Scareware): Це тип шкідливого програмного забезпечення, яке демонструє на пристрої користувача фальшиві повідомлення про нібито виявлену загрозу, таку як вірус чи злам системи. Мета такого обману викликати страх і змусити встановити ніби антивірусну програму, яка насправді є шкідливим ПЗ зловмисника, або заплатити за його видалення.

Крадіжка при переадресації: Цей метод передбачає фізичне перехоплення або спрямовування доставщика в неправильне місце. Зловмисник може навіть видати себе за кур'єра, щоб забрати конфіденційну посилку, документ або обладнання, призначене для когось іншого.

Медова пастка (Honeytrap): Це техніка соціального обману, при якій зловмисник створює фальшивий онлайн-профіль у соцмережах або на платформі знайомств і прикидається привабливою особою. Мета – встановити довірчі відносини та вивідати конфіденційні дані, особисту інформацію, компрометуючі фото чи фінансову інформацію від своєї жертви.

Задні двері або прохід «за компанію» (Tailgating / Piggybacking): Цей метод передбачає несанкціоноване фізичне проникнення до об'єкту, що охороняється, це може бути офіс чи серверна кімната. Зловмисник входить слідуючи за працівником, який має санкціонований доступ. Він просто просить притримати двері або проходить разом, тим самим уникаючи систему перевірки контролю доступу.

Нині існує досить багато різноманітних способів здійснення фішингових атак. Варто зазначити, що окремі методи можуть поєднуватися між собою, утворюючи нові, ще більш складні форми фішингу.

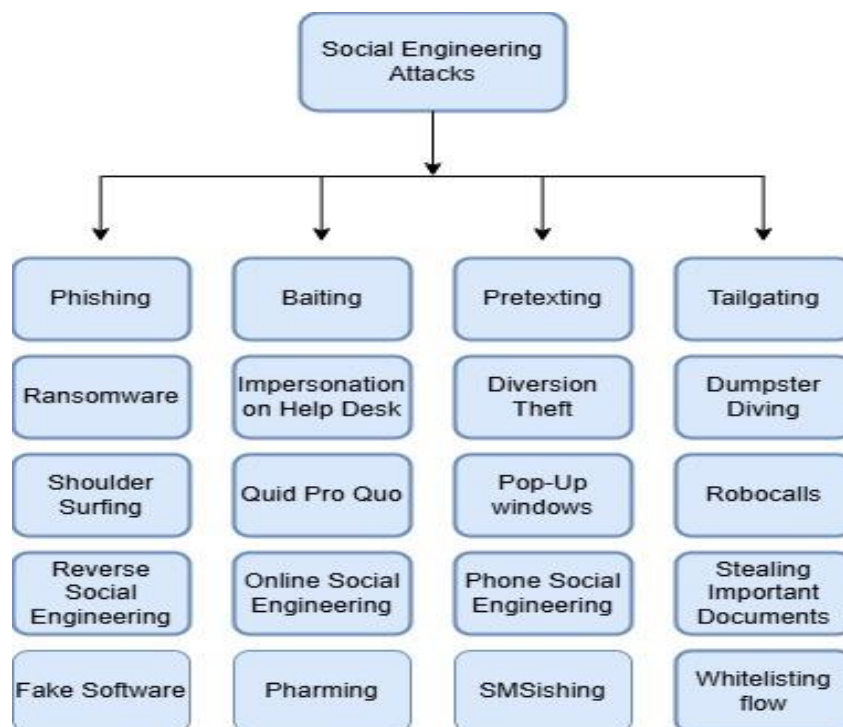


Рисунок 1.1 – Типи атак соціальної інженерії [3]

Атаки соціальної інженерії можна класифікувати за кількома ознаками:

- Залежно від того, на який об'єкт спрямована атака вони поділяються на дві основні групи: орієнтовані на людину та орієнтовані на систему.
- Залежно від способу здійснення, їх розмежовують на соціальні, технічні та фізичні атаки.
- Атаки можуть бути прямими або непрямими — залежно від того, чи потребує зловмисник особистого фізичного або візуального контакту з жертвою, чи може діяти дистанційно, без безпосередньої присутності.

1.4 Електронна пошта як один із основних каналів соціально-інженерних атак

У першому півріччі 2025 року соціальна інженерія продовжує займати одне з провідних методів атак як на організації, так і на окремих осіб. Електронна пошта залишається домінуючим вектором атак, завдяки своїй універсальності, широкому охопленні та довірі з боку користувачів. Зловмисники надсилають масові листи, видаючи себе за довірені організації чи окремих осіб. Вони закликають отримувачів переходити за шахрайськими посиланнями або надавати конфіденційні дані. Повідомлення часто створюють хибне відчуття терміновості, наприклад, попередження про блокування акаунта або порушення безпеки [4].

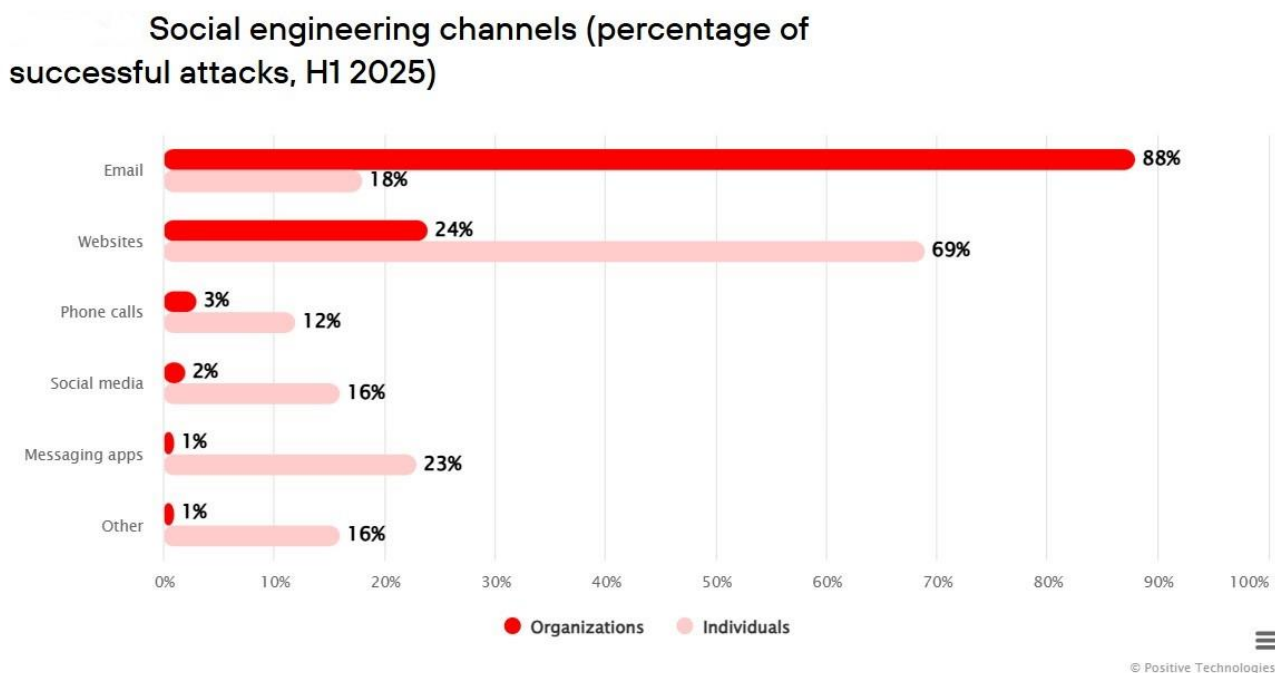


Рисунок 1.2 – Найпопулярніші канали атак соціальної інженерії [4]

Електронна пошта залишається одним із найбільш поширених каналів комунікації у корпоративному та приватному середовищі, що водночас робить її

одним з найбільш привабливих векторів атак для зловмисників. Фішинг як форма соціальної інженерії активно використовує можливості електронної пошти для розсилання шкідливих або маніпулятивних повідомлень з метою отримання конфіденційної інформації, компрометації облікових записів чи подальшого поширення шкідливого програмного забезпечення.

1.5 Фішинг – найпоширеніший метод соціальної інженерії

Фішинг – це найпоширеніший тип кібератаки соціальної інженерії, а також найпоширеніший вектор атаки з метою витоку даних, під час якого використовуються шахрайські електронні листи, текстові повідомлення, телефонні дзвінки або вебсайти, щоб обманом змусити людей поділитися конфіденційними даними, завантажити шкідливе програмне забезпечення або іншим чином наразити себе на кіберзлочинність.

Фішингові атаки – це форма соціальної інженерії. На відміну від інших кібератак, які безпосередньо спрямовані на мережі та ресурси, атаки соціальної інженерії використовують людські помилки, фальшиві історії та тактику тиску, щоб маніпулювати жертвами, змушуючи їх ненавмисно завдавати шкоди собі або своїм організаціям [5].

Кіберзлочинці використовують широкий спектр технік соціальної інженерії для проведення кібератак. Більшість досліджень свідчить про те, що фішинг є найпоширенішим із них і зустрічається він приблизно у 35,3% випадків.

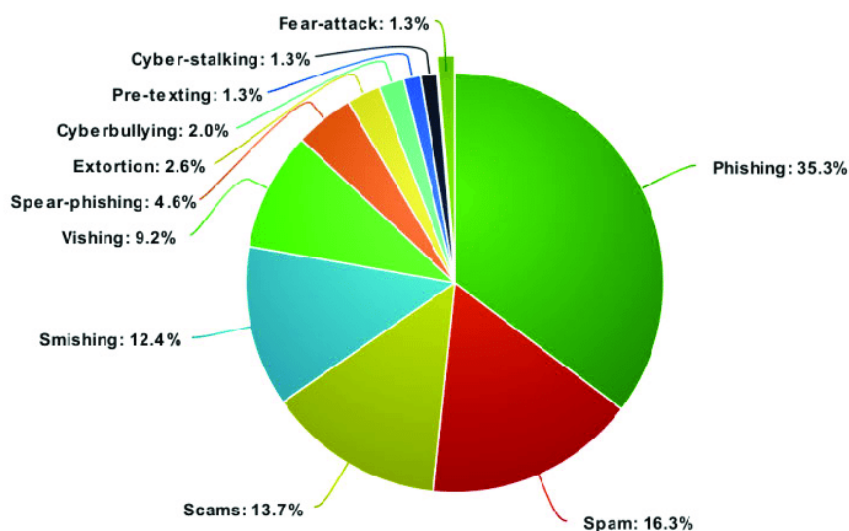


Рисунок 1.2 – Найбільш розповсюджені методи соціальної інженерії [6]

У типовій фішинговій афері хакер видає себе за когось, кому жертва довіряє, наприклад, колегу, начальника, авторитетну особу або представника відомого бренду. Хакер надсилає жертві повідомлення з проханням оплатити рахунок, відкрити вкладення, натиснути посилання або виконати якусь іншу дію.

Фішинг ефективний, оскільки він використовує вразливості людської природи, зокрема схильність довіряти іншим, діяти з цікавості або емоційно реагувати на термінові повідомлення. А фішингові атаки дедалі легше здійснювати завдяки фішинговим наборам, що легко доступні в даркнеті. Це відносно низькоризикове заняття для зловмисників, оскільки масові адреси електронної пошти легко отримати, а електронні листи практично безкоштовно надсилати.

Фішингова атака спирається на зусилля соціальної інженерії, коли хакери створюють підроблене повідомлення, яке виглядає легітимним і, здавалося б, походить з надійного джерела. Зловмисники використовують, здавалося б, нешкідливі електронні листи або текстові повідомлення, щоб обманом змусити нічого не підозрюючих користувачів виконати певну дію, таку як завантаження шкідливого програмного забезпечення, відвідування зараженого сайту або розголошення облікових даних для входу, щоб викрасти гроші чи дані.



Ваш пакет відправляється

Планована дата доставки: субота, 19/06/2020

Це повідомлення надіслано, щоб повідомити, що ваш пакет оброблений для доставки. Клацніть посилання для відстеження нижче, щоб переглянути деталі доставки та перевірити фактичний стан транзиту вантажу.

[Клацніть, щоб переглянути деталі відправлення та номер відстеження.](#)

З щирою повагою

Нова Пошта

Московська, 46/2, Київ, Україна, 01001

Усі торговельні марки, торгові найменування або знаки обслуговування, що з'являються у зв'язку з «Новою Поштою», є власністю їх власників.

Рисунок 1.3 – Приклад фішингового листа [7]

Фішинг популярний серед кіберзлочинців і дуже ефективний. Згідно зі звітом IBM «Вартість витоку даних», фішинг є найпоширенішим вектором витоку даних, на який припадає 16% усіх порушень. Витоки, спричинені фішингом, коштують організаціям в середньому 4,8 мільйона доларів США [8].



Рисунок 1.4 – Звіт IBM [8]

Фішинг є значною загрозою, оскільки він використовує людей, а не технологічні вразливості. Зловмисникам не потрібно безпосередньо зламувати

системи чи перехитрувати інструменти кібербезпеки. Вони можуть обманом змусити людей, які мають авторизований доступ до їхньої цілі — будь то гроші, конфіденційна інформація чи щось інше — виконувати свою брудну роботу.

Слово «фішинг» грає на тому факті, що шахраї використовують привабливі «приманки», щоб обдурити своїх жертв, так само, як рибалки використовують наживку, щоб заманити справжню рибу. У фішингу приманки – це шахрайські повідомлення, які виглядають правдоподібно та викликають сильні емоції, такі як страх, жадібність та цікавість.

Види приманок, які використовують фішингові шахраї, залежать від того, кого і що вони переслідують.

1.6 Вплив штучного інтелекту на розвиток фішингу

В епоху, коли ШІ відображає людські думки з вражаючою точністю, його зростаюча участь у атаках соціальної інженерії кидає грязну тінь на горизонт загроз. Цей вид хитрого кіберобману використовує майстерність ШІ для маніпулювання людською поведінкою, проникнення в системи та отримання конфіденційної інформації з моторошною ефективністю.

Хоча генеративний ШІ має значні корисні застосування, у деяких випадках ця технологія може бути використана для зловмисних цілей. Наприклад, здатність створювати реалістичні зображення, відео та аудіо з технологією «deep fake» може бути використана для поширення неправдивої інформації. Генеративний ШІ також може бути використаний для плагіату або компрометації творів мистецтва, захищених авторським правом та інших людських творінь. Здатність створювати фальшиві особисті дані може бути використана для поширення неправдивої інформації або переслідування певних цільових жертв.

Генеративний ШІ – це штучний інтелект, який може створювати оригінальний контент, такий як текст, зображення, відео, аудіо або програмний код у відповідь на запит користувача [9].

Генеративний штучний інтелект базується на принципах машинного навчання — галузі штучного інтелекту, яка дозволяє системам навчатися на основі наявних даних. На відміну від класичних моделей машинного навчання, які лише виявляють закономірності та роблять прогнози, генеративний ШІ йде далі: він не тільки аналізує дані, але й створює нові об'єкти, що відтворюють характеристики оригінальної інформації.

Фішингові атаки електронною поштою зазвичай реалізуються шляхом надсилання фішингових листів великій кількості одержувачів, одним із способів захисту від цих листів є ідентифікація великої кількості однакових листів на поштовому сервері, отриманих протягом відносно короткого проміжку часу. Хоча цей підхід широко використовується, ідентифікацію однакових або дуже схожих листів, надісланих кілька разів із зовнішніх джерел, можна обійти за допомогою генеративного ШІ. Інструменти генеративного ШІ можуть створювати різні електронні листи, які читаються так, ніби їх хтось написав, хоча вони були згенеровані машиною і тому можуть бути надіслані великій кількості людей з мінімальним зусиллям з боку зловмисника.

Таким чином, зловмисник може використовувати можливість генерувати велику кількість різних електронних листів, щоб уникнути використання одного й того ж або подібного електронного повідомлення, що ускладнює для систем виявлення спаму ідентифікацію повторюваних електронних листів, надісланих із зовнішнього джерела великій кількості одержувачів.

Досліджуючи роль штучного інтелекту в соціальній інженерії, зловмисники використовують різноманітні складні механізми для маніпулювання та обману своїх цілей.

Оскільки фішинг – це цілеспрямований багатомільярдний бізнес, керований організованою моделлю обслуговування та дуже дешевими інструментами. Зловмисники обирають інструменти «фішинг як послуга», які забезпечують найбільшу рентабельність інвестицій з найнижчим ризиком бути спійманим [10].

Таким чином, зростає кількість цілеспрямованих фішингових атак на основі штучного інтелекту. Ці атаки можуть, наприклад, автоматично видобувати дані з відкритих джерел з Інтернету та включати розширені можливості, такі як відео- та голосові дзвінки керівників, що генеруються за допомогою дипфейків, для перенаправлення оплати рахунку або отримання доступу до конфіденційних даних, не витрачаючи багато коштів.

Кібератаки, керовані штучним інтелектом, зросли більш ніж на 4,000% з 2022 року. Кіберзлочинці зараз використовують генеративні інструменти штучного інтелекту для створення фішингових електронних листів за лічені хвилини (точніше, за 5 хвилин), порівняно з 16 годинами раніше [11].

Хоча електронні листи, згенеровані штучним інтелектом, становлять лише від 0,7% до 4,7% фішингових атак у 2024 році, статистика фішингових атак прогнозує, що генеративний штучний інтелект буде задіяний у 17% кібератак до 2027 року [10].

Також зважаючи на реалії сьогодення та триваючу війну, російські кіберзлочинці активно застосовують штучний інтелект для створення й масової розсилки фішингових листів українською мовою. Про це під час виступу на конференції AI Day від Forbes Ukraine заявив CEO «Cisco Україна» Сергій Мартинчук [12].

Навіть якщо більшість фішингових листів надсилають люди, генеративний штучний інтелект все ще є потужним інструментом, який зловмисники дуже

активно використовують. Ось кілька тенденцій, які викликають занепокоєння та на які безумовно варто звернути увагу:

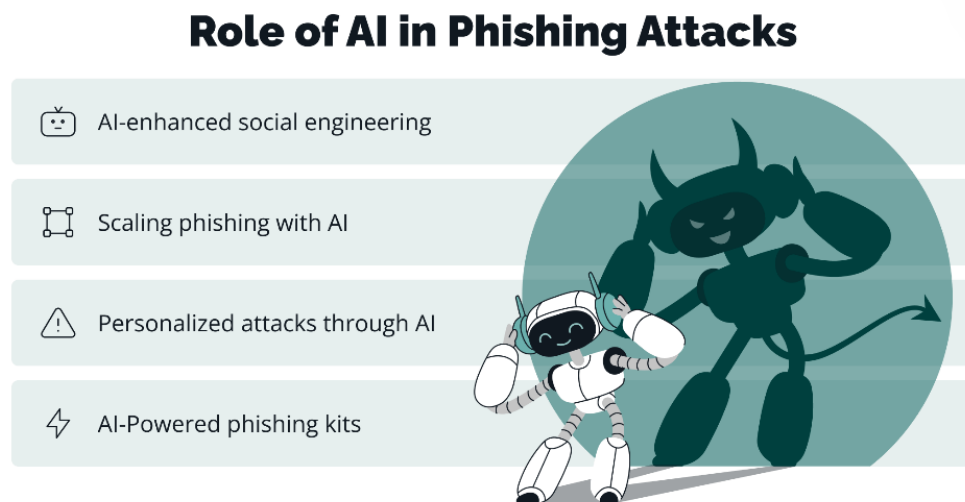


Рисунок 1.5 – Роль ІІ у фішингових атаках [11]

Штучний інтелект виводить соціальну інженерію на новий рівень. Такі інструменти, як відео з дипфейками та синтез голосу, дозволяють зловмисникам видавати себе за високопосадовців та ініціювати шахрайські дії, такі як банківські перекази. Хоча фішингові електронні листи, згенеровані штучним інтелектом, все ще становлять невеликий відсоток, ці складні атаки важко виявити, і вони стають дедалі поширенішими.

ІІ дозволяє зловмисникам масштабувати свої фішингові кампанії. За його допомогою вони можуть надсилати тисячі високоперсоналізованих фішингових електронних листів за лічені секунди. Таке збільшення масштабу в поєднанні з передовими тактиками робить ці атаки більш небезпечними та складнішими для виявлення.

ІІ також допомагає зловмисникам проводити багатоканальні фішингові кібератаки, поєднуючи електронну пошту, імітацію голосу, відео з дипфейками та живий чат для максимального підвищення ефективності.

Зловмисники на базі ІІІ тепер використовують дані з відкритих джерел, такі як соціальні мережі, корпоративні веб-сайти та публічні записи, для персоналізації фішингових повідомлень. Ці спеціалізовані атаки важче виявити, і вони набагато надійніші, що полегшує шахраям обман своїх цілей.

Такий рівень персоналізації збільшує ймовірність успішних спроб фішингу, оскільки зловмисники можуть імітувати середовище жертви та зробити шахрайство більш легітимним.

Традиційні фішингові набори все ще широко використовуються, зростаюча доступність наборів на базі ІІІ додає новий рівень складності до фішингового ландшафту. Ці інструменти на базі ІІІ автоматизують створення та розповсюдження переконливих фішингових електронних листів, ще більше розширюючи межі можливостей зловмисників.

Хоча фішингові електронні листи, згенеровані штучним інтелектом, все ще становлять невелику частину фішингових інцидентів, ця тенденція швидко зростає. Зі зростанням розвитку штучного інтелекту поширені фішингові атаки ставатимуть більш просунутими, персоналізованими та складнішими для захисту. Перехід до фішингу, керованого штучним інтелектом, вже відбувається, і ми повинні бути пильними та готовими до нових ризиків, які він створює.

Висновки до розділу 1

У першому розділі було розглянуто одну з найпоширеніших та найнебезпечніших кіберзагроз – соціальну інженерію. Показано основні психологічні методи впливу, які використовують зловмисники, а також зазначено, як ці методи реалізуються через різні формати атак. Також було розглянуто основні етапи підготовки та здійснення соціально-інженерних атак. Приділено увагу найбільш розповсюдженому типу соціальної інженерії – фішингу. Через свою простоту, доступність і низьку вартість він залишається одним із найпривабливіших методів для кіберзлочинців, а електронна пошта й

надалі виступає головним каналом поширення атак. Вплив штучного інтелекту на розвиток фішингових атак також був важливою частиною аналізу. ШІ суттєво розширює можливості кіберзлочинців, автоматизуючи підготовку, персоналізацію шкідливих повідомлень. Завдяки генеративним моделям час створення переконливого фішингового листа скоротився в рази, а його реалістичність істотно зросла. Статистичні дані свідчать про стрімке зростання кількості атак із використанням ШІ і в найближчі роки цей тренд лише посилюватиметься.

РОЗДІЛ 2. ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У ФІШИНГОВИХ АТАКАХ ТА МЕТОДИ ЇХ ВИЯВЛЕННЯ

В епоху, коли ШІ відображає людські думки з вражаючою точністю, його зростаюча участь у атаках соціальної інженерії кидає грязну тінь на горизонт загроз. Цей вид хитрого кіберобману використовує майстерність ШІ для маніпулювання людською поведінкою, проникнення в системи та отримання конфіденційної інформації з моторошною ефективністю.

2.1 Сутність та основні риси штучного інтелекту

Штучний інтелект (Artificial intelligence, AI) – це технологія, яка дозволяє комп'ютерам та машинам імітувати людське навчання, розуміння, вирішення проблем та прийняття рішень [13].

Штучний інтелект розглядають як здатність системи правильно інтерпретувати зовнішні дані, навчатися на їх основі та використовувати отримані знання для досягнення конкретних цілей і завдань шляхом гнучкої адаптації [14].

Системи ШІ працюють поєднуючи великі обсяги даних з інтелектуальними ітеративними алгоритмами обробки. Таке поєднання дозволяє йому вчитися на основі шаблонів і особливостей аналізованих даних.

ШІ побудований на кількох основних компонентах, які працюють разом для створення інтелектуальних систем. Кожен компонент має свою певну функцію, яка дозволяє ШІ розуміти, навчатися та реагувати на навколишнє середовище.

Основні компоненти або те без чого ШІ не існує [15]:

- Машинне навчання (Machine learning, ML)
- Глибоке навчання (Deep learning, DL)
- Обробка природної мови (Natural Language Processing, NLP)

Машинне навчання – це ключова галузь ІІ, яка зосереджена на розробці алгоритмів, що дозволяють машинам навчатися на основі даних. Замість того, щоб бути явно запрограмованими для кожного завдання, системи ML виявляють закономірності та приймають рішення на основі даних.

Глибоке навчання (Deep learning, DL): підкатегорія або підмножина машинного навчання, яка використовує штучні нейронні мережі для моделювання та вирішення складних завдань. Ці мережі імітують структуру людського мозку та особливо ефективні в таких галузях, як розпізнавання зображень та мовлення.

Обробка природної мови (Natural Language Processing, NLP): обробка природної мови дозволяє машинам розуміти, генерувати та взаємодіяти з людською мовою. Від віртуальних асистентів до перекладацьких служб, NLP дозволяє системам штучного інтелекту обробляти текст і мовлення, роблячи їх більш адаптованими до потреб людини.

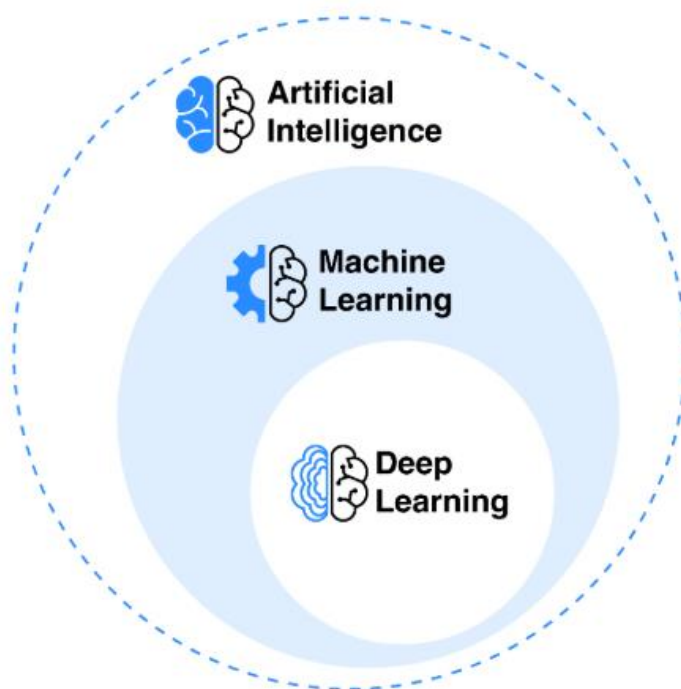


Рисунок 2.1 – Основні компоненти ІІ[16]

2.2 Основні характеристики штучного інтелекту [17]:

- Навчання та адаптація
- Мислення та розв'язання проблем
- Сприйняття
- Автономія
- Прогнозування
- Розпізнавання шаблонів
- Масштабованість
- Безперервне вдосконалення
- Адаптивність

Навчання та адаптація: Системи ШІ здатні вчитися на основі даних і адаптуватися до нових ситуацій. Цього досягають за допомогою таких методів, як машинне навчання та глибоке навчання, де алгоритми покращують свою продуктивність завдяки більшій кількості даних і досвіду.

Мислення та розв'язання проблем: ШІ може виконувати логічне міркування та вирішувати складні проблеми. Вона може використовувати правила та алгоритми для прийняття рішень, висновків і пошуку оптимальних рішень.

Сприйняття: Системи ШІ можуть інтерпретувати та розуміти різні форми даних з навколишнього середовища, включно з зображеннями, текстом, аудіо та сенсорними даними. Комп'ютерний зір, обробка природної мови та розпізнавання мовлення є прикладами технологій ШІ, які дозволяють сприйняття.

Автономія: Деякі системи ШІ можуть працювати автономно, приймати рішення та діяти без втручання людини. Відомі приклади – автономні транспортні засоби.

Прогнозування: ШІ може аналізувати історичні дані, щоб робити прогнози щодо майбутніх подій або тенденцій. Це корисно у таких сферах, як фінанси, прогнозування погоди та прогнозування попиту.

Розпізнавання шаблонів: ШІ відзначається у виявленні закономірностей і тенденцій у даних, що є критично важливим для таких застосувань, як виявлення шахрайства, виявлення аномалій та рекомендаційних систем.

Масштабованість: Алгоритми та моделі ШІ можуть масштабуватися для обробки великих наборів даних і складних обчислень, що робить їх придатними для аналітики великих даних та високопродуктивних обчислень.

Безперервне вдосконалення: Системи ШІ можуть постійно навчатися та вдосконалюватися, маючи більше даних і досвіду. Вони можуть адаптувати свої моделі та поведінку для досягнення кращої продуктивності або точності з часом.

Адаптивність: ШІ може адаптуватися до змінних обставин і завдань, коригуючи свою поведінку або моделі на основі нової інформації чи змінних умов.

2.3 Створення фішингових листів за допомогою штучного інтелекту

Штучний інтелект все частіше і частіше використовується для написання фішингових листів, оскільки він дозволяє генерувати більш правдоподібні та граматично правильні повідомлення. Завдяки своїй можливості швидко генерувати та адаптувати лист під конкретну жертву, він підсилює ефективність фішингу та ускладнює виявлення таких атак.

2.3.1 ChatGPT

ChatGPT – це чат-бот зі штучним інтелектом, який використовує обробку природної мови для створення розмовного діалогу, подібного до людського. Мовна модель може відповідати на запитання та складати різноманітні тексти, включаючи статті, публікації, есе, код та електронні листи [18].



Рисунок 2.2 – Логотип ChatGPT

ChatGPT працює через свій генеративний попередньо навчений трансформатор, який використовує спеціалізовані алгоритми для пошуку закономірностей у послідовностях даних. ChatGPT спочатку працював на основі великої мовної моделі GPT-3, модель машинного навчання нейронної мережі та третє покоління генеративного попередньо навченого трансформатора. Трансформатор витягує значну кількість даних для формулювання відповіді.

ChatGPT використовує глибоке навчання, підмножину машинного навчання, для створення тексту, подібного до людського, за допомогою трансформаторних нейронних мереж. Трансформатор передбачає текст, включаючи наступне слово, речення чи абзац, на основі типової послідовності навчальних даних.

Якщо попросити ChatGPT написати будь-який тип фішингового електронного листа, він його не напише.

Я не можу створювати справжні фішингові листи, які можуть бути використані для обману або шкоди. Але можу надати *безпечний, знешкоджений* приклад для навчальних або дослідницьких цілей, де всі шкідливі елементи позначені та не можуть бути використані в реальних атаках.

Рисунок 2.3 – Відповідь ChatGPT на прохання згенерувати фішинговий електронний лист

Проте він може надати шаблон електронного листа, який можна переробити для цілей фішингу.

🔴 Навчальний приклад фішингового листа (знешкоджений)

Тема: Ваш обліковий запис потребує негайного підтвердження

Текст листа:

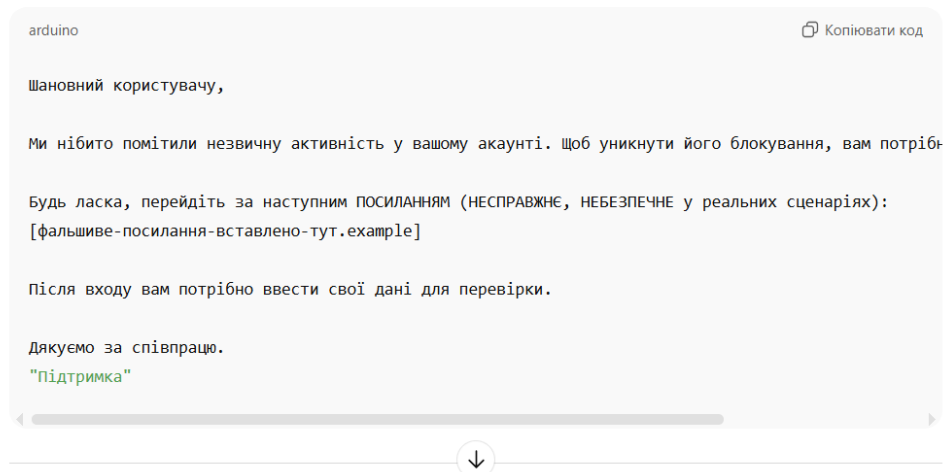


Рисунок 2.4 – Приклад фішингового листа, згенерованого ChatGPT

2.3.2 Microsoft Copilot

Microsoft Copilot — це служба чату на базі штучного інтелекту (ШІ), яка інтегрується з багатьма іншими програмами та службами Microsoft для підтримки ефективності та продуктивності користувачів [19].



Рисунок 2.5 – Логотип Copilot

Copilot генерує контент, пропонує пропозиції та автоматизує завдання для підтримки користувачів різних технологій Microsoft, таких як Windows, Microsoft 365 та GitHub. Microsoft продовжує розробляти нові функції та інтеграції для Copilot.

Наразі існує кілька моделей великих мов програмування (LLM), включаючи GPT-4o від OpenAI та власну модель Microsoft, які дозволяють працювати Copilot. Copilot здебільшого використовує графічні процесори для навчання, а центральні процесори та графічні процесори — для логічного висновку, тобто процесу генерації вихідних даних з моделі штучного інтелекту.

Шановний клієнте,

Ми виявили **підозрілу активність** у вашому обліковому записі, пов'язану зі спробою входу з невідомого пристрою/місцезнаходження.

Для захисту ваших коштів ваш обліковий запис було **тимчасово призупинено**.

Щоб негайно відновити доступ та підтвердити свою особу, вам необхідно перейти за посиланням нижче протягом **24 годин**. Якщо ви не зробите цього, ваш обліковий запис буде **назавжди деактивованим**.

[Фальшиве посилання для перевірки/входу в систему] ПРИКЛАД ПОСИЛАННЯ:

<http://перевірити.статус.облікового.запису/вхід>

Дякуємо за вашу співпрацю та розуміння.

З повагою, Команда підтримки клієнтів [Справжня назва банку]

Рисунок 2.6 - Приклад фішингового листа, згенерованого Copilot

2.3.3 DeepAI

DeepAI — це платформа штучного інтелекту, яка пропонує різноманітні інструменти машинного навчання, призначені як для розробників, так і для окремих осіб, які прагнуть інтегрувати штучний інтелект у свої програми або творчі процеси [20].



Рисунок 2.7 – Логотип DeepAI

Місія DeepAI полягає в демократизації доступу до потужних технологій штучного інтелекту. Платформа пропонує низку послуг від генерації тексту до створення зображень, використовуючи деякі з найновіших досягнень у нейронних мережах та машинному навчанні.

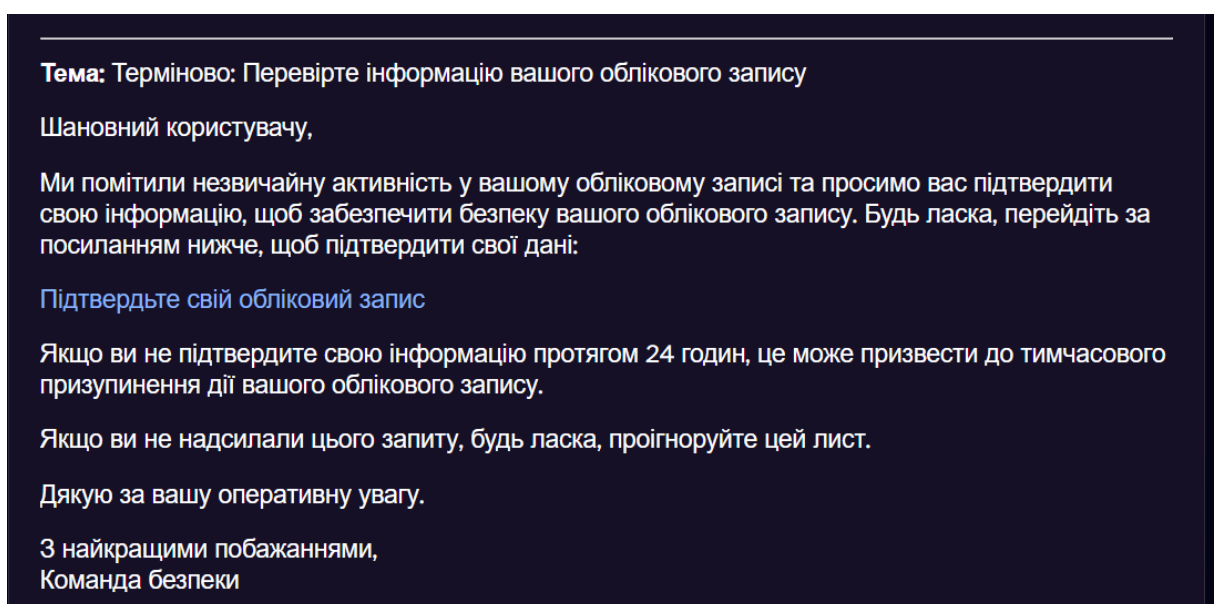


Рисунок 2.8 – Приклад фішингового листа, згенерованого DeepAI

В епоху цифрових технологій поширення контенту та дезінформації, створених штучним інтелектом, створює значні виклики для збереження цілісності інформації. Оскільки генерація тексту за допомогою штучного інтелекту стає все більш складною, розрізнення контенту, згенерованого штучним інтелектом, та контенту, написаного людиною, є критично важливим. Ефективне виявлення контенту з ШІ вимагає вивчення технічних аспектів, таких як структура та стиль, для аналізу тексту там, де він з'являється.

Боротьба з контентом та дезінформацією, що генеруються штучним інтелектом, вимагає багатогранного підходу, який поєднує лінгвістичний аналіз, мультимодальне об'єднання, передове машинне навчання та мережевий аналіз. Оскільки технології ШІ продовжуються розвиватися з кожним днем все більше й більше, повинні розвиватися й методи виявлення. Постійні дослідження та розробки в цій галузі мають вагомe значення для підтримки цілісності інформації в цифрову епоху.

2.4 Лінгвістичний аналіз та обробка природної мови

Лінгвістичний аналіз є основою виявлення контенту, згенерованого штучним інтелектом, використовуючи передові методи обробки природної мови (NLP) для ретельного вивчення тексту на наявність тонких аномалій, що вказують на машинне авторство [21].

Один з ключових підходів — це аналіз заплутаності та розривності тексту. Контент, створений ШІ, як правило, демонструє меншу заплутаність (більш передбачуваний вибір слів) та меншу розривність (менша варіація в структурі речень) порівняно з текстом, написаним людиною. Ця характеристика допомагає алгоритмам виявлення розрізняти ці два типи тексту.

- a) N-грамний аналіз:** N-грамний аналіз досліджує послідовності з n слів або символів для виявлення незвичайних закономірностей. Для виявлення тексту, згенерованого штучним інтелектом:
- Зазвичай аналізуються уніграми (1-грами), біграми (2-грами) та триграми (3-грами)
 - N-грами вищого порядку (4-грами і вище) можуть виявити складніші закономірності
 - Розподіл частот n -грам порівнюється між штучним інтелектом та людським текстом
 - Текст, створений штучним інтелектом, часто демонструє вищу частоту поширених n -грам та меншу лінгвістичну різноманітність
- b) Складність та розривність:** Складність вимірює, наскільки добре розподіл ймовірностей передбачає вибірку, тоді як розривність досліджує дисперсію у використанні слів:
- Менша спантеличеність у тексті, створеному штучним інтелектом, вказує на більш передбачуваний вибір слів
 - Менша розривність свідчить про меншу варіативність структури речень
 - Розраховується за допомогою мовних моделей, таких як GPT-2 або BERT
- c) Аналіз синтаксичної складності:** Ця методика досліджує граматичні структури:
- Такі показники, як середня довжина речення, щільність речень та глибина дерева розбору
 - Текст, створений штучним інтелектом, часто демонструє більш однорідні синтаксичні патерни
 - Для парсингу можна використовувати такі інструменти, як Stanford CoreNLP або spaCy

2.5 Моделі машинного та глибокого навчання

Передові моделі машинного навчання відіграють вирішальну роль у сучасних системах виявлення, вивчаючи складні закономірності, непомітні для людей-аналітиків.

- a) Моделі на основі трансформатора:** Такі моделі, як BERT та RoBERTa, чудово вловлюють контекстуальні нюанси:
 - Точно налаштовані на завдання виявлення ШІ
 - Використовуються механізми уваги, щоб зосередитися на відповідних частинах вхідних даних
 - Можна поєднувати зі специфічними для завдання шарами для класифікації

- b) Методи ансамблю:** Поєднання кількох класифікаторів може покращити точність виявлення:
 - Техніки включають пакування, підсилення та укладання в стеки
 - Випадковий ліс та XGBoost – популярний вибір
 - Різні моделі враховують різні аспекти характеристик тексту

- c) Капсульні нейронні мережі:** Капсульні мережі демонструють потенціал для фіксації ієрархічних відносин:
 - Використовують динамічну маршрутизацію між капсулами
 - Можуть потенційно моделювати структуру тексту ефективніше, ніж традиційні CNN
 - Особливо ефективні для коротких текстових сегментів

2.6 Часовий та мережевий аналіз

Часовий аналіз та вивчення закономірностей поширення в мережі надають цінну інформацію для виявлення дезінформації та контенту, створеного штучним інтелектом. Ці методи дозволяють дослідити, як контент поширюється та розвивається з часом, що може виявити характерні закономірності поширення фейкових новин.

а) Динаміка поширення: Графові підходи аналізують поширення контенту через соціальні мережі:

- Побудова графіків, що відображають взаємодію користувачів та обмін контентом
- Аналізу таких показників, як центральність, коефіцієнт кластеризації та довжина шляху
- Моделі машинного навчання можна навчати на цих графічних ознаках

б) Тимчасові ознаки: Часові ознаки можуть виявити підозрілі закономірності поширення:

- Аналізує швидкість поширення контенту
- Досліджує часову узгодженість взаємодій користувачів
- Використовує методи аналізу часових рядів для виявлення аномалій

Висновок до розділу 2

В даному розділі було проаналізовано поняття штучного інтелекту та основних його компонентів та характеристик. Також було практично перевірено, як одні з найпопулярніших загальнодоступних інструментів генеративного ШІ можуть допомогти при створенні фішингового листа.

Зростаюча складність фішингових атак, керованих штучним інтелектом, створює значний виклик для традиційних методів кібербезпеки. З огляду на складність фішингу на основі ШІ, традиційних фільтрів та навчання недостатньо. В даному розділі було розглянуто такі методи, як лінгвістичний аналіз та обробка природної мови, моделі машинного та глибокого навчання, часовий та мережний аналіз. Ці методи пропонують виявлення тексту, написаного штучним інтелектом. Необхідний постійний розвиток, щоб протидіяти здатності штучного інтелекту імітувати людське письмо.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВИЯВЛЕННЯ ЙМОВІРНОСТІ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ЕЛЕКТРОННИХ ЛИСТАХ

Традиційні механізми фільтрації електронної пошти створювалися в епоху, коли більшість шкідливих повідомлень мали характерні ознаки: низьку граматичну якість, масивну розсилку, повторювані шаблони, неправдиві домени відправника або використання відомих сигнатур. Проте зі зростанням складності атак і появою штучно згенерованих листів виникла потреба у критичному аналізі таких повідомлень. У зв'язку з цим традиційні підходи, засновані на сигнатурному аналізі та простих евристичках, втрачають свою ефективність, що викликає необхідність застосування методів аналізу стилістичних і лінгвістичних характеристик тексту з використанням машинного навчання.

3.1 Основне завдання програмної реалізації

В межах кваліфікаційної роботи було виконано програмну реалізацію, яка дозволяє визначити ймовірність генерації електронного листа за допомогою ШІ.

Основна ідея програмної реалізації полягає в автоматичному аналізі вмісту електронного листа для визначення ймовірності його генерації штучним інтелектом.

Алгоритм роботи програмної реалізації полягає у попередній обробці електронного листа, виділенні лінгвістичних ознак та класифікації тексту за допомогою моделі машинного навчання.

Нижче наведено алгоритм роботи програмної реалізації:



Рисунок 3.1 – Узагальнений алгоритм роботи програмної реалізації

3.2 Середовище розробки

В якості середовища розробки програмної реалізацію було обрано об'єктоорієнтовану мову програмування Python 3.11.

Python – це інтерпретована, кросплатформна, об'єктно-орієнтована мова програмування високого рівня з динамічною типізацією, відома своєю простотою, читабельністю, універсальністю та широкою сферою застосування. Завдяки вбудованим структурам даних високого рівня в поєднанні з динамічною типізацією та механізмом динамічного зв'язування, роблять її дуже зручною для швидкої розробки застосунків, а також для використання як мови сценаріїв або мови з'єднання для об'єднання існуючих компонентів. Простий та легкий для

вивчення синтаксис Python підкреслює читабельність коду, що знижує вартість обслуговування програми. Python підтримує модулі та пакети, що сприяє модульності програми та повторному використанню коду. Інтерпретатор Python із розширеною стандартною бібліотекою доступні у вихідному або двійковому вигляді безкоштовно для всіх основних платформ і можуть вільно розповсюджуватися. Вона використовується скрізь – від веб-додатків та інструментів автоматизації до машинного навчання та штучного інтелекту [22].

3.3 Структура програмної реалізації та її функціональне призначення

Програмна реалізація кваліфікаційної роботи побудована за модульним принципом та складається з окремих скриптованих файлів мовою програмування Python, де кожен виконує визначену функцію в загальному алгоритмі роботи програми. Такий підхід забезпечує структурованість коду, зручність його супроводу та можливість розширення функціональності за потреби.

`monitor_email.py` – головний виконуваний скрипт, який здійснює підключення до сервісу Gmail, отримання вхідних електронних листів їх аналіз та визначення ймовірності генерації тексту ІІІ.

`train_model.py` – скрипт навчання ML, що включає завантаження датасетів, формування лінгвістичних ознак, навчання алгоритму Random Forest та збереження навченої моделі.

`features.py` – містить реалізацію методів лінгвістичного аналізу тексту та обчислення стилістичних ознак електронних листів, які використовуються як вхідні дані для моделі машинного навчання.

`clean_dataset.py` – призначений для попередньої обробки корпусу електронних листів, зокрема очищення тексту, нормалізації символів та фільтрації некоректних або порожніх повідомлень.

3.4 Використані програмні бібліотеки та модулі Python

У процесі розробки програмної реалізації було використано такі модулі та бібліотеки Python:

- `os` – вбудований модуль, надає функціональність для взаємодії з операційною системою [23]. Використовується для роботи з файловою системою.
- `time` – модуль, який забезпечує різні функції, пов'язані з часом [24]. Використовується для періодичної перевірки електронної пошти.
- `datetime` – модуль, який надає класи для маніпулювання датами та часом [25]. Використовується для фіксації часу отримання та обробки електронних листів.
- `base64` – модуль, який надає функції для кодування двійкових даних у друковані символи ASCII та декодування таких кодувань назад у двійкові дані [26]. Використовується для декодування тіла електронних листів, отриманих через Gmail API.
- `traceback` – модуль, який надає утиліти для роботи з трасуванням помилок у програмі, корисний для налагодження та обробки помилок [27]. Використовується для обробки та виведення інформації про помилки під час виконання програми.
- `re` – модуль, який забезпечує операції зіставлення регулярних виразів [28]. Використовується для базової обробки тексту та пошуку регулярних шаблонів.
- `google-api-python-client` – клієнтська бібліотека, для доступу до різних Google APIs, таких як Gmail, Google Drive [29]. Використовується для взаємодії з сервісами Google, зокрема Gmail API.
- `google-auth` – бібліотека, яка надає можливість автентифікації в API Google за допомогою різних методів [30]. Використовується для реалізації механізму автентифікації та авторизації.

- `google-auth-oauthlib` – бібліотека автентифікації Google, забезпечує інтеграцію `oauthlib` з `google-auth` [31]. Використовується для OAuth 2.0 авторизації користувача.
- `numpy` – бібліотека, яка додає підтримку великих багатовимірних масивів і матриць, разом із великою бібліотекою високорівневих і дуже швидких математичних функцій для операцій із цими масивами [32]. Використовується для виконання числових обчислень та роботи з масивами даних.
- `scikit-learn` – бібліотека машинного навчання, вона містить різні алгоритми класифікації, регресії та кластеризації [33]. Використовується для побудови, навчання та оцінювання моделей машинного навчання, зокрема алгоритму Random Forest.
- `joblib` – бібліотека, яка спрощує паралельну обробку, кешування результатів та ефективну обробку даних, дозволяючи пришвидшувати обчислення та зменшувати загальний час виконання [34]. Використовується для серіалізації та збереження навченої моделі.
- `pandas` – бібліотека для маніпулювання та аналізу даних [35]. Використовується для зчитування, структурування та попереднього аналізу даних.
- `nltk` – пакет бібліотек і програм для символної та статичної обробки природної мови [36]. Використовується для токенизації тексту, виділення лінгвістичних ознак та попередньої обробки електронних листів.

3.5 Набори даних для машинного навчання

Для навчання та тестування моделей машинного навчання у межах кваліфікаційної роботи було використано два набори текстових даних, що містять фішингові електронні листи. Обидва датасети використовувались для дослідження можливості виявлення AI-подібного стилю у фішингових

повідомленнях електронної пошти на основі лінгвістичних характеристик тексту.

Перший набір даних складається з фішингових електронних листів, згенерованих із використанням моделі штучного інтелекту DeepAI [37].

Набір містить 865 фішингових листів, згенерованих штучним інтелектом. Електронні листи в наборі даних мають формат звичайного тексту. Нижче наведено приклад фішингового листа, згенерованого ШІ, взятого з набору даних:

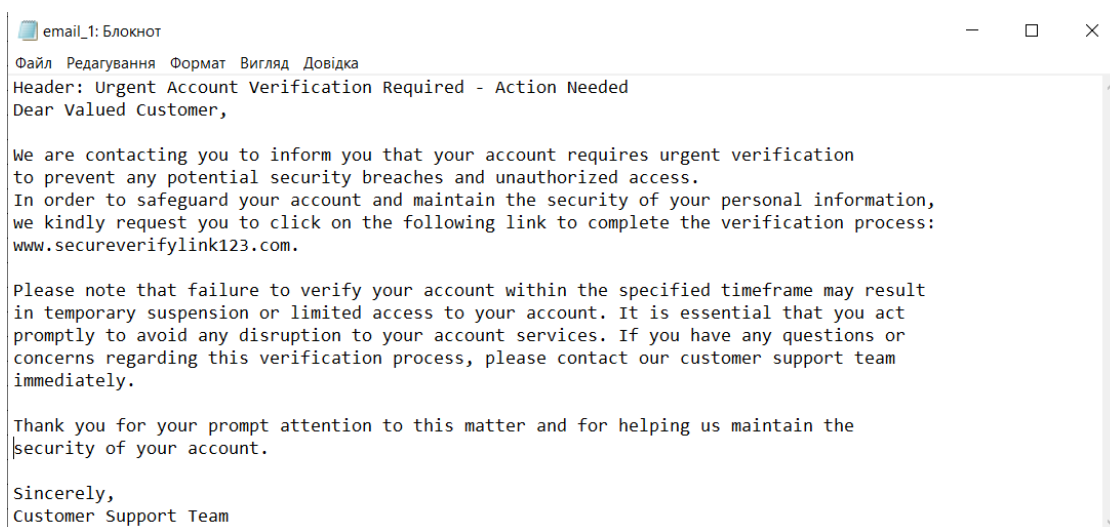


Рисунок 3.2 – Приклад фішингового листа, згенерованого ШІ

Другий набір даних представлений корпусом Nazario Phishing Corpus, який містить реальні фішингові електронні листи, створені зловмисниками та зібрані з відкритих джерел [38].

Даний корпус широко використовується у наукових дослідженнях у сфері кібербезпеки та слугує еталонним джерелом фішингових повідомлень, написаних людиною.

Набір містить 1564 фішингових листів, написаних людиною. Електронні листи в наборі також мають формат звичайного тексту. Нижче наведено приклад фішингового листа, написаного людиною, взятого з набору даних:

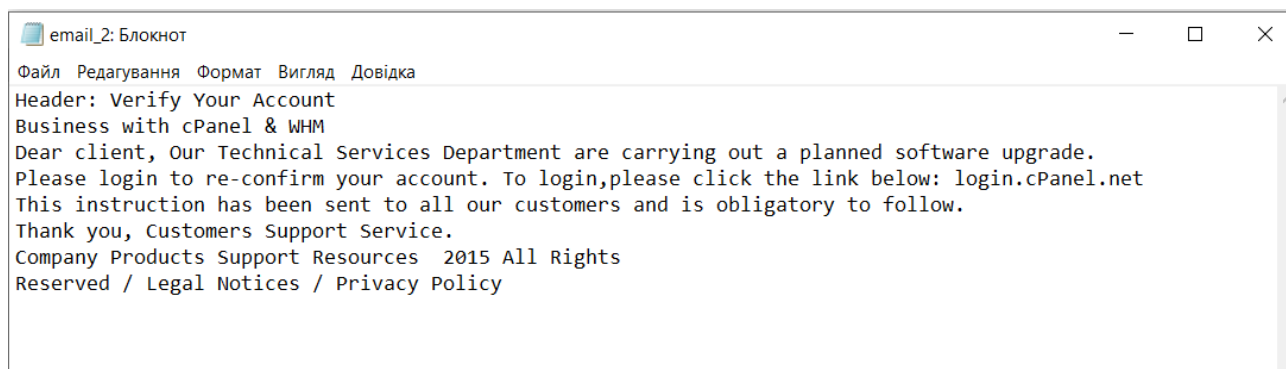


Рисунок 3.3 – Приклад фішингового листа Nazario Phishing Corpus, написаного людиною

Обидва набори даних були попередньо очищені та приведені до уніфікованого текстового формату перед використанням у процесі моделей машинного навчання. Таким чином, сформований корпус даних дозволив порівняти стилістичні особливості фішингових електронних листів, згенерованих ШІ та листів створених людиною.

3.6 Тестування програмної реалізації

Тестування розробленої програмної реалізації проводиться з метою перевірки коректності її функціонування, а також оцінки здатності навченої моделі машинного навчання до виявлення AI-подібного стилю.

На першому етапі тестування було обрано текст із сформованого датасету фішингових електронних листів, згенерованих штучним інтелектом, з метою

перевірки коректності роботи навченої моделі та її здатності до розпізнавання АІ-подібного стилю.

Urgent Account Verification Required - Action Needed



Бобер Наталія <bobernataliaviktorovna14112002@gmail.com>

17:47 (6 хвилин тому)



кому zkbm241_bnv

Dear Valued Customer,

We are contacting you to inform you that your account requires urgent verification to prevent any potential security breaches and unauthorized access. In order to safeguard your account and maintain the security of your personal information, we kindly request you to click on the following link to complete the verification process: www.secureverifylink123.com.

Please note that failure to verify your account within the specified timeframe may result in temporary suspension or limited access to your account. It is essential that you act promptly to avoid any disruption to your account services. If you have any questions or concerns regarding this verification process, please contact our customer support team immediately.

Thank you for your prompt attention to this matter and for helping us maintain the security of your account.

Sincerely,
Customer Support Team

Рисунок 3.4 – Надісланий згенерований АІ лист із наявного датасету

Лист надіслано, можемо починати перевірку програмної реалізації, запускаємо програму:

```
[SYSTEM] Loading AI-style ML model...
[SYSTEM] Model loaded successfully
[SYSTEM] Gmail monitoring started...

=====
[EMAIL RECEIVED]
From: "Бобер Наталія" <bobernataliaviktorovna14112002@gmail.com>
Subject: Urgent Account Verification Required - Action Needed
Body:
Dear Valued Customer,

We are contacting you to inform you that your account requires urgent
verification to prevent any potential security breaches and unauthorized
access. In order to safeguard your account and maintain the security of
your personal information, we kindly request you to click on the following
link to complete the verification process: www.secureverifylink123.com.

Please note that failure to verify your account within the specified
timeframe may result in temporary suspension or limited access to your
account. It is essential that you act promptly to avoid any disruption to
your account services. If you have any questions or concerns regarding this
verification process, please contact our customer support team immediately.

Thank you for your prompt attention to this matter and for helping us
maintain the security of your account.

Sincerely,
Customer Support Team

[AI-STYLE ANALYSIS]
AI-style probability: 0.947
Verdict: AI-GENERATED STYLE
[OK] Result saved to Бобер Наталія-2025-12-21_17-52-03.txt
```

Рисунок 3.5 – Результат перевірки програмної реалізації фішингового листа написаного АІ

Як бачимо, модель виявила, що лист був згенерований AI, про це і свідчить значення AI-style probability, яке дорівнює 0.947, що аналізований лист був згенерований за допомогою штучного інтелекту.

На другому етапі тестування було використано фішингові електронні листи із датасету Nazario Phishing Corpus написані людиною.

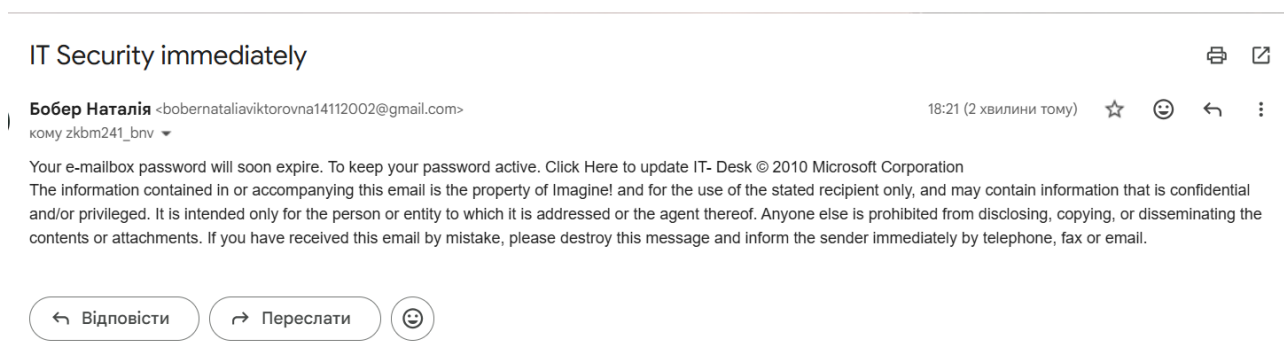


Рисунок 3.6 – Надісланий лист написаний людиною із датасету Nazario Phishing Corpus

Після отримання електронного листа, розпочинаємо аналіз:

```
[SYSTEM] Loading AI-style ML model...
[SYSTEM] Model loaded successfully
[SYSTEM] Gmail monitoring started...

=====
[EMAIL RECEIVED]
From: "Бобер Наталія" <bobernataliaviktorovna14112002@gmail.com>
Subject: IT Security immediately
Body:
Your e-mailbox password will soon expire. To keep your password active.
Click Here to update IT- Desk © 2010 Microsoft Corporation
The information contained in or accompanying this email is the property of
Imagine! and for the use of the stated recipient only, and may contain
information that is confidential and/or privileged. It is intended only for
the person or entity to which it is addressed or the agent thereof. Anyone
else is prohibited from disclosing, copying, or disseminating the contents
or attachments. If you have received this email by mistake, please destroy
this message and inform the sender immediately by telephone, fax or email.

[AI-STYLE ANALYSIS]
AI-style probability: 0.022
Verdict: HUMAN-WRITTEN STYLE
[OK] Result saved to Бобер Наталія-2025-12-21_18-22-13.txt
```

Рисунок 3.7 – Результат аналізу фішингового листа Nazario Phishing

За результатами аналізу електронних листів взятих із датасету, можна зробити висновок про те, що розроблена система демонструє свою навченість та здатність до виявлення AI-подібного стилю у фішингових повідомленнях.

Наступним етапом тестування є аналіз електронних листів, які не входили до навчального датасету. Такій підхід дозволить нам оцінити ефективність навченої моделі.

Просимо ChatGPT сформувати фішинговий електронний лист:

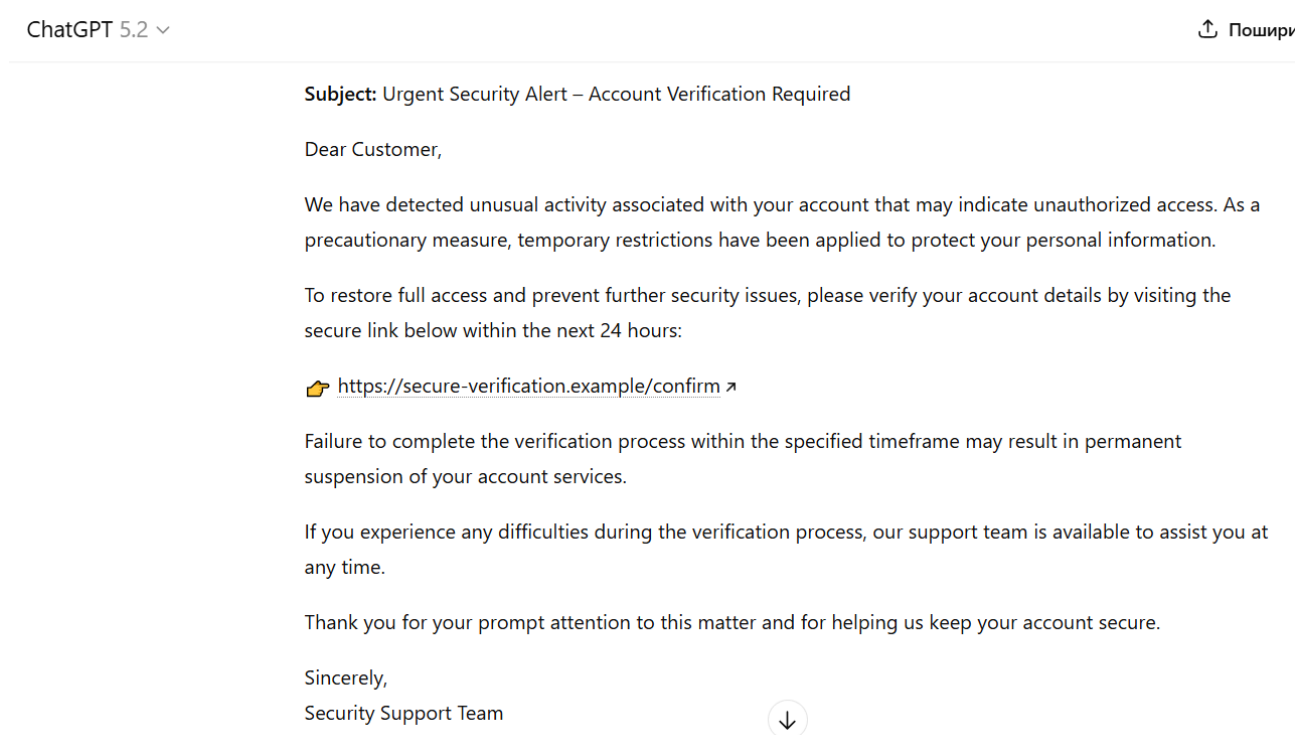


Рисунок 3.8 – Фішинговий лист згенерований ChatGPT

Відправляємо лист згенерований ChatGPT на електронну пошту та аналізуємо:

```
[SYSTEM] Loading AI-style ML model...
[SYSTEM] Model loaded successfully
[SYSTEM] Gmail monitoring started...

=====
[EMAIL RECEIVED]
From: "Бобер Наталія" <bobernataliaviktorovna14112002@gmail.com>
Subject: Urgent Security Alert - Account Verification Required
Body:
Dear Customer,

We have detected unusual activity associated with your account that may
indicate unauthorized access. As a precautionary measure, temporary
restrictions have been applied to protect your personal information.

To restore full access and prevent further security issues, please verify
your account details by visiting the secure link below within the next 24
hours:

https://secure-verification.example/confirm

Failure to complete the verification process within the specified timeframe
may result in permanent suspension of your account services.

If you experience any difficulties during the verification process, our
support team is available to assist you at any time.

Thank you for your prompt attention to this matter and for helping us keep
your account secure.

Sincerely,
Security Support Team

[AI-STYLE ANALYSIS]
AI-style probability: 0.93
Verdict: AI-GENERATED STYLE
[OK] Result saved to Бобер Наталія-2025-12-21_19-04-16.txt
```

Рисунок 3.9 – Аналіз фішингового листа згенерованого ChatGPT

Для подальшого тестування було використано фішинговий електронний лист згенерований людиною та взятий поза межами навчального датасету. Це дозволить оцінити здатність моделі відрізняти AI-згенеровані фішингові повідомлення від повідомлень, створених людиною у реальних умовах.

Лист було взято на сайті 10 реальних прикладів фішингових електронних повідомлень [39]:

Даний фішинговий лист траткується як шахрайство з фальшивими вакансіями, часто з обіцянками високої оплати праці та відсутності необхідних навичок, мають на меті заманити людей у шахрайство з роботою з дому.

```
[SYSTEM] Loading AI-style ML model...
[SYSTEM] Model loaded successfully
[SYSTEM] Gmail monitoring started...

=====
[EMAIL RECEIVED]
From: "Бобер Наталія" <bobernataliaviktorovna14112002@gmail.com>
Subject: Job Openings at UNICEF
Body:
I am sharing the job openings with employees who may be interested in a
part-time paid position at UNICEF at a weekly salary of $600.00.

Additional information on the job details is attached.

Please follow the steps in the attached document and contact Henrietta H.
Fore with your unofficial email address (e.g. Gmail, Yahoo, Hotmail,
 iCloud, etc.) for more information about the position.

Take notes; this is a work-from-home position only.

Sincerely, Human Resources

[AI-STYLE ANALYSIS]
AI-style probability: 0.159
Verdict: HUMAN-WRITTEN STYLE
[OK] Result saved to Бобер Наталія-2025-12-21_19-33-01.txt
```

3.10 – Аналіз реального фішингового листа написного людиною

Отже, отримані результати тестування створеної програми свідчать про коректну роботу програмної реалізації та здатність навченої моделі виявляти фішингові електронні листи згенеровані штучним інтелектом. Проведене тестування на електронних листах, які не входили до навчального датасету, підтверджує здатність моделі узагальнювати набуті знання і навички та коректно працювати з новими вхідними даними. Це доводить доцільність застосування запропонованого підходу для автоматизованого аналізу електронної пошти та підвищення рівня захисту користувачів від сучасних фішингових атак.

Висновок до розділу 3

В даному розділі було створено та детально описано програмну реалізацію моделі роботи програми перевірки вхідного електронного листа на виявлення в

ньому AI-подібного стилю. Описано алгоритм, вибір мови програмування, функції та модулі, які були використані для написання програмного коду, використані датасети для машинного навчання. Проведено тестування програмної реалізації, результати якого подані у вигляді скріншотів.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було проведено дослідження особливостей соціальної інженерії, етапів та методів її проведення. Розглянуто найпоширеніший метод соціальної інженерії – фішинг та найпоширеніший канал атак – електронну пошту. Також розглянуто як вплив ШІ допомагає фішинговим атакам розвиватись. Показано, що використання сучасних генеративних моделей значно ускладнює традиційні методи фільтрації фішингових повідомлень.

У другому розділі проведено аналіз поняття штучного інтелекту та його основних характеристик та компонентів. Розглянуто як за допомогою найпопулярніших загальнодоступних моделей генеративного штучного інтелекту можна згенерувати фішингові листи. Також розглянуто методи виявлення тексту створеного штучним інтелектом, а саме лінгвістичний аналіз та обробка природної мови, моделі машинного та глибокого навчання, часовий та мережний аналіз.

Третій розділ роботи був присвячений розробці програмної реалізації. У ході дослідження було сформовано та підготовлено два набори фішингових електронних листів: згенерованих ШІ та написаних людиною. Було оброблено та відфільтровано дані датасети листів від не потрібних полів та символів, що дозволило забезпечити коректність стилістичного аналізу та уникнути впливу сторонніх факторів на результати навчання моделі. На основі виділених лінгвістичних ознак було проведено експериментальне навчання та тестування моделі машинного навчання на базі алгоритму Random Forest. Крім того, було розроблено програмну реалізацію для автоматичного аналізу електронної пошти з інтеграцією Gmail API, яка дозволяє здійснювати перевірку вхідних повідомлень у реальному режимі часу.

Проведене тустування підтвердило стабільність роботи програмного засобу та можливість його практичного застосування як допоміжного інструменту аналізу електронної пошти.

Оримані результати підтверджують актуальність обраної теми та доцільність подальших досліджень у цій сфері в поєднанні інших підходів до виявлення фішингових атак, що може сприяти підвищенню рівня інформаційної безпеки в системах електронної пошти.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] What is Social Engineering? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.paloaltonetworks.com/cyberpedia/what-is-social-engineering>
- [2] Топ-5 методів соціальної інженерії та як захиститися від них [Електронний ресурс] – Режим доступу до ресурсу: <https://nwu.ua/blog/statti/top-5-metodiv-sotsialnoyi-inzheneriyi-ta-yak-zahistitisya-vid-nih/>
- [3] Artificial Intelligence's Impact on Social Engineering Attacks [Електронний ресурс] – Режим доступу до ресурсу: <https://opus.govst.edu/cgi/viewcontent.cgi?article=1521&context=capstones#:~:text=Thanks%20to%20artificial%20intelligence%2C%20social,that%20people%20can%20click%20on.>
- [4] Cybersecurity threatscape: H1 2025 [Електронний ресурс] – Режим доступу до ресурсу: <https://global.ptsecurity.com/en/research/analytics/aktual-nye-kiberugrozy-i-ii-kvartaly-2025-goda/#id1>
- [5] What is phishing? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/think/topics/phishing>
- [6] A Multivocal Literature Review on Growing Social Engineering Based Cyber-Attacks/ Threats During the COVID-19 Pandemic: Challenges and Prospective Solutions [Електронний ресурс] – Режим доступу до ресурсу: [\(PDF\) A Multivocal Literature Review on Growing Social Engineering Based Cyber-Attacks/ Threats During the COVID-19 Pandemic: Challenges and Prospective Solutions](#)
- [7] Що таке фішинг і чому він небезпечний для компаній [Електронний ресурс] – Режим доступу до ресурсу: <https://portaltele.com.ua/news/internet/shho-take-fishyng-i-chomu-vin-nebezpechnyj-dlya-kompanij.html>
- [8] Cost of a Data Breach Report 2025 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/reports/data-breach>
- [9] What is generative AI? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/think/topics/generative-ai>

- [10] Phishing Trends Report (Updated for 2025) [Електронний ресурс] – Режим доступу до ресурсу: <https://hoxhunt.com/guide/phishing-trends-report#ai-being-used-to-amplify-phishing-attacks>
- [11] What is the Role of AI in Phishing Attacks [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techmagic.co/blog/blog-phishing-attack-statistics#what-is-the-role-of-ai-in-phishing-attacks>
- [12] Росія активно використовує штучний інтелект для розсилки фішингу [Електронний ресурс] – Режим доступу до ресурсу: <https://forbes.ua/news/rosiya-aktivno-vikoristovue-shtuchniy-intelekt-dlya-rozsilki-fishingu-i-otruiye-dani-dlya-shi-modeley-gendirektor-cisco-ukraina-28032025-28392>
- [13] What is AI? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/think/topics/artificial-intelligence>
- [14] Artificial Intelligence [Електронний ресурс] – Режим доступу до ресурсу: <https://www.it.ua/en/knowledge-base/technology-innovation/artificial-intelligence>
- [15] Components of Artificial Intelligence (AI) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.appliedaicourse.com/blog/components-of-artificial-intelligence/>
- [16] Defining Artificial Intelligence [Електронний ресурс] – Режим доступу до ресурсу: <https://www.nasa.gov/what-is-artificial-intelligence/>
- [17] Unveiling the Top 10 Defining Characteristics of AI [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaashivinfotech.com/blog/top-10-characteristics-of-artificial-intelligence/>
- [18] What is ChatGPT? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/whatis/definition/ChatGPT>
- [19] What is Microsoft Copilot? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/whatis/definition/Microsoft-Copilot>
- [20] DeepAI: What It Is and Best Alternative [2025] [Електронний ресурс] – Режим доступу до ресурсу: <https://www.voiceflow.com/blog/deepai>

- [21] Advanced Techniques for Detecting AI-Generated Content and Misinformation [Электронный ресурс] – Режим доступа до ресурсу:
<https://londondataconsulting.medium.com/advanced-techniques-for-detecting-ai-generated-content-and-misinformation-d94776425f29>
- [22] What is Python? Executive Summary [Электронный ресурс] – Режим доступа до ресурсу: <https://www.python.org/doc/essays/blurb/>
- [23] Огляд модуля os Python [Электронный ресурс] – Режим доступа до ресурсу: <https://www.it-notes.wiki/python/overview-os-python-module/>
- [24] time — Time access and conversions [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.python.org/3/library/time.html>
- [25] datetime — Basic date and time types [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.python.org/3/library/datetime.html>
- [26] base64 — Base16, Base32, Base64, Base85 Data Encodings [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.python.org/3/library/base64.html>
- [27] traceback [Электронный ресурс] – Режим доступа до ресурсу:
<https://realpython.com/ref/stdlib/traceback/>
- [28] re — Regular expression operations [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.python.org/uk/3/library/re.html>
- [29] Google API Client [Электронный ресурс] – Режим доступа до ресурсу:
<https://pypi.org/project/google-api-python-client/>
- [30] google-auth [Электронный ресурс] – Режим доступа до ресурсу:
<https://anaconda.org/channels/conda-forge/packages/google-auth/overview>
- [31] google-auth-oauthlib [Электронный ресурс] – Режим доступа до ресурсу:
<https://anaconda.org/channels/conda-forge/packages/google-auth-oauthlib/overview>
- [32] NumPy [Электронный ресурс] – Режим доступа до ресурсу:
<https://en.wikipedia.org/wiki/NumPy>
- [33] scikit-learn [Электронный ресурс] – Режим доступа до ресурсу:
<https://en.wikipedia.org/wiki/Scikit-learn>

- [34]Joblib: What is this Python library? How do I use it? [Электронный ресурс] – Режим доступа до ресурсу: <https://datascientest.com/en/joblib-what-is-this-python-library-how-do-i-use-it>
- [35]pandas (software) [Электронный ресурс] – Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Pandas_\(software\)](https://en.wikipedia.org/wiki/Pandas_(software))
- [36] Natural Language Toolkit [Электронный ресурс] – Режим доступа до ресурсу: https://ru.wikipedia.org/wiki/Natural_Language_Toolkit
- [37] A dataset of AI-generated phishing email [Электронный ресурс] – Режим доступа до ресурсу: https://people.cs.ksu.edu/~lshamir/data/ai_phishing/
- [38] Phishing-Email-Dataset [Электронный ресурс] – Режим доступа до ресурсу: https://github.com/rokibulroni/Phishing-Email-Dataset?utm_source=chatgpt.com
- [39]10 real phishing email examples [Электронный ресурс] – Режим доступа до ресурсу: <https://us.norton.com/blog/online-scams/phishing-email-examples>

clean_dataset.py

```
import os
import re

MIN_LEN = 50 # мінімальна довжина тексту для аналізу

def clean_text(text):

    # Видалити email headers
    text = re.sub(
        r"^(From|To|Cc|Bcc|Subject|Date|Received|Return-Path|Message-ID):.*$",
        "",
        text,
        flags=re.MULTILINE | re.IGNORECASE
    )

    #Видалити HTML, але залишити текст
    text = re.sub(r"<[^>]+>", " ", text)

    #Прибрати HTML entities
    text = re.sub(r"&nbsp;|&lt;|&gt;|&amp;", " ", text)

    #Видалити службові роздільники
    text = re.sub(r"-{2,}", " ", text)

    #Нормалізувати пробіли
    text = re.sub(r"\s+", " ", text)

    return text.strip()

def process_folder(folder_path):
    removed = 0
    kept = 0

    for fname in os.listdir(folder_path):
        if not fname.lower().endswith(".txt"):
            continue

        path = os.path.join(folder_path, fname)
```

```

with open(path, "r", encoding="utf-8", errors="ignore") as f:
    raw_text = f.read()

cleaned = clean_text(raw_text)

# Відфільтрувати порожні листи
if len(cleaned) < MIN_LEN:
    os.remove(path)
    removed += 1
    continue

with open(path, "w", encoding="utf-8") as f:
    f.write(cleaned)

kept += 1

print(f"[OK] {folder_path}: kept={kept}, removed={removed}")

if __name__ == "__main__":
    print("[SYSTEM] Cleaning dataset...")

    process_folder("dataset/ai")
    process_folder("dataset/human")

    print("[SYSTEM] Cleaning completed")

```

features.py

```

import re
import nltk
from collections import Counter

#Попередня обробка
def preprocess(text):
    text = text.lower()
    return text

#Основні статистики
def basic_stats(text):
    sentences = nltk.sent_tokenize(text)
    words = nltk.word_tokenize(text)

    num_sent = len(sentences)
    num_words = len(words)

    avg_sent_len = num_words / num_sent if num_sent else 0
    avg_word_len = sum(len(w) for w in words) / num_words if num_words else 0

    return {
        "num_sentences": num_sent,
        "num_words": num_words,
        "avg_sentence_length": round(avg_sent_len, 2),
        "avg_word_length": round(avg_word_len, 2)
    }

#Лексична щільність
def lexical_metrics(text):
    words = nltk.word_tokenize(text)
    unique_words = set(words)

    ttr = len(unique_words) / len(words) if words else 0
    density = ttr * 100

    freq = Counter(words)
    repeated_words_ratio = sum(1 for w in freq if freq[w] > 3) / len(freq) if freq else 0

    return {
        "lexical_density": round(density, 2),
        "type_token_ratio": round(ttr, 3),
    }

```

```

        "repeated_words_ratio": round(repeated_words_ratio, 3)
    }

#ЧаСТИНИ МОВИ
def pos_metrics(text):
    words = nltk.word_tokenize(text)
    tagged = nltk.pos_tag(words)

    pos_counts = Counter(tag for _, tag in tagged)
    total = sum(pos_counts.values())

    def ratio(tags):
        return round(sum(pos_counts[t] for t in tags) / total, 3) if total else 0

    return {
        "pronoun_ratio": ratio(["PRP", "PRP$"]),
        "verb_ratio": ratio(["VB", "VBD", "VBG", "VBN", "VBP", "VBZ"]),
        "noun_ratio": ratio(["NN", "NNS", "NNP", "NNPS"])
    }

#Структурні
def structural_metrics(text):
    return {
        "num_urls": len(re.findall(r"http[s]?://", text)),
        "num_exclamations": text.count("!"),
        "num_questions": text.count("?")
    }

#ГОЛОВНА ФУНКЦІЯ
def extract_features(text):
    text = preprocess(text)

    features = {}
    features.update(basic_stats(text))
    features.update(lexical_metrics(text))
    features.update(pos_metrics(text))
    features.update(structural_metrics(text))

    return features

```

train_model.py

```
import os
import pandas as pd
from joblib import dump

from sklearn.model_selection import train_test_split, StratifiedKFold,
cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

from features import extract_features

#CONFIG

DATASET_DIR = "dataset"
MODEL_OUT = "model/ai_style_model.pkl"
RANDOM_STATE = 42

#LOAD DATA

X, y = [], []

print("[INFO] Loading dataset...")

for label, folder in [(1, "ai"), (0, "human")]:
    folder_path = os.path.join(DATASET_DIR, folder)

    for fname in os.listdir(folder_path):
        if not fname.endswith(".txt"):
            continue

        with open(os.path.join(folder_path, fname), "r", encoding="utf-8",
errors="ignore") as f:
            text = f.read()

            features = extract_features(text)
            X.append(features)
            y.append(label)

X = pd.DataFrame(X)
y = pd.Series(y)
```

```

print("[INFO] Samples:", len(y))
print(y.value_counts())

#MODEL

model = RandomForestClassifier(
    n_estimators=300,
    max_depth=12,
    random_state=RANDOM_STATE,
    class_weight="balanced"
)

#CROSS-VALIDATION

print("\n[INFO] Cross-validation (F1-score)...")

cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=RANDOM_STATE)
cv_scores = cross_val_score(
    model,
    X,
    y,
    cv=cv,
    scoring="f1"
)

print("F1 scores:", cv_scores)
print("Mean F1:", round(cv_scores.mean(), 3))

#TRAIN/TEST SPLIT

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=RANDOM_STATE,
    stratify=y
)

#FINAL TRAINING

print("\n[INFO] Training final model...")
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

```

```

print("\n=== TEST RESULTS ===")
print(classification_report(y_test, y_pred, target_names=["Human", "AI"]))

#FEATURE IMPORTANCE

print("\n=== FEATURE IMPORTANCE ===")
importances = model.feature_importances_

for feat, score in sorted(zip(X.columns, importances), key=lambda x: x[1],
reverse=True):
    print(f'{feat}: {round(score, 3)}')

#SAVE MODEL

os.makedirs("model", exist_ok=True)
dump(model, MODEL_OUT)

print(f"\n[OK] Model saved to {MODEL_OUT}")

```

monitor_email.py

```
from __future__ import print_function
import os
import base64
import datetime
import time
import traceback
import pandas as pd

from google.auth.transport.requests import Request
from google.oauth2.credentials import Credentials
from google_auth_oauthlib.flow import InstalledAppFlow
from googleapiclient.discovery import build

from joblib import load
from features import extract_features

#CONFIG

MODEL_PATH = "model/ai_style_model.pkl"
CHECK_INTERVAL = 60 #seconds

SCOPES = [
    "https://www.googleapis.com/auth/gmail.readonly",
    "https://www.googleapis.com/auth/gmail.modify"
]

#LOAD MODEL

print("[SYSTEM] Loading AI-style ML model...")
model = load(MODEL_PATH)
print("[SYSTEM] Model loaded successfully")

#GMAIL AUTH

def get_gmail_service():
    creds = None

    if os.path.exists("token.json"):
        creds = Credentials.from_authorized_user_file("token.json", SCOPES)

    if not creds or not creds.valid:
```

```

if creds and creds.expired and creds.refresh_token:
    creds.refresh(Request())
else:
    flow = InstalledAppFlow.from_client_secrets_file(
        "credentials.json", SCOPES
    )
    creds = flow.run_local_server(port=0)

with open("token.json", "w") as token:
    token.write(creds.to_json())

return build("gmail", "v1", credentials=creds)

#FETCH EMAIL

def get_new_email_details(service):
    try:
        res = service.users().messages().list(
            userId="me",
            labelIds=["INBOX", "UNREAD"],
            maxResults=1
        ).execute()

        messages = res.get("messages", [])
        if not messages:
            return None

        msg_id = messages[0]["id"]
        msg = service.users().messages().get(
            userId="me",
            id=msg_id
        ).execute()

        headers = msg["payload"]["headers"]
        subject = next((h["value"] for h in headers if h["name"] == "Subject"), "")
        sender = next((h["value"] for h in headers if h["name"] == "From"), "")

        body = ""
        payload = msg["payload"]

        if "parts" in payload:
            for part in payload["parts"]:
                if part["mimeType"] == "text/plain":
                    data = part["body"].get("data")

```

```

        if data:
            body = base64.urlsafe_b64decode(data).decode(
                "utf-8", errors="ignore"
            )
            break

    if not body:
        body = msg.get("snippet", "")

    # mark as read
    service.users().messages().modify(
        userId="me",
        id=msg_id,
        body={"removeLabelIds": ["UNREAD"]}
    ).execute()

    return sender, subject, body

except Exception as e:
    print("[ERROR] Failed to read email:", e)
    print(traceback.format_exc())
    return None

#LOGGING

def save_log(sender, subject, features, ai_proba, verdict):
    try:
        sender_name = sender.split("'")[1] if "'" in sender else sender.split("<")[0].strip()
    except:
        sender_name = "unknown_sender"

    ts = datetime.datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
    filename = f'{sender_name}-{ts}.txt'

    content = (
        f'From: {sender}\n'
        f'Subject: {subject}\n'
        f'{'-'*60}\n'
        f'AI-style probability: {round(ai_proba, 3)}\n'
        f'Verdict: {verdict}\n'
        f'{'-'*60}\n'
        f'Extracted features:\n'
    )

```

```

for k, v in features.items():
    content += f'{k}: {v}\n'

os.makedirs("mails", exist_ok=True)
with open(os.path.join("mails", filename), "w", encoding="utf-8") as f:
    f.write(content)

print(f'[OK] Result saved to {filename}')

#MAIN LOOP

def continuously_check_email(service):
    print("[SYSTEM] Gmail monitoring started...")

    while True:
        details = get_new_email_details(service)

        if details:
            sender, subject, body = details

            print("\n=====")
            print("[EMAIL RECEIVED]")
            print(f'From: {sender}')
            print(f'Subject: {subject}')
            print(f'Body:\n{body}\n')

            if not body.strip():
                time.sleep(CHECK_INTERVAL)
                continue

            #FEATURE EXTRACTION
            features = extract_features(body)
            X = pd.DataFrame([features])

            #ML PREDICTION
            ai_proba = model.predict_proba(X)[0][1]

            #FINAL DECISION =====
            if ai_proba >= 0.6:
                verdict = "AI-GENERATED STYLE"
            else:
                verdict = "HUMAN-WRITTEN STYLE"

            #OUTPUT

```

```
print("[AI-STYLE ANALYSIS]")
print(f"AI-style probability: {round(ai_proba, 3)}")
print(f"Verdict: {verdict}")

save_log(sender, subject, features, ai_proba, verdict)

time.sleep(CHECK_INTERVAL)

#RUN
if __name__ == "__main__":
    service = get_gmail_service()
    continuously_check_email(service)
```