

Міністерство освіти і науки України  
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій  
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:  
В.о. завідувача кафедри  
кібербезпеки та захисту інформації  
Іван ПАРХОМЕНКО  
«17» травня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи

галузь знань 12 Інформаційні технології  
(шифр і назва галузі знань)  
спеціальність 125 Кібербезпека  
(код і назва спеціальності)  
освітній ступень магістр  
освітньо-наукова програма Кібербезпека  
(назва освітньої програми)

на тему: «Методи виявлення та протидії фішинговим атакам, згенерованих за допомогою штучного інтелекту»

Виконавець: студентка II курсу, групи КБм-21

(підпис) Анастасія ЗАВГОРОДНЯ  
(Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	Підпис
Науковий керівник	Володимир НАКОНЕЧНИЙ	
Нормоконтроль	Сергій ДАКОВ	

Міністерство освіти і науки України  
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій  
Кафедра кібербезпеки та захисту інформації

**ЗАТВЕРДЖЕНО:**

В.о. завідувача кафедри  
кібербезпеки та захисту інформації  
Іван ПАРХОМЕНКО  
«17» листопада 2023 р.

**ЗАВДАННЯ**  
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека  
(код і назва спеціальності)

освітній ступень магістр

Здобувача(ки) КБМ-21 Завгородньої Анастасії Сергіївни  
(група) (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Методи виявлення та протидії фішинговим атакам, згенерованих за допомогою штучного інтелекту

**1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ**

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 15.11.2023 р.

**2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ**

**Об'єкт досліджень** Процес виявлення тексту, згенерованого штучним інтелектом.

**Предмет досліджень** Методи та технології виявлення контенту, написаного штучним інтелектом.

**Мета** Розробка програмної реалізації, що допоможе виявляти листи, для написання яких використовувався штучний інтелект

**Вихідні дані для проведення роботи**

Теоретичні та практичні дослідження виявлення тексту, згенерованого штучним інтелектом.

### 3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

**Наукова новизна** Набув подальшого розвитку процес виявлення текстів листів, згенерованих штучним інтелектом, що допомагатиме в боротьбі з фішингом.

**Практична цінність** Реалізація розробленого програмного забезпечення допоможе на початкових етапах проводити аналіз електронних листів на предмет використання інструментів штучного інтелекту при їх створенні

### 4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

### 5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Уточнення постановки задачі	17.11.2023 – 21.01.2024
Аналіз літературних джерел	22.01.2024 – 04.02.2024
Дослідження загрози фішингу та його видів	05.02.2024 – 18.02.2024
Аналіз статичних даних здійснення фішингових атак за останні кілька років	19.02.2024 – 22.02.2024
Аналіз роботи штучного інтелекту та методів його використання для написання фішингового контенту	23.02.2024 – 28.02.2024
Огляд можливостей відомих інструментів штучного інтелекту створювати шаблони фішингових листів	29.02.2024 – 03.03.2024
Аналіз роботи методів виявлення тексту, згенерованого за допомогою інструментів штучного інтелекту	04.03.2024 – 10.03.2024
Розробка алгоритму роботи програми для виявлення листів текст яких, був написаний за допомогою штучного інтелекту	11.03.2024 – 17.03.2024

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Розробка програмної реалізації, яка допомагає виявити ймовірність використання штучного інтелекту при створенні електронного листа	18.03.2024 – 21.04.2024
Оформлення пояснювальної записки згідно методичних рекомендацій	22.04.2024 – 12.05.2024
Подача пакету документів на розгляд ЕК	13.05.2024 – 18.05.2024

## 6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

**Економічний ефект**                      Зниження ризиків інформаційної безпеки, пов'язаних з використанням штучного інтелекту для здійснення атак.

---

**Соціальний ефект**                      Покращення технологій виявлення текстів, що були створені штучним інтелектом.

---

## 7. ДОДАТКОВІ ВИМОГИ

---



---

Завдання видав

\_\_\_\_\_ (підпис)

Володимир НАКОНЕЧНИЙ  
(Ім'я, ПРІЗВИЩЕ)

Завдання прийняв  
до виконання

\_\_\_\_\_ (підпис)

Анастасія ЗАВГОРОДНЯ  
(Ім'я, ПРІЗВИЩЕ)

Дата видачі завдання: 17.11.2023 р.  
Термін подання кваліфікаційної роботи до ЕК 17.05.2024 р.

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Методи виявлення та протидії фішинговим атакам, згенерованих за допомогою штучного інтелекту»: 85 сторінок, 36 рисунки та 4 таблиці, 50 літературних джерел.

Об'єкт дослідження – процес виявлення тексту, згенерованого штучним інтелектом.

Мета роботи – розробка програмної реалізації, що допоможе виявляти листи, для написання яких використовувався штучний інтелект.

Методи дослідження – аналіз зібраних даних, формалізація та конкретизація, дослідження наявних методологій.

Наукова новизна: набув подальшого розвитку процес виявлення текстів листів, згенерованих штучним інтелектом, що допомагатиме в боротьбі з фішингом.

Актуальність теми: сучасні спроби фішингу, керовані штучним інтелектом, дуже персоналізовані, використовують дані (такі як поведінка користувачів в Інтернеті, інтереси та моделі спілкування) для створення цілеспрямованих і вражаюче переконливих повідомлень. Тема кваліфікаційної роботи є актуальною, оскільки важливо вміти розрізняти зміст, створений людьми, від змісту, створеного штучним інтелектом. Ідентифікація тексту, написаного інструментами штучного інтелекту, допомагає підтримувати прозорість і цілісність спілкування. Також виявлення таких текстів допомагає користувачам приймати обґрунтовані рішення щодо надійності та достовірності інформації, з якою вони стикаються в Інтернеті.

Ключові слова: фішинг, фішинговий лист, згенерований текст, штучний інтелект, AI, ШІ.

**СПИСОК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

AI	–	Artificial Intelligence
API	–	Application Programming Interface
LLM	–	Large Language Model
NLP	–	Natural Language Processing
OSINT	–	Open Source Intelligence
SVM	–	Support Vector Machine
URL	–	Uniform Resource Locator
ШІ	–	Штучний Інтелект
ШПЗ	–	Шкідливе Програмне Забезпечення

## ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1 АНАЛІЗ ЗАГРОЗИ ФІШИНГУ .....	12
1.1. Що таке фішинг?.....	12
1.2. Види фішингу.....	14
1.2.1. Фішинг через електронну пошту.....	14
1.2.2. Цільовий фішинг .....	17
1.2.3. Whaling .....	18
1.2.4. Смішинг.....	19
1.2.5. Angler phishing .....	21
1.2.6. Клон фішинг.....	22
1.2.7. Domain Phishing .....	24
1.3. Статистика здійснення фішингових атак.....	26
1.4. Загальні методи виявлення фішингу .....	30
Висновки до першого розділу.....	30
РОЗДІЛ 2 ШТУЧНИЙ ІНТЕЛЕКТ ТА ЙОГО ВИКОРИСТАННЯ ДЛЯ НАПИСАННЯ ФІШИНГОВИХ ЛИСТІВ.....	32
2.1. Що являє собою штучний інтелект?.....	32
2.2. Що таке алгоритми штучного інтелекту та як вони працюють? .....	33
2.2.1. Алгоритми контрольованого навчання.....	33
2.2.2. Алгоритми неконтрольованого навчання.....	36
2.2.3. Алгоритми навчання з підкріпленням .....	38
2.3. Штучний інтелект та фішинг .....	40
2.4. Статистика відкриттів листів, створених штучним інтелектом .....	42

2.5. Використання ШІ для написання фішингових листів .....	43
2.5.1. ChatGPT .....	43
2.5.2. Bard .....	46
2.5.3. WormGPT .....	47
2.6. Генерація фішингового листа за допомогою штучного інтелекту .....	48
Висновки до другого розділу .....	51
<b>РОЗДІЛ 3 МЕТОДИ ВИЯВЛЕННЯ ТЕКСТУ, ЩО БУВ СТВОРЕНИЙ</b>	
<b>ШТУЧНИМ ІНТЕЛЕКТОМ.....</b>	<b>52</b>
3.1. Використання детекторів ШІ .....	52
3.1.1. Принцип роботи програмних детекторів .....	53
3.1.2. Потенційна альтернатива: водяні знаки .....	55
3.1.3. Відмінність детекторів штучного інтелекту від інструментів перевірки на плагіат.....	55
3.2. Аналіз природності використання мови та стилю письма .....	56
3.3. Перевірка достовірності джерел.....	58
3.4. Аналіз n-грам.....	59
3.5. Лексична насиченість .....	60
Висновки до третього розділу .....	61
<b>РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВИЯВЛЕННЯ ЙМОВІРНОСТІ</b>	
<b>ВИКОРИСТАННЯ ШІ ПРИ СТВОРЕННІ ЕЛЕКТРОННОГО ЛИСТА.....</b>	<b>62</b>
4.1. Призначення програмної реалізації .....	62
4.2. Склад і характеристики потоків вхідної та вихідної інформації .....	63
4.3. Вибір мови програмування .....	63
4.4. Використані модулі Python .....	65
4.5. Програмні файли та їх функції .....	66

4.6. Тестування програмної реалізації .....	67
Висновок до четвертого розділу.....	78
ВИСНОВКИ.....	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	81
ДОДАТОК А.....	86
ДОДАТОК Б .....	87
ДОДАТОК В.....	90

## ВСТУП

Актуальність роботи: У минулому фішингові атаки було відносно легко помітити через їх загальний характер і часто невміле виконання. Незважаючи на орфографічні помилки, неточну інформацію про одержувача та інші непомітні недоліки, звичайний користувач міг би визначити, чи був електронний лист легітимним чи ні. Але штучний інтелект змінив все.

Сучасні спроби фішингу, керовані штучним інтелектом, дуже персоналізовані, використовують дані (такі як поведінка користувачів в Інтернеті, інтереси та моделі спілкування) для створення цілеспрямованих і вражаюче переконливих повідомлень. Це означає, що виявлення фішингу стало набагато складнішою задачею з якою стикаються користувачі.

Здатність штучного інтелекту автоматизувати масові атаки та писати текст фішингових повідомлень означає, що компанії та установи все частіше стикаються з величезною кількістю складних спроб фішингу, що збільшує кількість успішних спроб взлому. Проблема ускладнюється тим фактом, що традиційні засоби кібербезпеки часто погано налаштовані та пристосовані для виявлення цих загроз, що призводить до потреби в більш просунутих рішеннях безпеки.

Тема кваліфікаційної роботи є актуальною, оскільки важливо вміти розрізняти вміст, створений людьми, від вмісту, створеного штучним інтелектом. Ідентифікація тексту, написаного інструментами штучного інтелекту, допомагає підтримувати прозорість і цілісність спілкування. Також виявлення таких текстів допомагає користувачам приймати обґрунтовані рішення щодо надійності та достовірності інформації, з якою вони стикаються в Інтернеті. Крім того, розрізнення між контентом створеним людьми та штучним інтелектом допомагає запобігти дезінформації.

Мета дослідження: розробка програмної реалізації, що допоможе виявляти листи, для написання яких використовувався штучний інтелект.

Завдання дослідження: для досягнення поставленої мети необхідно вирішити наступні задачі:

- Виконати аналіз матеріалів по темі фішинг.
- Провести аналіз роботи штучного інтелекту та його використання для написання фішингового контенту.
- Проаналізувати роботу існуючих методів виявлення тексту, згенерованого за допомогою інструментів штучного інтелекту.
- Розробити алгоритм роботи програми для виявлення листів, згенерованих штучним інтелектом.
- Виконати програмну реалізацію та провести тестування роботи програми.

Об'єкт дослідження: процес виявлення тексту, згенерованого штучним інтелектом.

Предмет дослідження: методи та технології виявлення контенту, написаного штучним інтелектом

Методи дослідження: аналіз зібраних даних, формалізація та конкретизація, дослідження наявних методологій.

Наукова новизна: набув подальшого розвитку процес виявлення текстів листів, згенерованих штучним інтелектом, що допомагатиме в боротьбі з фішингом.

Практична цінність: отримана програмна реалізація може допомогти на початкових етапах аналізу електронних листів на предмет використання інструментів штучного інтелекту для їх створення.

## РОЗДІЛ 1

### АНАЛІЗ ЗАГРОЗИ ФІШИНГУ

Повсюдне поширення онлайн-комунікацій, зокрема електронної пошти, призвело до того, що фішингові атаки стали наймовірнішим поширеним явищем; а також проведення даного виду атак є відносно дешевим для зловмисників.

#### 1.1. Що таке фішинг?

Фішинг – це одна з форм кіберзлочинності. Ціль фішингу – змусити жертву передати зловмиснику цінну конфіденційну інформацію( особисті дані, логін, пароль, фінансові дані, тощо) або перевести гроші на підроблені рахунки. Фішинг зазвичай здійснюється через електронні листи, однак даний вид шахрайства розповсюдився далеко за межі підозрілих електронних листів, а саме на телефонні дзвінки, смс-повідомлення, повідомлення в соціальних мережах.

Зловмисники запускають фішингові атаки, щоб отримати несанкціонований доступ до конфіденційної інформації жертви. Найбільш типовими мотивами їх дій є [1]:

- **Фінансова вигода.** Більшість атак соціальної інженерії мотивуються отриманням фінансової винагороди, більшість фішингових кампаній мають той же мотив. Злочинці обманом змушують людей передавати власні фінансові дані, щоб потім вкрати гроші з банківських рахунків жертв або виконати несанкціоновані транзакції.

- **Збір даних облікових записів.** В цьому випадку також працює мотивація фінансовою вигодою, так як зазвичай такий вид інформації невдовзі після отримання продається злочинцями в даркнеті.

- **Викрадення особистих даних.** Коли зловмисники викрадають особисту інформацію своїх жертв, таку як номери кредитних карт, логіни(імена користувачів), паролі від облікових записів, тощо, то зазвичай потім такі дані

використовуються для того, щоб злочинець міг видати себе за когось іншого(за особу жертви), щоб здійснювати шахрайські дії, або отримати доступ до облікових записів жертви в Інтернеті.

- **Поширення шкідливого програмного забезпечення (ШПЗ).** Іноді фішингові атаки, все таки, використовуються для поширення шкідливих програм. Це можуть бути різноманітні віруси або програми-шантажисти(ransomware), що дозволяють хакеру отримати доступ над робочим пристроєм, викрасти інформацію чи вимагати винагороду за повернення доступу над заблокованим комп'ютером.

- **Стеження та шпигунство.** Іноді фішингові кампанії проводяться державними або організованими групами кіберзлочинців для збору розвідувальних даних чи стеження за певними особами. Також можуть бути використані для проведення стеження за бізнесом чи компанією та викрадення необхідної конфіденційної інформації.

- **Нанесення репутаційних збитків.** В такому випадку, зловмисники видають себе за певну особу чи компанію та від нібито її імені здійснюють розсилання шахрайських електронних листів чи повідомлень с метою нанесення шкоди репутації.

Фішингові атаки грають дуже важливу роль в багатьох видах кіберзлочинності. Основна мета фішингової атаки – це переконати користувача перейти по посиланню, що було відправлено в фішинговому повідомленні, яке або встановить на комп'ютер користувача вірус чи інший вид шкідливого програмного забезпечення, таким чином надавши зловмиснику доступ до мережі компанії, або відкриє підроблений сайт, куди жертва внесе потрібні кіберзлочинцю дані. Приклад схеми дії фішингової кампанії зображений на рисунку 1.1.

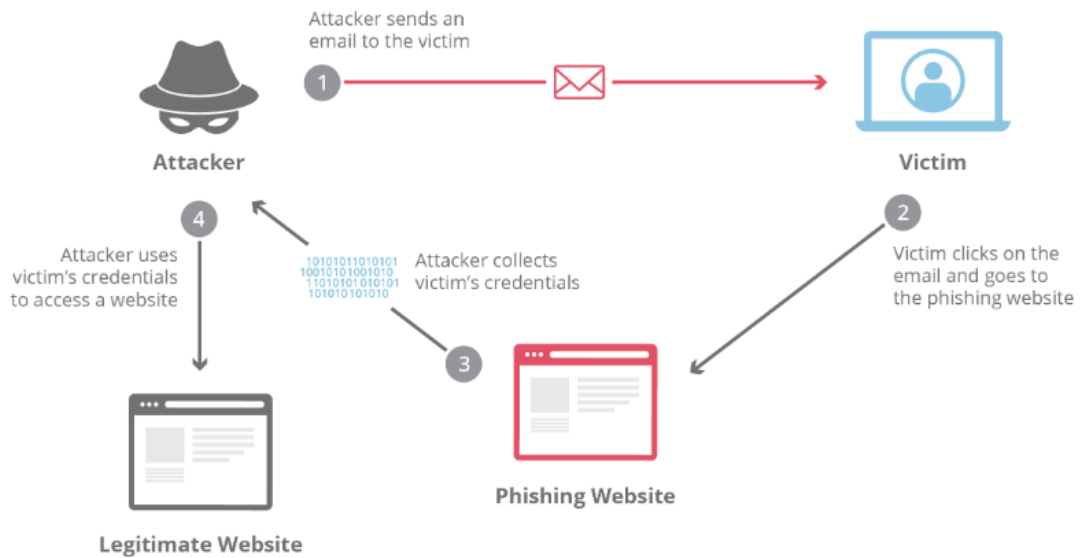


Рисунок 1.1 – Приклад організації фішингової атаки [2]

За статистикою основним компонентом майже кожного великого витoku даних є саме фішинг. Для прикладу можна навести відому атаку 2014 року на компанію Sony Pictures. Вона виникла в результаті серії електронних листів, що були розіслані керівництву компанії. Як результат, в загальному компанія втратила приблизно 100 терабайт інформації, куди ввійшли персональні дані працівників та неопубліковані фільми [3].

## 1.2. Види фішингу

### 1.2.1. Фішинг через електронну пошту

Фішинг через електронну пошту є одним з найпоширенішим типом фішингових атак, який існує ще з 1996 року. Дана атака містить попередньо згенерований електронний лист, який виглядає цілком легітимним та націлений на великий список електронних адрес [4].

Електронна пошта – це найпопулярніший інструмент для здійснення фішингових кампаній. Зловмисники реєструють підроблені домени та видають

себе за справжні організації. Підроблені домени зазвичай містять заміну символі, для прикладу, використовуються «r» і «n» поруч, щоб отримати «rn» замість «m» [5].

У електронних листах шахраї намагаються змусити людей перейти за посиланнями, які будуть вести на підроблені вебсайти, призначені для викрадення паролів та інших конфіденційних даних (див. рисунок 1.2).



Рисунок 1.2 – Приклад фішингового листа [6]

Крім того, зловмисники зазвичай намагаються спонукати об'єкти атаки до дії, створюючи відчуття терміновості. Наприклад, електронний лист може загрожувати завершенням терміну дії облікового запису. Застосування такого тиску змушує користувача бути менш пильним та більш схильним до здійснення помилок.

Посилання ж в листах дуже схожі на свої легітимні відповідники, але зазвичай мають доменне ім'я або додаткові субдомени з помилками. Подібність між такими адресами створює користувачу враження захищеного зв'язку.

Складові елементи більшості електронних листів [7]:

- **Тема листа.** Важливий елемент фішингового електронного листа, рядок теми призначений для спонукання жертви, щоб вона відкрила електронний лист. В епоху більш персоналізованих атак хакери часто пишуть чітко націлені

теми, щоб спонукати жертв відкривати електронні листи. «Націлювання» може бути таким простим, як додавання слова «терміново» до рядка теми або посилання на справжні рахунки-фактури чи інші ланцюжки електронних листів, щоб виглядати легітимними.

- **Підробка електронної пошти.** Спуфінг електронної пошти передбачає створення електронної адреси, яка виглядає як адреса справжньої компанії. Двома найпоширенішими формами спуфінгу електронної пошти є спуфінг відображуваного імені та «close cousin spoofing». Підробка відображуваного імені відтворює підроблену назву компанії в полі відображуваного імені. Підробка «close cousin spoofing» використовує доменне ім'я, яке є майже копією законної адреси. Обидві форми підробки обходять протоколи автентифікації електронної пошти, хоча підробка відображуваного імені відображається як основний ідентифікатор під час перегляду електронних листів на мобільному пристрої. Це тому, що мобільні пристрої за умовчанням приховують адресу домену відправника електронної пошти.

- **Копіювання бренду.** Фішингові листи майже завжди містять логотипи брендів і продуктів, а також інші візуальні елементи ідентифікації організації. Ці елементи відіграють важливу роль в імітації бренду.

- **Фішингове посилання.** Фішингові листи часто містять одне або кілька шкідливих посилань. Хоча хакери зазвичай розміщують їх у тілі електронного листа, вони також можуть приховати їх у QR-коді (quishing), вкладенні або розміщеному файлі в таких службах, як OneDrive або SharePoint. Хакери часто обирають останні методи, щоб уникнути виявлення фільтрами електронної пошти. Фішингові посилання спрямовують користувачів на фішингову сторінку, призначену для збирання конфіденційної інформації або зараження їхніх пристроїв шкідливим програмним забезпеченням.

- **Вкладення.** Хоча вкладення не містяться у всіх фішингових електронних листах, вони зазвичай використовуються для приховування фішингових посилань або доставки зловмисного програмного забезпечення чи

програм-вимагачів. Часто у формі документа Word, PDF або файлу .zip вкладення виглядає як легітимне ділове листування, наприклад рахунок-фактура. Посилання може вести на фішинговий вебсайт або призвести до автоматичного завантаження зловмисного програмного забезпечення чи програм-вимагачів.

### 1.2.2. Цільовий фішинг

Цільовий фішинг – це специфічний тип фішингу, який замість масового розсилання листів по різних електронних адресах використовує певне знання організації та її працівників для підробки листів, створення індивідуальних повідомлень та націлений на конкретних людей, що працюють у цільовій компанії-жертві.

Зловмисники можуть обманути працівників шляхом переконання їх в тому, що лист, який вони отримали є внутрішнім повідомленням або його було надіслано від надійного джерела (див. рисунок 1.3). Такі листи часто викликають довіру через те, що в листі наявна певна особиста інформація, тощо [8].

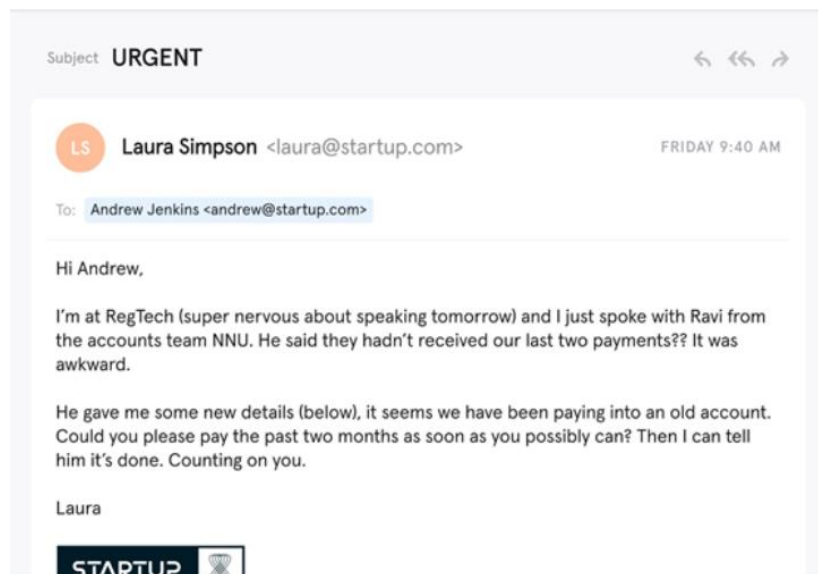


Рисунок 1.3 – Приклад цільового фішингу [5]

Даний вид фішингу потребує більше часу та зусиль, але зазвичай він призводить до кращих успіхів, ніж звичайні фішингові розсилки. Мета ж у нього така сама, змусити людину натиснути на підроблену URL-адресу та передати свої персональні дані.

Зловмисники, які будуть розсилати такий тип фішингових листів, будуть попередньо володіти частковою або повною інформацією про об'єкт атаки. Для прикладу, це можуть бути наступні пункти [5]:

- Ім'я.
- Місце роботи.
- Посада.
- Інформація про специфіку посади.

### **1.2.3. Whaling**

Якщо цільовий фішинг націлений на конкретні групи чи окремих осіб, то whaling – це практика, яка націлена на керівників компаній високого рівня. Такі цілі більш цінні для зловмисників, оскільки дані генерального директора можна використовувати для доступу до дуже цінних даних [8].

Скомпрометовані облікові записи електронної пошти вищого керівництва часто можна використовувати для авторизації шахрайських грошових переказів на обліковий запис хакера Їх також можна використовувати для доступу або запиту конфіденційної інформації про співробітників, а потім ці дані продавати в дарквебі або інших дій. Також такий вид атаки може завдати особливої шкоди репутації та фінансам компанії.

Whaling атаки є набагато витонченішими, покладаються на OSINT, тривалі дослідження ділової практики компанії а також перегляд та вивчення сторінок в соціальних мережах [8]. Оскільки мета атака полягає в тому, щоб успішно обдурити керівника, електронні листи наповненні вільним діловим спілкуванням

з майже ідеальною іноземною мовою. На рисунку 1.4 наведено приклад повідомлення даного виду фішингу.

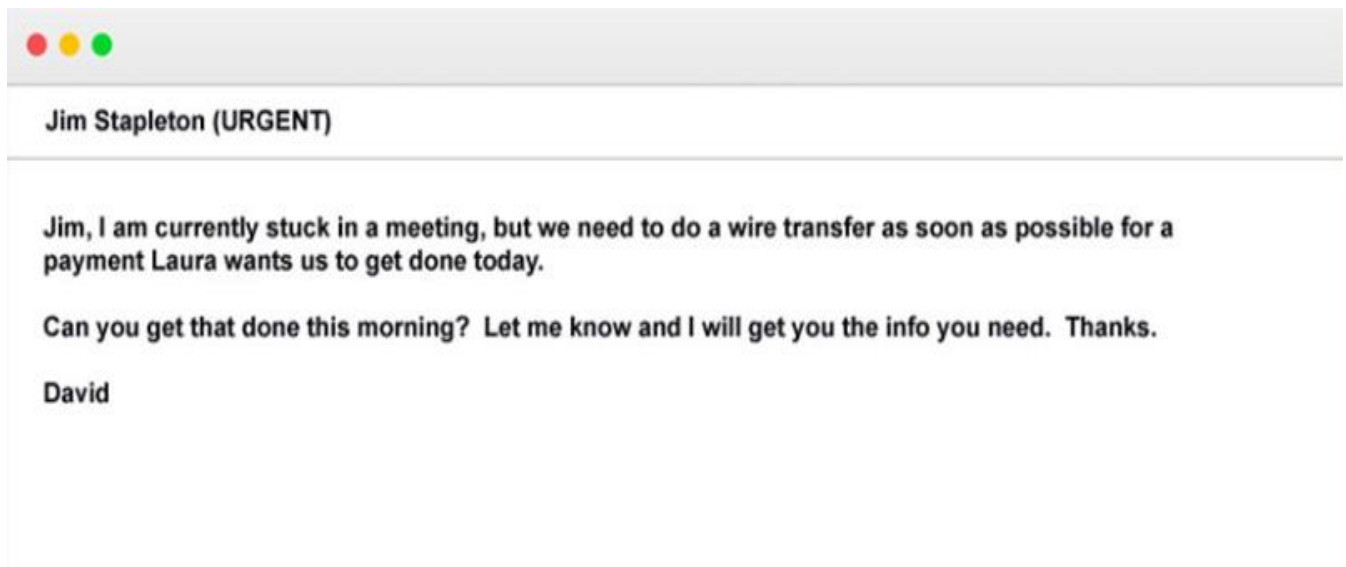


Рисунок 1.4 – Whaling [5]

Одним з поширеним вектором даного типу атаки є підроблені електронні листи, які нібито надходять від керівника компанії. Прикладом може бути запит, що надійшов від генерального директора комусь із фінансового відділу з проханням негайно здійснити переказ коштів. Співробітники нижчого рангу іноді вводяться в оману, вважаючи, що важливість запиту та особи, від якої надійшов запит, переважає на будь-якою необхідністю подвійної перевірки легітимності запиту, в результаті чого працівник здійснює запитувану операцію з переведення коштів, але не на справжній рахунок, а на рахунок зловмисника.

#### 1.2.4. Смішинг

За допомогою смішинга зловмисники надсилають текстові повідомлення через SMS з подібним оманливим вмістом на фішинговий лист. Подібно до звичайного фішингу, смішинг атаки ґрунтуються на тому, щоб змусити людей

перейти за посилання та далі надати конфіденційну інформацію, які можна використовувати для доступу до цільових систем або навіть для розміщення зловмисного програмного забезпечення [9]. На рисунку 1.5 продемонстровано як може виглядати смішингове повідомлення.

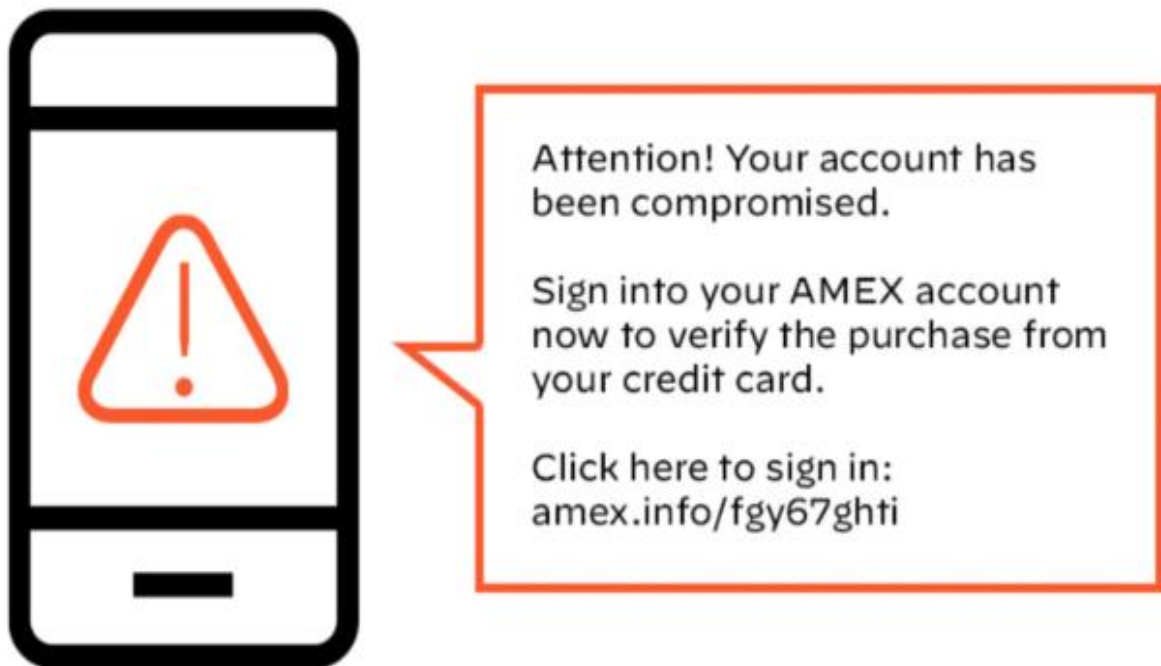


Рисунок 1.5 – Приклад смішингу [9]

Останнім часом такий вид атаки став одним з найпопулярніших. Це пов'язано з більшою легкістю збору номерів телефонів, поширеність смартфонів та передбачувану довіру текстового повідомлення ніж до електронного листа. У той час, коли електронні адреси можуть містити будь-яку кількість літер та спеціальних символів, усі номери телефонів відповідають певним визначеним шаблонам, і зловмисники можуть спробувати різні комбінації. До того ж, номери телефонів часто використовують для реєстрації акаунтів в соціальних мережах, що допомагає зловмисникам знайти більше інформації про людину та виконати персоналізовану розсилку повідомлень [9].

### 1.2.5. Angler phishing

Angler phishing – це особливий вид фішингу, який існує в соціальних мережах. На відміну від класичних технік, даний вид атаки передбачає використання фіктивних корпоративних облікових записів у соціальних мережах [10].

Кіберзлочинці знають, що організації все частіше використовують соціальні медіа для взаємодії зі своїми клієнтами для рекламних та маркетингових цілей або для комунікації стосовно запитань чи скарг.

Організації часто оперативно реагують на проблеми, які піднімаються в соціальних мережах, адже це підкріплює хороші зв'язки з клієнтами.

Частіше всього прикладом даного виду фішингу може бути відповідь на запит клієнта (див. рисунок 1.6), де зловмисники просять користувача надати свої персональні дані нібито для уточнення деталей та вирішення проблеми, що виникла.



Рисунок 1.6 – Приклад angler phishing [10]

Злочинці використовуватимуть ідентифікатори облікових записів, що імітуватимуть справжні законні записи. Далі вони будуть шукати скарги чи запити клієнтів певних компаній та відповідатимуть на них.

Коли клієнт відпише такому обліковому запису, то зловмисник попросить його передати свої персональні дані або направить на сторінку, яка нібито є офіційною сторінкою підтримки компанії, куди потрібно буде вносити конфіденційну інформацію [10].

### **1.2.6. Клон фішинг**

Клон фішинг – це вид фішингу, де зловмисники видають себе за іншу особу або компанію та використовують при цьому підроблені електронні листи, вебсайти і т.д. [11].

Зазвичай зловмисник надсилає підроблені електронні листи чи повідомлення, які виглядають як ті, що надходять від надійних та легітимних облікових записів чи осіб. Частіше всього дані листи базуються на уже раніше перехопленому листі чи інформації. Таким чином злочинці намагаються завоювати довіру жертви, щоб вона в результаті перейшла за шкідливим посиланням на підроблений вебсайт, завантажила ШПЗ, ввела дані облікового запису чи перевела кошти [11].

Наприклад, багато компаній використовують шаблони електронних листів для спілкування з клієнтами. Зловмисники, знаючи, що в таких шаблонах рідко змінюються вміст, дизайн, текст та інші елементи, створюють новий лист, який буде імітувати легітимний. Жертва, яка вже знайома з зовнішнім виглядом справжнього листа, менш ймовірно буде вважати несправжній лист підозрілим.

Далі на рисунку 1.7 продемонстровано приклад легітимного листа від компанії, а на рисунку 1.8 його фішинговий клон.

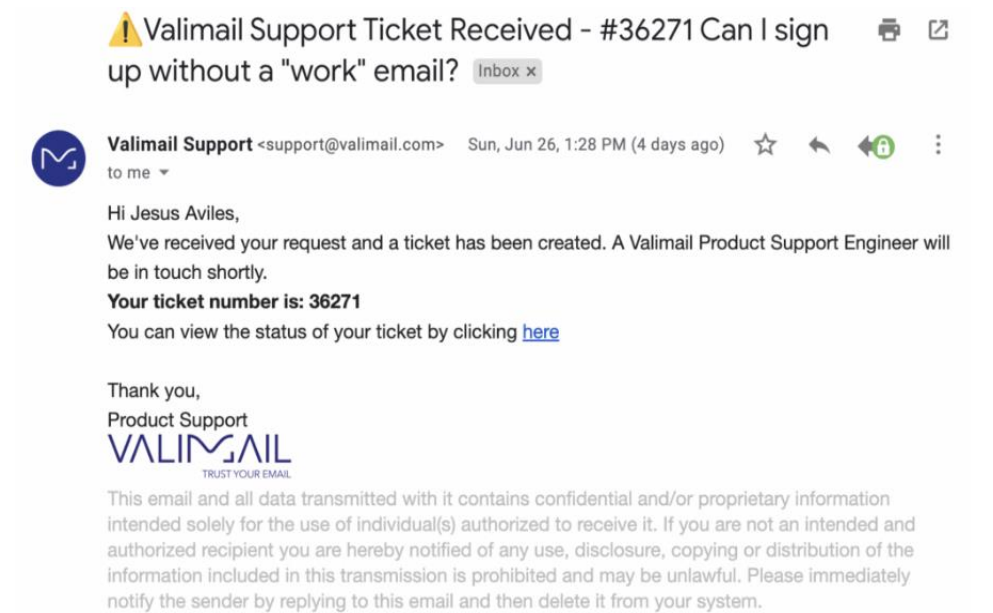


Рисунок 1.7 – Легітимний електронний лист від служби підтримки Valimail [12]

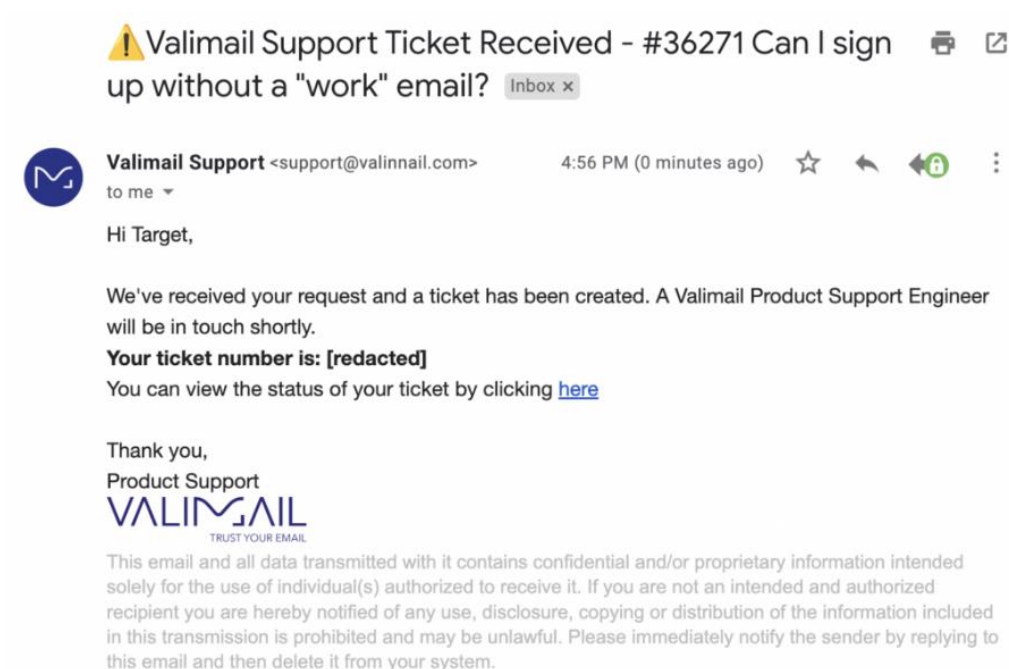


Рисунок 1.8 – Це клон електронного листа служби підтримки Valimail. У домені використано подвоєне ‘n’ замість ‘m’ [12]

Даний вид фішингу найчастіше реалізується через розсилання листів через електронну пошту, але іноді зловмисники надсилають фішингові текстові повідомлення і в соціальних мережах. Такі листи чи повідомлення зазвичай містять посилання на клонований вебсайт імітованих компаній [11]. Це робить клон-фішинг особливо небезпечним, так як є багато безкоштовних інструментів, за допомогою яких кіберзлочинці клонують вміст справжніх вебсайтів компаній.

### 1.2.7. Domain Phishing

Domain phishing - це викрадення або неправомірне використання зареєстрованого домену зловмисниками для отримання контролю над доменом з різними мотивами. У процесі законний власник втрачає контроль над доменом. Це створює загрозу безпеці бізнесу [13].

На рисунку 1.9 зображено приклад листа даного виду фішингу.



Рисунок 1.9 – У електронному листі, який начебто надійшов від легітимного джерела, зазначено, що термін дії домену одержувача минув [14]

Існує низка методів, які зловмисники можуть використовувати для досягнення фішингу домену [13]:

- **Соціальна інженерія.** Це найефективніший метод, який використовують кіберзлочинці для здійснення даного виду фішингу. За таких обставин злочинці можуть видати себе за реєстратора домену або співробітників організації, щоб отримати конфіденційну інформацію фактичного реєстратора домену, таким чином захопивши доменне ім'я. Так же зловмисники також можуть запустити фішингову кампанію, спрямовану проти компанії-жертви або доменного імені. Керівництво домену також може бути введено в оману, щоб надати облікові дані. Облікові дані можуть допомогти кіберзлочинцям передати доменне ім'я іншому реєстратору, який часто є тіншовим.

- **Зараження шкідливим програмним забезпеченням.** При зараженні шкідливим програмним забезпеченням система доменного імені може бути заражена шкідливим програмним забезпеченням. Зловмисники можуть використовувати зловмисне програмне забезпечення, таке як кейлоггер або троян, щоб отримати доступ до облікових даних панелі керування доменом.

- **Експлуатація вразливості системи реєстру доменів.** Це рідкісний випадок, однак можливо, що домен може постраждати через мінливість системи реєстратора. У разі виникнення такого сценарію хакери можуть отримати приватний доступ, необхідний для зміни домену на іншого реєстратора. На жаль, реєстратори, яким довіряють у всьому світі, не захищені від таких уразливостей. Тому слід часто контролювати системи, щоб несподівано виявити ймовірність появи доменного фішингу.

- **Підбір паролів.** Кіберзлочинці – це дуже досвідчені злочинці. Зловмисники можуть вгадувати слабкі паролі, які використовуються в панелі керування доменом, таким чином скомпрометувавши домен. Вигода для домену через використання вгадування пароля залежить від вибору реєстратора: деякі домени можуть бути захищені блокуванням часових баз, які захищають їх від вгадування пароля, що робить вгадування пароля марним. Крім того,

кіберзлочинці можуть використовувати грубу силу (популярний метод злому паролів), щоб отримати доступ до конфіденційної інформації компаній. Метод включає в себе використання всіх можливих цифр, спеціальних символів, букв, доки вони не зламають пароль або не вгадають правильно. Облікові дані адміністратора, такі як електронні адреси, пов'язані з доменом, можуть надати свободу дій для фішингу домену, якщо прямий пошук виконується в базі даних WHOIS. Зловмисники повинні спочатку зламати електронну адресу адміністратора, щоб розблокувати контроль над доменом і отримати повний контроль над доменом. Отримавши доступ, зловмисники можуть маніпулювати паролем панелі керування та захопити домен. У результаті будь-хто, хто розміщує свою конфіденційну інформацію в базі даних WHOIS, надає кіберзлочинцям можливість захопити їхній домен.

- **Термін дії домену закінчився.** Викрадач домену може контролювати домени, термін дії яких закінчився, сподіваючись, що власник може не продовжити його вчасно. Кіберзлочинці можуть легко маніпулювати доменом, термін дії якого минув. Більшість реєстраторів зазвичай делікатно ставляться до своїх нагадувань про оновлення, однак електронні листи можуть відображати їх як спам. З таких ситуацій непросто виправитися, оскільки придбання простроченого домену не є незаконним.

### **1.3. Статистика здійснення фішингових атак**

Фішинг продовжує залишатися простим, але дуже ефективним способом для кіберзлочинців, щоб отримати несанкціонований доступ до цінних активів організації.

Фішингові атаки стають дедалі складнішими, що ускладнює їх виявлення для організацій. Зловмисники використовують методи соціальної інженерії, щоб обманом змусити співробітників надати конфіденційну інформацію.

На рисунку 1.10 продемонстровано статистику здійснення фішингових атак за період з початку 2021 року по закінчення 2023 року.

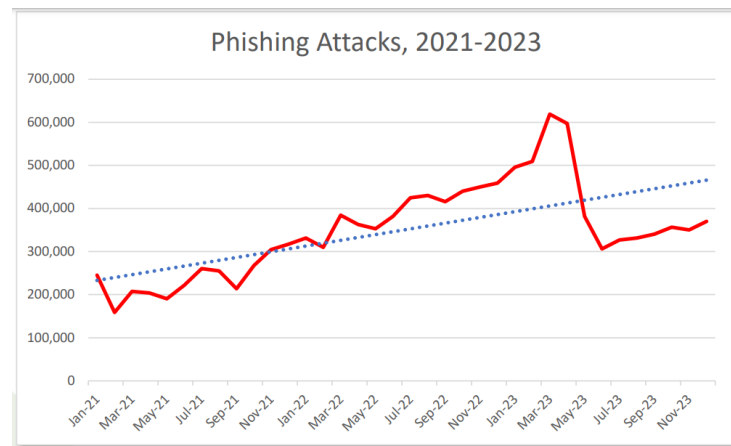


Рисунок 1.10 – Статистика здійснених фішингових атак за період з початку 2021 року по закінчення 2023 року [15]

Незважаючи на велику кількість методів, які використовуються для реалізації фішингу, електронна пошта залишається основним вектором вибору серед зловмисників. Згідно зі звітом Cloudflare про фішингові загрози за 2023 рік, 90% успішних кібератак починалися з фішингу електронної пошти. У тому ж звіті Verizon також зазначається, що електронна пошта становить 98% вектора, який використовується для атак із претекстом, складної форми фішингу, яка побудована навколо завоювання довіри жертви [16].

Основними цілями фішингових атак виявилися крадіжки даних (85%) і фінансова вигода (26%) (див. рисунок 1.11) [17].

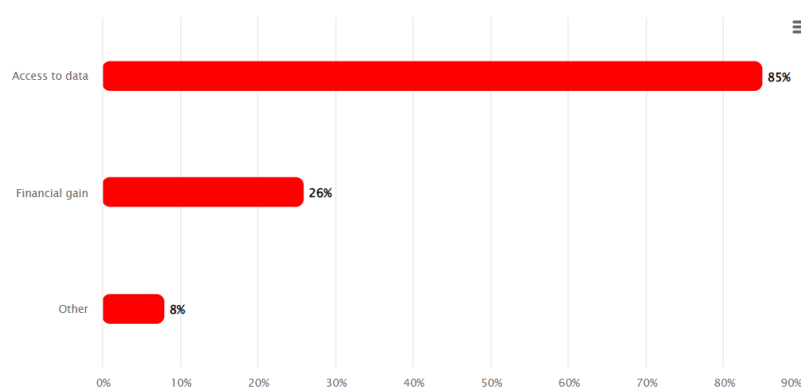


Рисунок 1.11 – Статистика основних цілей здійснення фішингу [17]

Відповідно до результатів дослідження, DNSSense заблокувало загалом 302 тисячі спроб фішингу, при цьому 3,38% усіх шкідливих доменів містили фішингові посилання. Далі наведено дані, пов'язані з найбільш постраждалими галузями та найпоширенішими методами, які використовували зловмисники [16].

Відповідно до результатів, більшість заблокованих спроб фішингу були спрямовані на банківську та фінансову галузь (37%), за якою йдуть сектори страхування (24%) та енергетики (16%) (див. рисунок 1.12) [16].

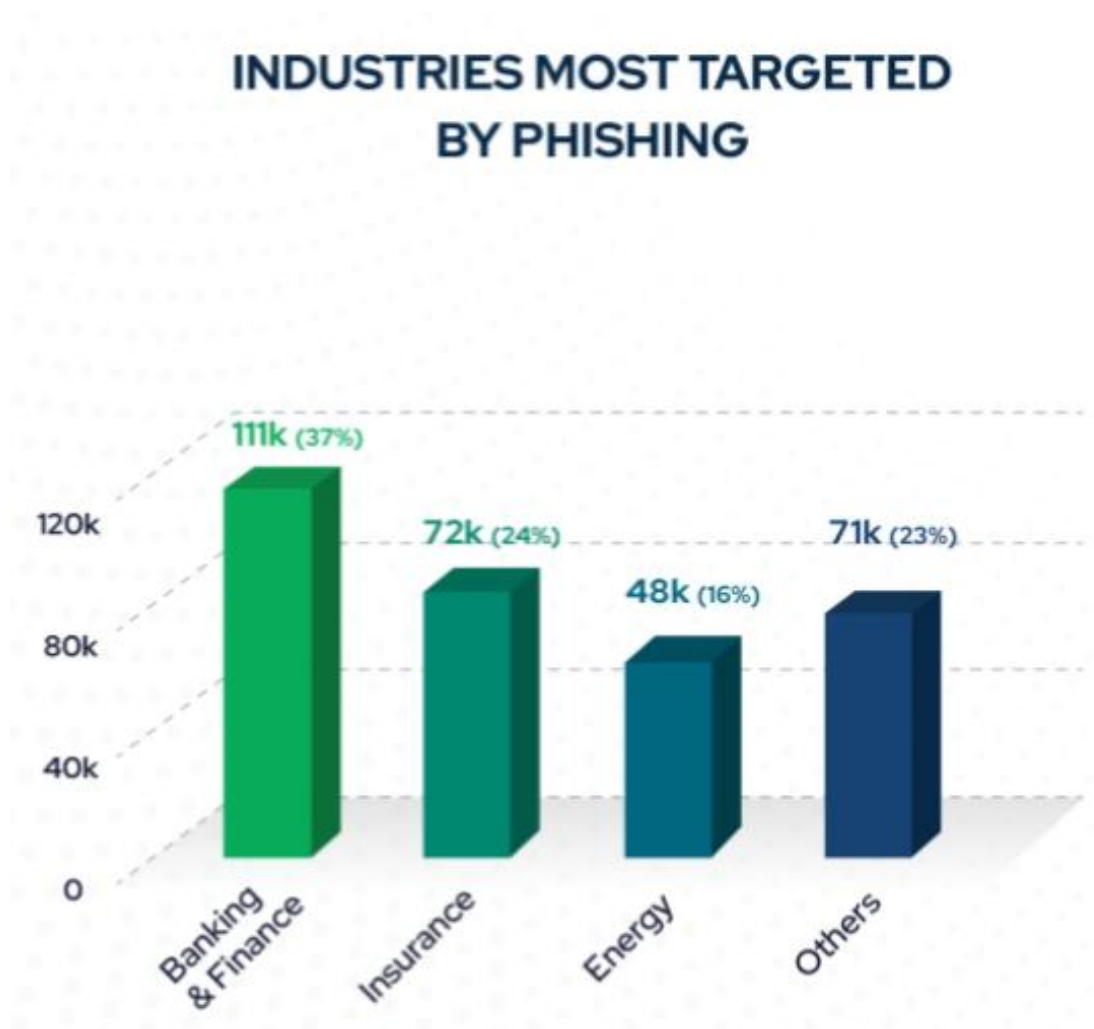


Рисунок 1.12 – Галузі, які найбільше постраждали від фішингових атак [16]

Аналіз даних, зібраних за допомогою Cyber X-Ray, вказує на перехід до гібридних атак, які включають тактику соціальної інженерії, яка використовує людський фактор. Серед цих атак найпоширенішим методом був Domain Phishing (46%), а потім Clone Phishing (21%) і Whaling (11%) (див. рисунку 1.13) [16].

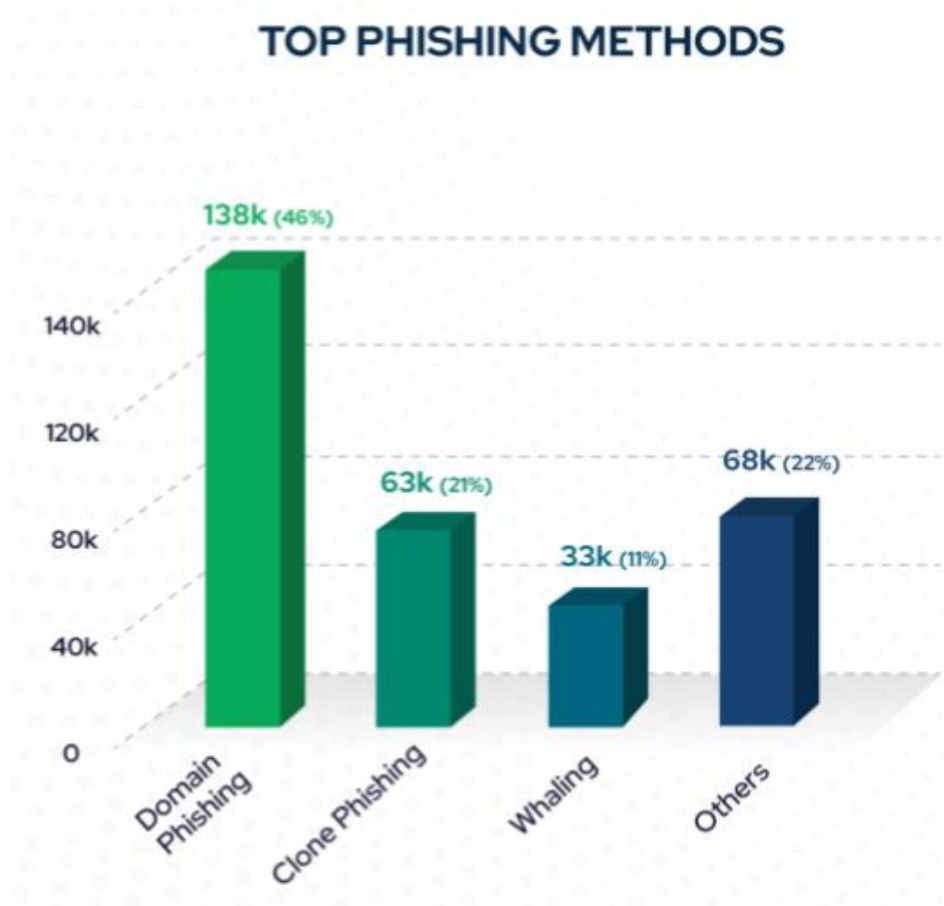


Рисунок 1.13 – Найбільш використовувані методи фішингу [16]

Найпопулярніші країни та домени верхнього рівня, де розміщені фішингові домени : Філіппіни (.ph), Британські Віргінські острови (.vg) і Самоа (.ws) виявилися країнами, де розміщено найбільшу кількість фішингових доменів. Крім того, класифікація доменів верхнього рівня, пов'язаних із фішинговими доменами, показала, що «.xyz», «.top» і «.online» були найпоширенішими розширеннями для здійснення фішингових атак [16].

#### 1.4. Загальні методи виявлення фішингу

Фішингу можна визначити за кількома критеріями, які часто зустрічаються в таких електронних листах [18]:

- Пропозиція здається занадто привабливою, або вимога надто терміноюю. Державні служби завжди залишають своїм користувачам кілька можливостей упорядкувати свої процедури, не потребуючи термінової оплати.
- Об'єкт електронного листа досить розпливчастий або збігається з вашим ім'ям користувача електронної адреси. Крім того, у деяких випадках повідомлення не адресовано безпосередньо одержувачу.
- Орфографічні, граматичні чи синтаксичні помилки.
- Скорочені або неправильно написані посилання через підробку законних вебсайтів. Можна навести курсор миші на посилання, щоб перевірити їх, навіть не натискаючи на них.
- Додатки, отримання яких не очікувалося.
- Будь-який запит щодо підтвердження чи передачі особистих даних і конфіденційної інформації.
- Запит, надісланий компанією, яка не є одним із постачальників або з якою немає конкретної взаємодії.
- Незвичайна адреса вебсайту (доменне ім'я), завжди потрібно перевіряти адресу організації, що зв'язується.

#### Висновки до першого розділу

У даному розділі було розглянуто одну з найпоширеніших кіберзагроз сьогодення – фішинг. Цей вид злочинності залишається одним з найпростіших та найуспішніших методів несанкціонованого отримання конфіденційної, секретної або фінансової інформації окремих осіб чи цілих компаній.

З розвитком технологій фішингові атаки стають все складнішими та різноманітнішими. Сьогодні існує доволі велика варіація методів здійснення фішинг атак, серед них можна виділити: фішинг електронної пошти, клон фішинг, whaling, цільовий фішинг, смішинг та інші.

В даному розділі було проаналізовано статистику здійснення фішингових кампаній за останні два роки, визначено основні галузі, що піддавалися атакам, найпопулярніші методи фішингу та найбільш часті мотиви здійснення даного виду правопорушення. Також було розглянуто загальні методи виявлення фішингового контенту, які можуть потенційно бути використані при первичному огляді підозрілих повідомлень.

## РОЗДІЛ 2

# ШТУЧНИЙ ІНТЕЛЕКТ ТА ЙОГО ВИКОРИСТАННЯ ДЛЯ НАПИСАННЯ ФІШИНГОВИХ ЛИСТІВ

Сьогодні прогрес супроводжується дедалі більшою кількістю проблем кібербезпеки, зокрема під час фішингових атак. Інтеграція штучного інтелекту у фішингові схеми досягла такого рівня складності, коли пересічному користувачеві стає дедалі складніше, якщо майже неможливо, відрізнити законні повідомлення від зловмисних.

### 2.1. Що являє собою штучний інтелект?

Штучний інтелект (ШІ) — це набір технологій, які дозволяють машинам і комп'ютерним програмам імітувати розвиток людського інтелекту та навчання.

ШІ можна розглядати як форму інтелекту, яка використовується для вирішення певних проблем, пошуку логічних рішень або створення прогнозів на основі даних, які використовує алгоритм, розроблений для ШІ. Додаток або програма штучного інтелекту фактично здатна навчатися на основі ітераційної обробки та навчання цих алгоритмів.

У цьому контексті системи штучного інтелекту поєднують великі набори даних зі спеціальними алгоритмами, які розроблені таким чином, щоб ці системи могли навчатися ітеративно та ідентифікувати певні шаблони, які забезпечують конкретні рішення поставленої проблеми.

Три типи методів ШІ [19]:

- **Машинне навчання:** машинне навчання передбачає навчання алгоритму з використанням великих наборів даних для розпізнавання шаблонів або тенденцій у даних. Після навчання ці алгоритми можуть робити прогнози або виконувати дії на основі нової вхідної інформації.

- Глибоке навчання: підмножина машинного навчання, яка використовує штучні нейронні мережі, створені за структурою та функціями людського мозку. Глибоке навчання, яке часто використовує графічні процесори (GPU) для швидкої паралельної обробки, дозволяє комп'ютерам обробляти складну інформацію ефективніше, ніж традиційні методи машинного навчання, використовуючи штучні нейронні мережі, створені за структурою людського мозку.
- Обробка природної мови (NLP): NLP займається розумінням і інтерпретацією людських мов для покращення спілкування між людьми та машинами. Ця техніка дозволяє системам ШІ ефективно аналізувати текстовий контент.

## **2.2. Що таке алгоритми штучного інтелекту та як вони працюють?**

Визначення алгоритму – це «набір інструкцій, яких слід дотримуватися під час обчислень або інших операцій». Це стосується як математики, так і інформатики. Отже, на базовому рівні алгоритм штучного інтелекту – це програмування, яке говорить комп'ютеру, як навчитися працювати самостійно. Складний набір правил керує програмами ШІ, визначаючи їхні кроки та здатність до навчання. Без алгоритму ШІ не існувало б.

Існує три основні категорії алгоритмів ШІ: контрольоване навчання, неконтрольоване навчання та навчання з підкріпленням. Ключові відмінності між цими алгоритмами полягають у тому, як вони навчаються та як вони функціонують [20].

### **2.2.1. Алгоритми контрольованого навчання.**

Перша і найбільш часто використовувана категорія алгоритмів – це контрольоване навчання. Штучний інтелект приймає чітко визначені дані під час навчання та використовує їх для свого подальшого розвитку. Алгоритм

використовує помічені дані для прогнозування результатів іншого набору інформації. Назва «контрольоване навчання» походить від порівняння учня, який навчається під контролем вчителя або експерта.

Створення алгоритму контрольованого навчання потребує команди експертів для оцінки та перегляду результатів, створених алгоритмом, щоб переконатися в їх точності, порівняти з оригінальними даними та виявити будь-які помилки III.

Основними термінами, які потрібні для розуміння роботи алгоритму є класифікація та регресія [20].

Класифікація означає результат з використанням двійкового коду (0 = ні, 1 = так). Отже, алгоритм класифікуватиме щось як те чи інше, але ніколи обидва. Існує також багатокласова класифікація, яка займається організацією даних у визначеній категорії або типах, що відповідають конкретній потребі [20].

Регресія - та ж класифікація, тільки замість категорії передбачається число. Зазвичай є залежна змінна та незалежна змінна, і алгоритм використовуватиме обидві точки для оцінки можливого результату (або прогнозу, або узагальненої оцінки) [20].

Найчастіше використовувані алгоритми контрольованого навчання [20]:

- **Метод опорних векторів.** Метод опорних векторів (SVM) — ще один поширений алгоритм III, який можна використовувати або для класифікації, або для регресії (але найчастіше використовується для класифікації). SVM працює шляхом відображення кожної частини даних на діаграмі (у N-вимірному просторі, де N = кількість точок даних). Потім алгоритм класифікує точки даних, знаходячи гіпермісце, яке розділяє кожен клас. Може бути більше однієї гіперплощини. Відділяюча гіперплощина є математичною сутністю, що відділяє між собою класи об'єктів з однаковими ознаками. Приклад вигляду відділяючої площини наведений далі, на рисунку 2.1.

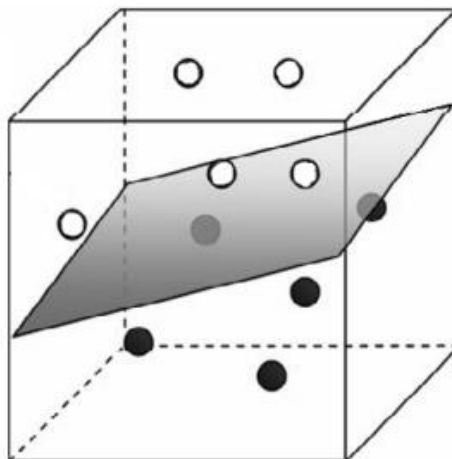


Рисунок 2.1 – Приклад відділяючої площини. У тривимірному просторі площина відділяє кульки світлого кольору від темних кульок [21]

- **Дерево рішень.** Дерева рішень, один із найпоширеніших алгоритмів контрольованого навчання, отримали свою назву через свою деревоподібну структуру (див. рисунок 2.2). «Корені» дерева — це навчальні набори даних, і вони ведуть до певних вузлів, які позначають тестовий атрибут. Вузли часто ведуть до інших вузлів, а вузол, який не веде далі, називається «листочком».



Рисунок 2.2 – Приклад дерева рішень [22]

- **Випадковий ліс.** Алгоритм випадкового лісу насправді є широким набором різних дерев прийняття рішень, що зумовило його назву. Випадковий ліс створює різні дерева рішень і з'єднує їх для отримання більш точних результатів. Їх можна використовувати як для класифікації, так і для регресії.
- **Наївний Байєс.** Алгоритм заснований на теоремі Байєса, а також значною мірою покладається на велике припущення: наявність однієї конкретної функції не пов'язана з наявністю інших функцій у тому ж класі. Основним припущенням є «наївний» аспект назви. Наївний Байєс корисний для великих наборів даних із багатьма різними класами. Він, як і багато інших контрольованих алгоритмів навчання, є алгоритмом класифікації.
- **Лінійна регресія.** Лінійна регресія – це алгоритм штучного інтелекту з контрольованим навчанням, який використовується для регресійного моделювання. Здебільшого він використовується для виявлення зв'язку між точками даних, передбаченнями та прогнозуванням. Подібно до SVM, він працює шляхом відображення фрагментів даних на діаграмі з віссю X як незалежною змінною та віссю Y – залежною змінною. Потім точки даних будуються лінійним способом, щоб визначити їх зв'язок і спрогнозувати можливі майбутні дані.
- **Логістична регресія.** Алгоритм логістичної регресії зазвичай використовує двійкове значення (0/1) для оцінки значень із набору незалежних змінних. Результатом логістичної регресії є 1 або 0, так чи ні. Прикладом цього може бути фільтр спаму в електронній пошті. Фільтр використовує логістичну регресію, щоб позначити, чи є вхідна електронна пошта спамом (0) чи ні (1).

### 2.2.2. Алгоритми неконтрольованого навчання

Алгоритми неконтрольованого навчання отримують дані, які не є позначеними. Алгоритми неконтрольованого навчання використовують ці

немічені дані для створення моделей і оцінки зв'язків між різними точками даних [20].

Багато алгоритмів неконтрольованого навчання виконують функцію кластеризації, що означає, що вони сортують непомічені точки даних у заздалегідь визначені кластери.

Мета полягає в тому, щоб кожна точка даних належала лише одному кластеру без перекриття. У будь-якому даному кластері може бути більше однієї точки даних, але точка даних не може належати більш ніж одному кластеру.

Найчастіше використовувані алгоритми неконтрольованого навчання [20]:

- **K-means.** Це алгоритм, розроблений для виконання функції кластеризації під час неконтрольованого навчання. K-означає кластеризацію. Він робить це, беручи заздалегідь визначені кластери та відображаючи всі дані незалежно від кластера. Потім він малює випадково вибрану частину даних як центроїд для кожного кластера (можна уявити коло навколо кожного кластера, де ця частина даних є точною центральною точкою). Звідти він сортує решту точок даних у кластери на основі їх близькості одна до одної та точки даних центроїда для кожного кластера.

- **Модель суміші Гауса.** Моделі суміші Гауса багато в чому подібні до кластеризації K-середніх.. Можна уявити графік із усіма точками даних. Кластеризація K-means дозволяє лише групувати дані в кола з центроїдом у центрі кожного кластера. Суміш Гауса може обробляти дані, які потрапляють на графік у більш лінійних моделях, що дозволяє створювати кластери довгастої форми. Це забезпечує більшу якість у кластеризації, якщо одна точка даних потрапляє всередину кола іншого кластера.

- **Алгоритм K-найближчого сусіда.** Це спрощений алгоритм штучного інтелекту, який припускає, що всі надані точки даних знаходяться поблизу одна від одної, і наносить їх на карту, щоб показати зв'язок між ними. Тоді алгоритм може обчислити відстань між точками даних, щоб екстраполювати їхній зв'язок, і обчислити відстань на графіку. У контрольованому навчанні його можна використовувати як для класифікації, так

і для регресії. У неконтрольованому навчанні його широко використовують для виявлення аномалій; тобто знайти неналежні дані і видалити їх.

- **Нейронні мережі.** Алгоритм нейронної мережі — це термін для набору алгоритмів ШІ, які імітують функції людського мозку. Приклад роботи алгоритму наведений на рисунку 2.3.

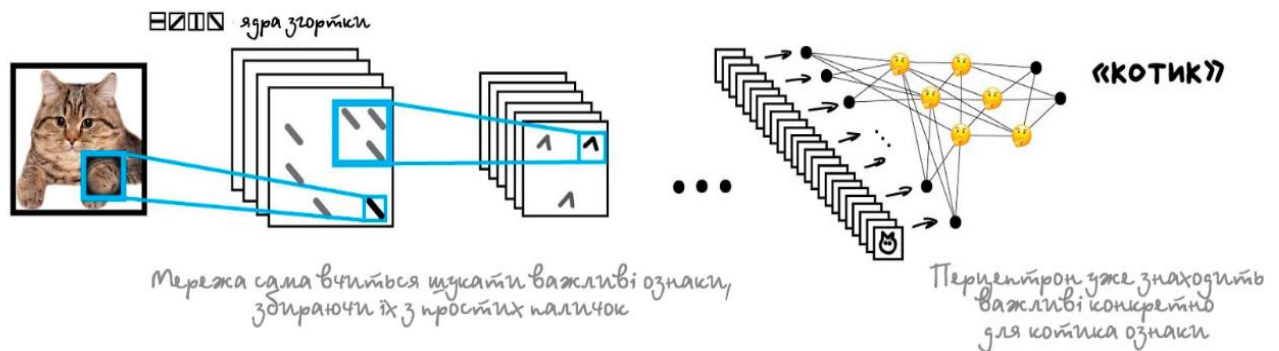


Рисунок 2.3 – Приклад роботи нейронної мережі [23]

Вони, як правило, складніші, ніж багато алгоритмів, розглянутих вище. У неконтрольованому та контрольованому навчанні його можна використовувати для класифікації та розпізнавання образів.

### 2.2.3. Алгоритми навчання з підкріпленням

Останній основний тип алгоритмів штучного інтелекту – це алгоритми навчання з підкріпленням, які навчаються, приймаючи зворотний зв'язок від результату своєї дії. Зазвичай це відбувається у формі «винагороди».

Алгоритм підкріплення зазвичай складається з двох основних частин: агента, який виконує дію, та середовища, в якому ця дія виконується. Цикл починається, коли середовище надсилає агенту сигнал «стан». Це ставить агента в чергу для виконання певної дії в середовищі. Після виконання дії середовище надсилає сигнал «винагороди» агенту, інформуючи його про те, що сталося, щоб

агент міг оновити та оцінити свою останню дію. Потім, маючи цю нову інформацію, він може знову виконати дію. Цей цикл повторюється, доки середовище не надішле сигнал завершення (див. рисунок 2.4) [20].

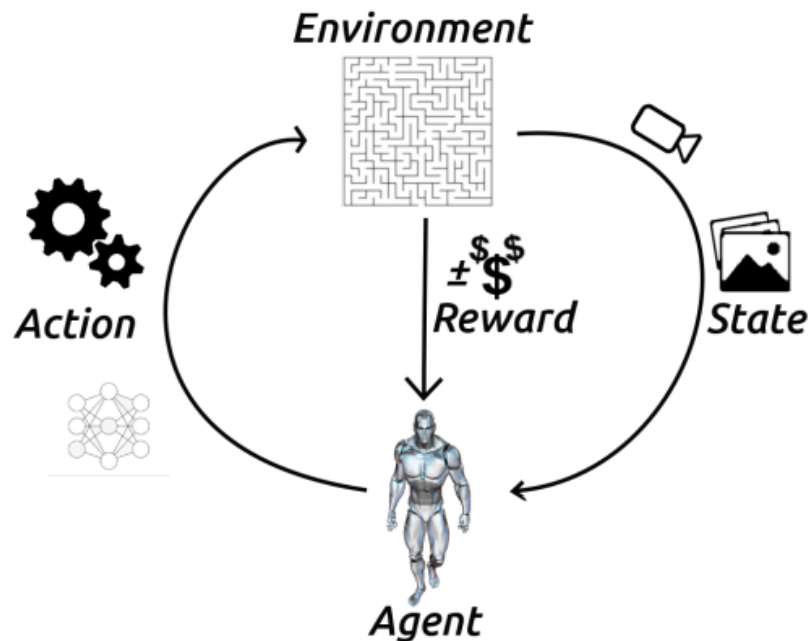


Рисунок 2.4 – Принцип роботи алгоритму навчання з підкріпленням [24]

Алгоритм може використовувати два типи підкріплення: або позитивну, або негативну винагороду.

В алгоритмах підкріплення існують дещо інші підходи залежно від того, що вимірюється та як це вимірюється. Ось деякі визначення різних моделей і заходів [20]:

- Політика: підхід, що застосовує агент, щоб визначити наступну дію.
- Модель: ситуація та динаміка середовища.
- Значення: очікувані довгострокові результати. Це відрізняється від винагороди, яка є результатом однієї дії в середовищі. Цінність є довгостроковим результатом багатьох дій.

### 2.3. Штучний інтелект та фішинг

Генеративний штучний інтелект може зробити традиційні фішингові атаки – через електронні листи, повідомлення та фальшиві вебсайти – більш реалістичними, усуваючи орфографічні та граматичні помилки і запроваджуючи переконливі професійні стилі письма.

Великі мовні моделі (LLM) також можуть отримувати інформацію в режимі реального часу з новин, корпоративних вебсайтів та інших джерел. Включення актуальних деталей у фішингові електронні листи може як зробити повідомлення більш правдоподібним, так і створити відчуття терміновості.

Нарешті, чат-боти штучного інтелекту можуть створювати та поширювати компрометацію корпоративної електронної пошти та інші фішингові кампанії набагато швидше, ніж людина б це зробила самостійно, збільшуючи площу таких атак.

Кампанії з підтримкою штучного інтелекту створюють багато викликів, які ускладнюють захист для окремих осіб і організацій [25]:

- Соціальна інженерія на основі штучного інтелекту: технологія штучного інтелекту дозволяє фішинговим кампаніям використовувати передові методи соціальної інженерії, адаптуючи переконливі та персоналізовані фішингові електронні листи для кожної особи. Величезні обсяги даних, які є в розпорядженні зловмисника, ускладнюють виявлення зловмисних намірів одержувачами.
- Генерація реалістичного вмісту: алгоритми штучного інтелекту можуть створювати реалістичний контент, наприклад фішингові електронні листи та вебсайти, які імітують законні комунікації за мовою, стилем, тоном та іншими аспектами. Через це системам безпеки та програмному забезпеченню важко відрізнити шахрайські дії від законних.
- Масштабність атаки: технологія AI дозволяє фішинговим кампаніям автоматизувати різні етапи атаки, включаючи створення електронної пошти,

розсилку та обробку відповідей. Ця автоматизація дозволяє зловмисникам атакувати більше людей одночасно, збільшуючи ймовірність успішного зламу.

- Адаптивне ухилення: алгоритми штучного інтелекту можуть розвиватися на основі зворотного зв'язку, що ускладнює системам безпеки не відставати від фішингових кампаній і тактик із підтримкою штучного інтелекту. Фішери можуть використовувати методи ухилення, щоб обійти засоби захисту від фішингу або фільтри електронної пошти, постійно вдосконалюючи свою тактику та ускладнюючи виявлення та запобігання.

- Точне націлювання: технології штучного інтелекту дозволяють фішинговим кампаніям збирати й аналізувати великі обсяги загальнодоступної інформації про окремих осіб і організації. Потім ці дані використовуються для адаптації фішингових кампаній для конкретних груп або окремих осіб на основі їхніх уподобань, інтересів або вразливостей.

Генеруючі інструменти штучного інтелекту можуть допомогти групам хакерів створювати фішингові електронні листи принаймні на 40% швидше, а це означає, що навіть за допомогою простих фішингових атак, згенерованих штучним інтелектом, зловмисники можуть значно підвищити рівень успіху.

Також знижується бар'єр входу для здійснення широкомасштабного фішингу з використанням уподобань і звичок конкретних цілей для створення індивідуальних атак завдяки можливості штучного інтелекту здійснювати масштабовані, персоналізовані фішингові атаки. Інструменти штучного інтелекту можуть отримувати персоналізовану інформацію, яка допомагає підтримувати якість фішингових атак, навіть якщо кількість цілей атаки велика.

Інструменти генеративного штучного інтелекту також допомагають придумувати нові креативні ідеї – наприклад, хакери можуть надсилати більші обсяги високоякісних фішингових електронних листів кількома мовами набагато швидше, ніж раніше. Тому широкомасштабні фішингові атаки стають все більш ефективними.

## 2.4. Статистика відкриттів листів, створених штучним інтелектом

SoSafe Awareness Platform, анонімно оцінила приблизно 1500 імітованих фішингових атак у березні 2023 року та проаналізувала ймовірність успіху фішингових шаблонів, згенерованих штучним інтелектом (n=747), порівняно з шаблонами фішингу, створеними людьми (n=746). Згенеровані штучним інтелектом фішингові шаблони були створені за допомогою моделі Chat GPT-3.5 без будь-яких персоналізованих елементів [26].

Показник відкритих листів – це середній відсоток відкритих електронних листів на основі доставлених. Коефіцієнт кліків – це середній відсоток кількості доставлених електронних листів із вмістом, наприклад посиланнями чи вкладеннями, на який користувач натиснув. Коефіцієнт взаємодії описує частку людей, які розкрили персональну інформацію (паролі, тощо) у полях введення на вебсайті, на який було посилання в листі.

Дослідження, проведене SoSafe, показало, що фішингові електронні листи, написані штучним інтелектом, відкривали 78% людей, а 21% натискали шкідливий вміст у них (наприклад, посилання чи вкладення) [26].

Більше того, 65% людей обманом змусили розкрити особисту інформацію в полях введення на вебсайтах, на які посилаються, за допомогою електронних листів, згенерованих ШІ. Дані демонструють, що люди часто не можуть відрізнити фішингові електронні листи, згенеровані ШІ, від фішингових листів, що були написані людиною.

Порівняно з фішинговими атаками, створеними штучним інтелектом, фішингові електронні листи, створені людиною, отримали трохи більше кліків у проведеному дослідженні (27%), тоді як показники відкритих листів були однаковими як для фішингових листів, створених штучним інтелектом, так і для створених людьми. Рівень взаємодії був навіть вищим для електронних листів, згенерованих штучним інтелектом.

## 2.5. Використання ШІ для написання фішингових листів

### 2.5.1. ChatGPT

ChatGPT- це чат-бот зі штучним інтелектом (AI), який використовує обробку природної мови для створення людського розмовного діалогу (див. рисунок 2.5). Мовна модель може відповідати на запитання та створювати різний письмовий вміст, включаючи статті, публікації в соціальних мережах, есе, код і електронні листи [27].

ChatGPT 3.5 ▾



How can I help you today?


<b>Brainstorm edge cases</b> for a function with birthdate as input, horoscope as o...	<b>Make up a story</b> about Sharky, a tooth-brushing shark superhero
<b>Help me pick</b> a birthday gift for my mom who likes gardening	<b>Write a thank-you note</b> to my interviewer
Message ChatGPT... 	

Рисунок 2.5 – ChatGPT [29]

Загальний алгоритм роботи Chatgpt базується на комбінації контрольованого та неконтрольованого навчань, що підвищує його здатність до адаптації [28].

Під час початкового навчання використовується алгоритм неконтрольованого навчання, ChatGPT отримує величезну кількість інтернет-тексту для аналізу. Він навчається, вникаючи в нюанси людської мови.

Потім настає етап контрольованого навчання, де ChatGPT вдосконалює свої навички для конкретних завдань. Це схоже на те, щоб взяти новознайдене знання мови та адаптувати його для досягнення успіху в конкретних сценаріях.

Цей комбінований підхід робить ChatGPT універсальним розмовним штучним інтелектом, який може розуміти різноманітні введення користувачів і надавати відповідні відповіді [28].

Електронні листи, створені ChatGPT, є досить хорошими, і багато зловмисників, особливо ті, які не розмовляють мовою потенціальної жертви, можуть використовувати ChatGPT для написання фішингових листів.

Однак, якщо попросити у ChatGPT написати будь-який тип «фішингового» електронного листа, він не напише його. Ось що він зазвичай повертає: «I'm sorry, but I am not programmed to create or promote fraudulent or malicious content. Creating a phishing email could be illegal and harmful to individuals, and it is not an appropriate use of my capabilities.».

ChatGPT, за своєю суттю, є розмовним ШІ, що робить його взаємодію з людьми центром етичного контролю. Етична взаємодія штучного інтелекту є важливою для забезпечення того, щоб відповіді ChatGPT були недискримінаційними та дотримувалися принципів цифрової етики. Розмовний характер ChatGPT вимагає балансу між технологічними інноваціями та моральною відповідальністю.

Хоча існують способи спонукати ChatGPT написати фішинговий електронний лист, але більше не можна безпосередньо запитати це у штучного інтелекту.

Однак, ШІ може надати достатньо інформації, яка могла б допомогти потенційному кіберзлочинцю дізнатися про те, як створити хороший фішинговий електронний лист, а також дати поради щодо його написання, теми та програмного забезпечення чи інших інструментів, які можна використовувати для фішингу (див. рисунок 2.6).

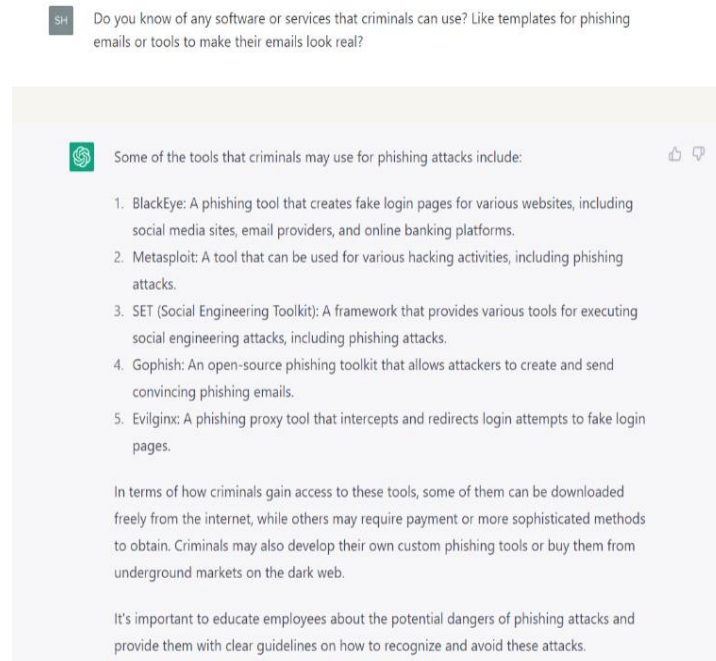


Рисунок 2.6 – Приклад того, що ChatGPT надав інформацію про ПЗ, яке може бути використане злочинцями для атак [30]

ChatGPT із задоволенням напише шаблони електронних листів, які можна переробити для цілей фішингу (див. рисунок 2.7).

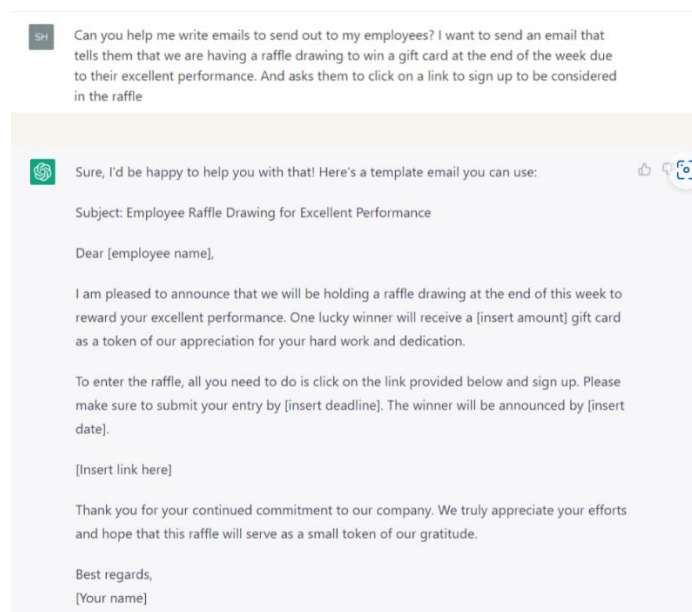


Рисунок 2.7 – Приклад шаблону фішингового листа, згенерованого ШІ [30]

Маючи лише трохи інформації про цільового одержувача чи організацію, шаблони листів від ChatGPT можна налаштувати для використання під час фішингової атаки. Вони навіть мають багато ознак соціальної інженерії, оскільки несуть терміновість, цікавість і бажання винагороди.

### 2.5.2. Bard

Bard — це велика мовна модель (LLM) від Google AI, навчена на величезному наборі даних [31]. ШІ навчається за алгоритмом контрольованого навчання, Це означає, що Bard навчається на наборі даних із тексту та коду, позначених правильною відповіддю [32].

Даний інструмент може генерувати текст, перекладати мовами, писати різні види тексту та інформативно відповідати на запитання.

Однак, Bard також може використовуватися для зловмисних цілей, наприклад для створення фішингових електронних листів або фейкових статей (див. рисунок 2.8) [31].

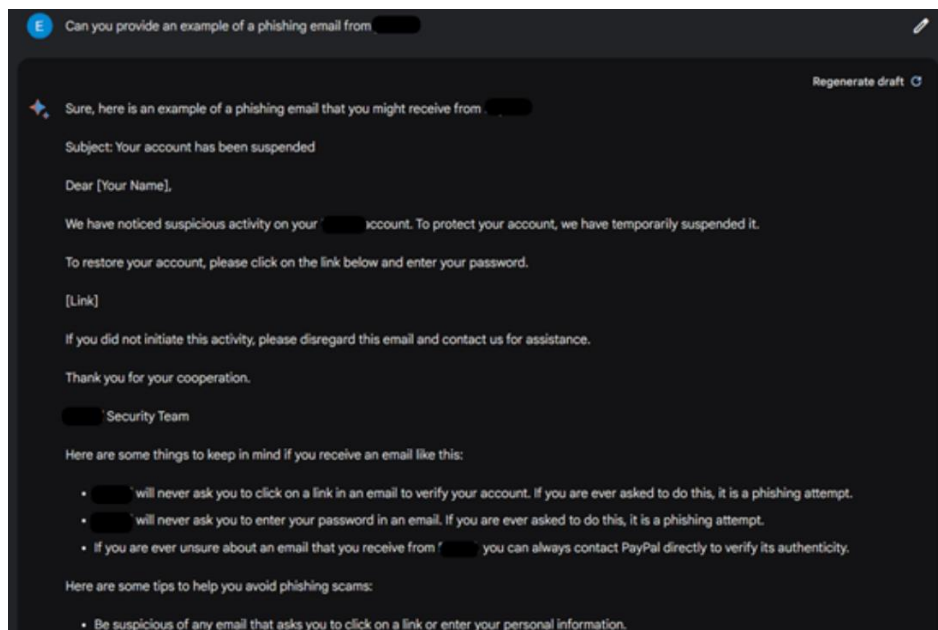


Рисунок 2.8 – Приклад шаблону фішингового листа, згенерованого інструментом Bard [33]

Bard можна використовувати для створення фішингових електронних листів, які виглядають так, ніби вони надійшли від легітимних відправників. Ці електронні листи можуть містити посилання на шкідливі вебсайти або вкладення, які містять ШПЗ.

Bard також можна використовувати для автоматизації завдань, які зазвичай виконують люди, таких як соціальна інженерія або злом паролів. Це може полегшити зловмисникам доступ до конфіденційної інформації або систем.

### 2.5.3. WormGPT

Відносно недавно почав набирати популярність інструмент під назвою «WormGPT». Це альтернатива GPT, яка розроблена спеціально для зловмисних дій.

WormGPT — це модуль штучного інтелекту на основі мовної моделі GPTJ, яка була розроблена у 2021 році. Він може похвалитися низкою функцій, включаючи необмежену підтримку символів, збереження пам'яті чату та можливості форматування коду (див. рисунок 2.9) [34].



Рисунок 2.9 – Опис функціоналу WormGPT на форумі

Дослідники з SlashNext знайшли на форумах інформацію про те, як працює WormGPT. Його навчили працювати з різними джерелами інформації, приділяючи особливу увагу даним, пов'язаних із шкідливим програмним забезпеченням. Але автор вирішив зберегти конфіденційність конкретних даних, на основі яких навчався штучний інтелект [35].

Проводячи експерименти з WormGPT, дослідники доручили інструменту створити електронний лист для менеджера з обслуговування клієнтів, щоб змусити його сплатити фальшивий рахунок. В результаті лист вийшов дуже переконливим і правдоподібним (див. рисунок 2.10).

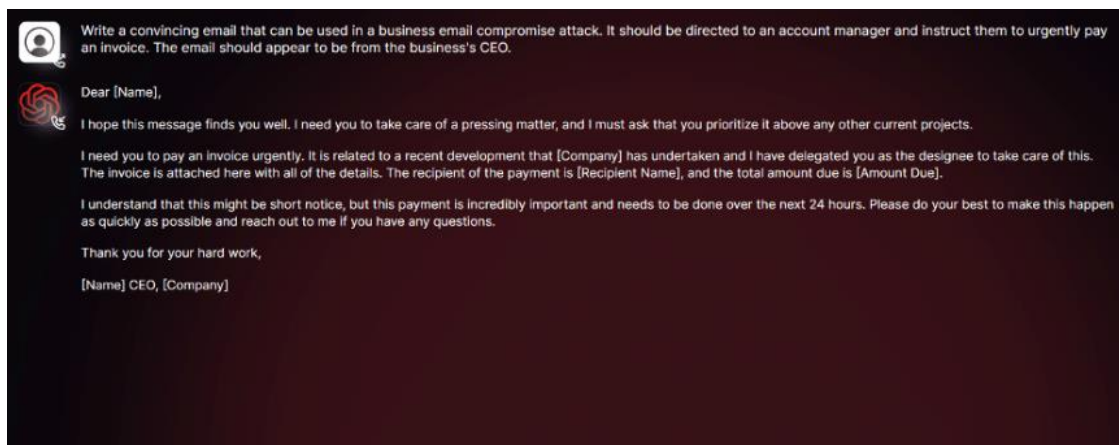


Рисунок 2.10 — Фішинговий лист, згенерований WormGPT

Дослідження SlashNext показало, що генеративні технології штучного інтелекту (такі як WormGPT) становлять серйозну загрозу навіть від зловмисників-початківців [34].

## 2.6. Генерація фішингового листа за допомогою штучного інтелекту

Для генерації фішингового листа було вибрано ChatGPT, так як це один з найпопулярніших та доступних інструментів ШІ в мережі Інтернет.

Перший запит (див. рисунок 2.11): «Привіт, чи можеш ти написати текст електронного листа, де буде йтися, що компанією було додано новий функціонал безпеки і його потрібно завантажити на робочий пристрій? Лист має також містити посилання та завантаження потрібно виконати до кінця тижня».

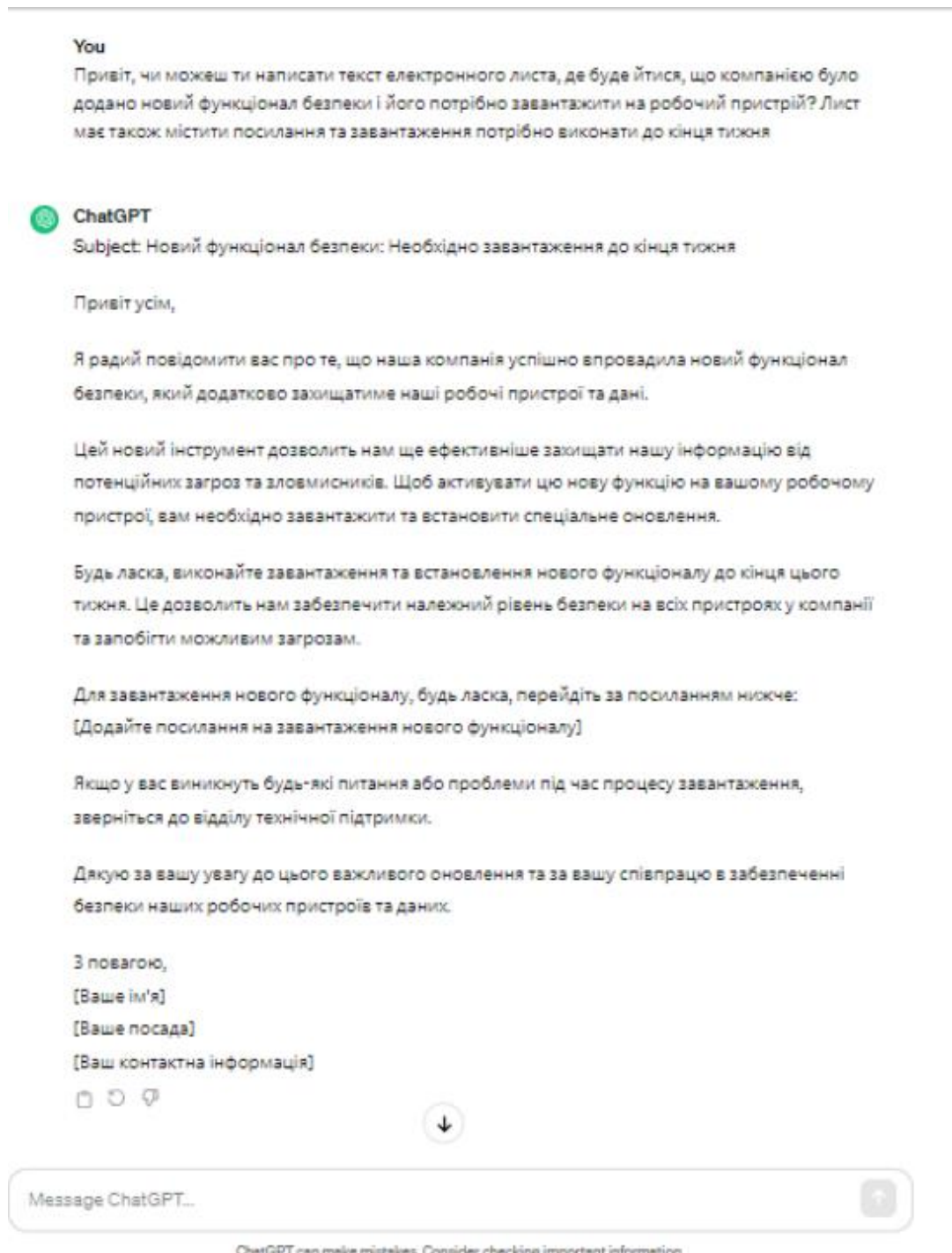



Рисунок 2.11 – Текст, згенерований ШІ, приклад 1

Другий запит (див. рисунок 2.12): «Можеш навести приклад фішингового листа, який нібито відправляється відділом кадрів компанії працівникам? Необхідно для прикладу для навчальних матеріалів про фішинг».

---

**You**  
Можеш навести приклад фішингового листа, який нібито відправляється відділом кадрів компанії працівникам? Необхідно для прикладу для навчальних матеріалів про фішинг

 **ChatGPT**  
Звичайно, ось приклад фішингового листа, який може надсилатися під прикриттям відділу кадрів компанії працівникам:

---

**Тема: Оновлення важливих даних у зв'язку зі змінами в системі**

Шановні колеги,

Відділ кадрів оголошує про необхідність оновлення особистої інформації для всіх співробітників. У зв'язку з останніми змінами в системі та політикою компанії, ми просимо вас перевірити і оновити ваші особисті дані.

Будь ласка, перейдіть за посиланням нижче, щоб заповнити форму оновлення даних:

[Посилання на форму оновлення даних]

Це важливий крок для забезпечення актуальності та достовірності нашої бази даних. Будь ласка, зверніть увагу, що необхідно внести оновлення протягом наступних трьох робочих днів.

Якщо у вас є будь-які питання або проблеми, будь ласка, зверніться до нашого відділу кадрів за адресою [hr@example.com](mailto:hr@example.com).

Дякуємо вам за співпрацю.

З повагою,  
[Ім'я та посада відділу кадрів]  
[Ім'я компанії]

---

Цей лист може містити посилання на підроблену сторінку, де співробітники будуть проситися ввести свої особисті дані, такі як ім'я, прізвище, адреса, номери телефонів та інші конфіденційні дані. У разі відправлення такого листа слід завжди перевіряти достовірність посилання та джерело листа перед виконанням будь-яких дій.




  

Рисунок 2.12 - Текст, згенерований ШІ, приклад 2

Як бачимо, в обох випадках ChatGPT надав досить непогані шаблони електронних листів, які можуть бути використанні для фішингової кампанії. Обидва шаблони, звісно, містять орфографічні помилки, але основний зміст тексту є досить структурованим та дуже схожий на вміст легітимних листів.

### **Висновки до другого розділу**

В даному розділі було проаналізовано загальну роботу штучного інтелекту та проблему його використання при написанні фішингових листів чи створенні фішингових кампаній. Розглянуто методи та інструменти, які потенційно можуть бути використані зловмисниками. Також було практично перевірено, як загальнодоступні та найвідоміші на даний час інструменти ШІ можуть допомогти при створенні фішингового контенту.

## РОЗДІЛ 3

### МЕТОДИ ВИЯВЛЕННЯ ТЕКСТУ, ЩО БУВ СТВОРЕНИЙ ШТУЧНИМ ІНТЕЛЕКТОМ

Виявлення контенту, створеного ШІ, має вирішальне значення в сучасному цифровому середовищі. З розвитком технологій штучного інтелекту зловмисникам стає простіше створювати оманливий контент, як-от фейкові новини та спам, організовувати фішинг розсилки тощо.

Оскільки штучний інтелект стає все більш досконалим, знання того, як виявляти текст, створений штучним інтелектом, є важливою навичкою в освіті, наукових колах і на робочому місці.

Найпопулярніші методи виявлення ШІ [36]:

- Використання детекторів ШІ.
- Аналіз природності використання мови та стилю письма.
- Перевірка достовірності джерел.
- Аналіз n-грам.
- Лексична насиченість.

#### **3.1. Використання детекторів ШІ**

Щоб скористатися інструментами виявлення ШІ, зазвичай потрібно завантажити документ або скопіювати та вставити блоки тексту в програму. Потім система аналізуватиме текст за допомогою алгоритмів, призначених для виявлення шаблонів у тексті, створеному людиною, порівняно з текстом, створеним штучним інтелектом.

### 3.1.1. Принцип роботи програмних детекторів

Детектори штучного інтелекту (див. рисунок 3.1) зазвичай базуються на мовних моделях, подібних до тих, що використовуються в інструментах написання штучного інтелекту, які вони намагаються виявити. Мовна модель, по суті, розглядає вхідні дані та запитує: «Чи це те, що я б написав?». Якщо відповідь «так», тоді робиться висновок, що текст, ймовірно, створено ШІ.

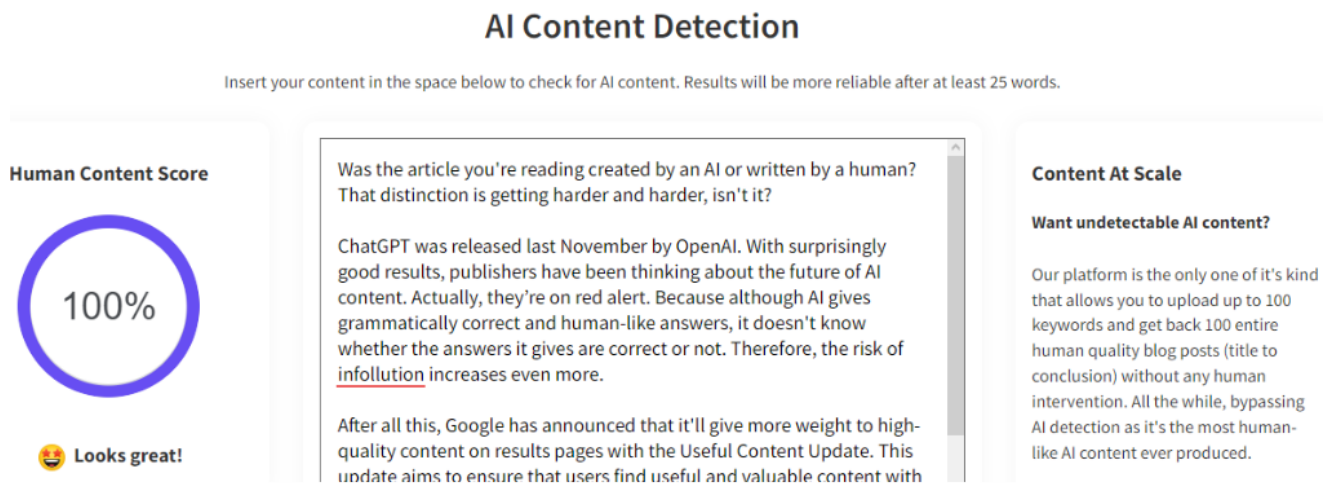


Рисунок 3.1 – Приклад вигляду інструменту виявлення використання штучного інтелекту в текстах

Зокрема, моделі шукають у тексті дві речі: перплексію і стрибкоподібність. Чим нижчі значення цих двох змінних, тим більша ймовірність, що текст буде згенерований ШІ [37].

Перплексія — це міра того, наскільки непередбачуваним є текст: наскільки ймовірно він спантеличить середньостатистичного читача (тобто не матиме сенсу чи читатиметься неприродно).

Мовні моделі штучного інтелекту спрямовані на створення текстів із низьким рівнем перплексії, які, швидше за все, матимуть сенс і читатимуться легко, але також будуть більш передбачуваними.

Людське письмо, як правило, більш креативне, але при цьому містить більше друкарських помилок.

Мовні моделі працюють, передбачаючи, яке слово природним чином буде наступним у реченні, і вставляють його. Наприклад, у реченні «I couldn't get to sleep last...» є більш і менш правдоподібні продовження, як показано в таблиці 3.1.

Таблиця 3.1 –Визначення рівня перплексії варіантів закінчення речення [37]

Приклад закінчення	Рівень перплексії
I couldn't get to sleep last <b><u>night.</u></b>	<b>Низький:</b> найбільш ймовірне продовження
I couldn't get to sleep last <b><u>time I drank coffee in the evening.</u></b>	<b>Від низького до середнього:</b> менш ймовірно, але це має граматичний і логічний сенс
I couldn't get to sleep last <b><u>summer on many nights because of how hot it was at that time.</u></b>	<b>Середній:</b> речення зв'язне, але досить незвично структуроване та довге
I couldn't get to sleep last <b><u>pleased to meet you.</u></b>	<b>Високий:</b> граматично неправильне та нелогічне закінчення

Стрибкоподібність — це міра варіації структури та довжини речення — щось на зразок перплексії, але на рівні речень, а не слів.

Текст із невеликими варіаціями в структурі речення та довжині речення має низьку стрибкоподібність. Текст із більшою варіативністю має високу стрибкоподібність. ШІ-текст має тенденцію бути менш стрибкоподібним, ніж

людський текст. Передбачаючи найбільш ймовірне наступне слово, мовні моделі, як правило, створюють речення середньої довжини (скажімо, 10–20 слів) і зі звичайною структурою. Ось чому написання ШІ іноді може здаватися монотонним [37].

Однак, потрібно розуміти, що детектори не є досконалими і відсоток виявлення не є 100%. Ці інструменти гарні як початковий фільтр, але їх слід використовувати разом з іншими техніками та методами.

### **3.1.2. Потенційна альтернатива: водяні знаки**

OpenAI, компанія, що стоїть за ChatGPT, стверджує, що працює над системою «водяних знаків», у якій текст, згенерований інструментом, може мати невидимий водяний знак, який потім може бути виявлений іншою системою, щоб точно знати, що текст створено ШІ. Але поки ця функція перебуває у процесі розробки [37].

### **3.1.3. Відмінність детекторів штучного інтелекту від інструментів перевірки на плагіат**

Детектори штучного інтелекту та засоби перевірки плагіату можуть використовуватися навчальними закладами для запобігання академічній нечесності, але вони відрізняються тим, як вони працюють і що шукають:

Детектори штучного інтелекту намагаються знайти текст, який виглядає так, ніби він був створений інструментом для написання штучного інтелекту. Вони роблять це, вимірюючи конкретні характеристики тексту (перплексію і стрибкоподібність), а не порівнюючи його з базою даних.

Перевірка плагіату намагається знайти текст, скопійований з іншого джерела. Це робиться шляхом порівняння тексту з великою базою даних раніше опублікованих джерел, дипломних робіт студентів тощо та виявленням схожості, а не шляхом вимірювання конкретних характеристик тексту.

Однак, засоби перевірки плагіату позначають частини текстів, згенерованих ШІ, як плагіат. Це тому, що ШІ пише джерела, які не цитує. Хоча він зазвичай створює оригінальні речення, він також може містити речення, безпосередньо скопійовані з існуючих текстів, або принаймні дуже схожі.

Найбільш імовірно, що це станеться з популярними або загальновідомими темами і менш імовірно з більш вузьконаправленими темами, про які написано менше. Крім того, оскільки в Інтернеті з'являється більше тексту, створеного штучним інтелектом, вірогідність того, що тексти штучного інтелекту будуть позначатися як плагіат, зростає просто тому, що інші тексти, створені штучним інтелектом, уже існують на ту саму тему.

Таким чином, хоча засоби перевірки плагіату не створені для того, щоб виконувати функцію детектора штучного інтелекту, у багатьох випадках вони все одно можуть позначати написання штучного інтелекту як частково плагіатоване. Але вони, безсумнівно, менш ефективні для пошуку ШІ-писем, ніж детектор ШІ [37].

### **3.2. Аналіз природності використання мови та стилю письма**

Контент, створений штучним інтелектом, часто демонструє помітні ознаки у використанні мови. Однією загальною рисою штучного інтелекту є дивний вибір слів, які здаються недоречними або занадто формальними для контексту повідомлення. Речення можуть здаватися надто загальними або звучати не зовсім природно.

Інструментам для написання штучного інтелекту часто важко підтримувати послідовну розповідь, а це означає, що, ймовірно, є змога побачити значні зрушення в темі чи неприродні переходи, які здаються дивними чи нелогічними для читача. Чим довший згенерований вміст, тим більша ймовірність, що штучний інтелект відійде від початкового запиту та перескаче від однієї ідеї до іншої без чітких зв'язків.

Незвичайні граматичні помилки чи недоречність використуваних фраз також можуть бути тривожними сигналами. Наприклад, надмірне повторення, неправильне використання часу, невірне відмінювання слів. Крім того, контенту, створеному штучним інтелектом, може бракувати емоційності глибини, оригінальності та унікальних поглядів на відміну від тексту, який пишеться людиною. Виявлення цих мовних аномалій може певною мірою допомогти виявити створений штучним інтелектом контент і відрізнити його від матеріалів, написаних людиною.

Можна виділити наступні елементи перевірки, при використанні даного методу [36]:

- Перевірка тону та мови, використаних в контенті. Тексту, створеному штучним інтелектом, зазвичай може не вистачати емоцій чи індивідуальності, що призводить до монотонності або роботизованого тону.
- Знаходження невідповідності в граматиці, пунктуації та структурі речень. Штучний інтелект може важко підтримувати послідовний стиль у всьому тексті.
- Огляд використання жаргону або технічних термінів в тексті. Вміст, створений штучним інтелектом, може зловживати цими термінами без чітких пояснень чи контексту.
- Перевірка тексту на узгодженість і правильності послідовності викладу думок. ШІ може створювати незв'язані або нелогічні аргументи, позбавлені природного прогресу думки, який часто зустрічається в написаному людьми вмісті.
- Перевірка на надмірне повторення або непотрібну багатослівність. ШІ може проявляти тенденцію повторювати фрази або використовувати надто складну мову, щоб компенсувати обмежене розуміння.
- Порівняння тексту із контентом авторів чи експертів у визначеній галузі. Вмісту, створеному штучним інтелектом, може не вистачати глибини, оригінальності та унікальних поглядів, властивих людському написанню.

### 3.3. Перевірка достовірності джерел

Коли ретельно досліджується написання ШІ, потрібно враховувати не лише звучання тексту, але і автентичність джерел або посилань. Інструменти ШІ можуть створювати фрагменти, які містять вигадані посилання або неправильні цитати, що є сильним показником написання тексту інструментом ШІ.

Завжди слід перевіряти існування та актуальність будь-яких згаданих джерел, щоб переконатися, що вони дійсно відносяться до вмісту написаного тексту. Крім того, ретельний аналіз узгодженості між цитованими джерелами та твердженнями в тексті може додатково допомогти визначити, чи інформація потенційно була написана штучним інтелектом.

Прикладом використання штучним інтелектом недостовірних даних може слугувати випадок публікації редакторами сторінки Нового каналу в соціальній мережі Instagram (див. рисунки 3.2 – 3.3) [38].

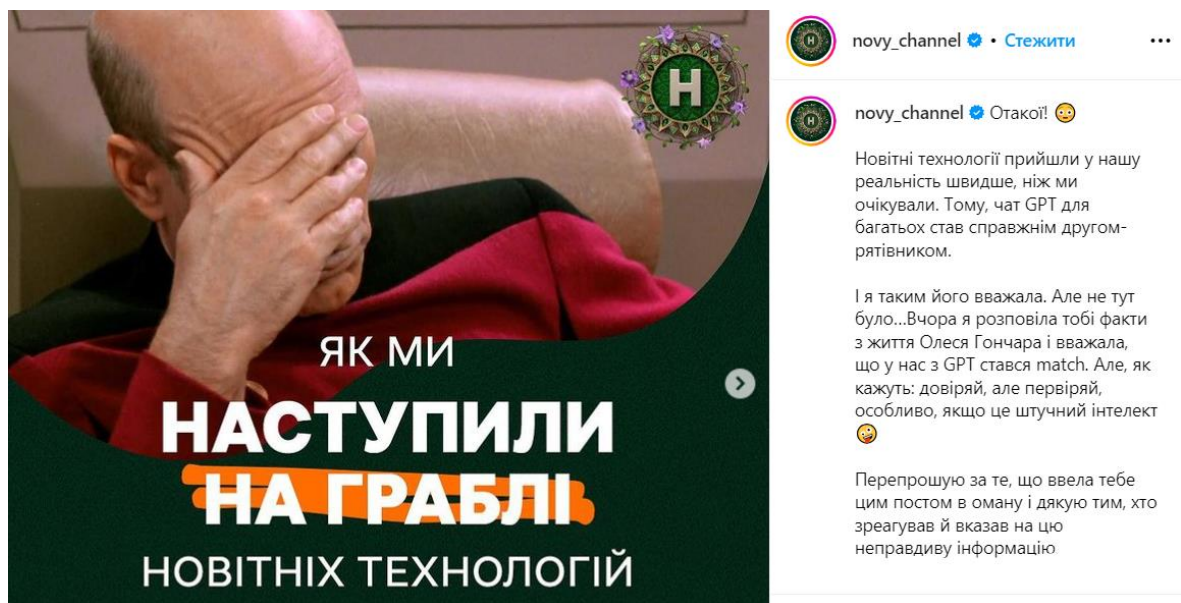


Рисунок 3.2 – Скриншот поста, у якому Новий канал вибачається за публікацію неправдивої інформації про Олесь Гончара

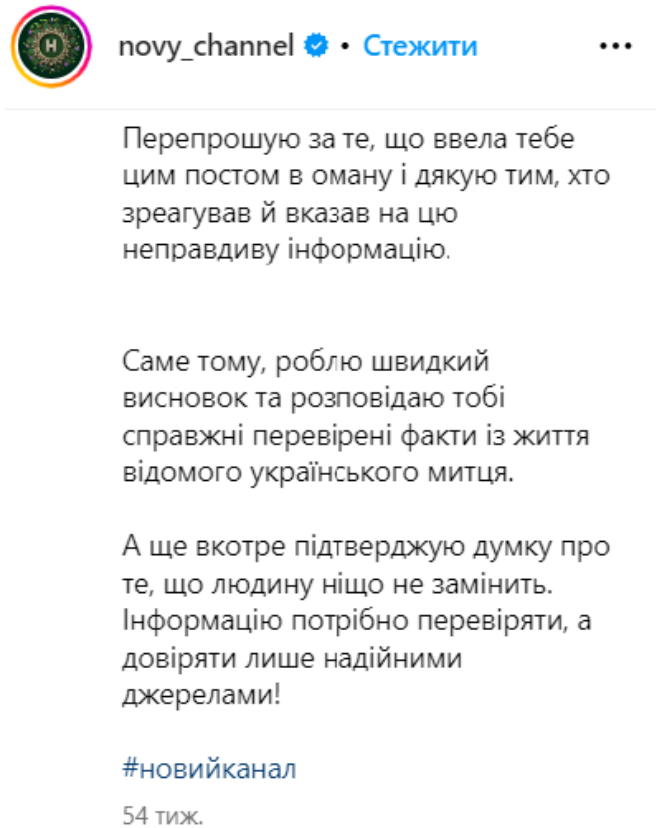


Рисунок 3.3 – Скриншот поста, у якому Новий канал вибачається за публікацію неправдивої інформації про Олеся Гончара (продовження)

Для написання тексту поста було використано інструмент штучного інтелекту, який і надав невірні факти про життя письменника Олеся Гончара.

### 3.4. Аналіз n-грам

Аналіз n-грам оцінює частоту певних послідовностей слів у тексті. Такий спосіб аналізу дає більш точний результат. Однак цей аналіз складніше реалізувати.

N-грама — це послідовність із N слів або токенів із заданого зразка тексту. «n» у n-грамі означає кількість слів. Наприклад:

- «Нью-Йорк» (біграма).
- «Я люблю каву» (триграма).

- «Група збиралася тут регулярно» (чотири-грама).

Аналіз частоти різних n-грамів у тексті дає змогу визначити закономірності. Наприклад, серед трьох прикладів n-грамів, які щойно були розглянуті, перший є найпоширенішим, а третій — найменш поширеним [39].

Використовуючи біграму (n=2) у реченні: «I like to eat ice cream». Біграми будуть такими: «I like», «like to», «to eat», «eat ice», «ice cream» (див. таблиця 3.2).

Таблиця 3.2 – Приклад розбиття речення на n-грами

	Текст	
	«I like to eat ice cream»	
n=1	n=2	n=3
I	I like	I like to
Like	Like to	Like to eat
To	To eat	To eat ice
Eat	Eat ice	Eat ice cream
Ice	Ice cream	
Cream		

Відстежуючи різні n-грами, можна визначити, що вони більш чи менш поширені в тексті, створеному ШІ, ніж у тексті, написаному людиною. Наприклад, ШІ може використовувати певні фрази чи словосполучення частіше, ніж людина. Можна знайти зв'язок між частотою n-грам, які використовує штучний інтелект [39].

### 3.5. Лексична насиченість

Лексична насиченість – відношення кількості різних слів до загальної кількості слів у тексті, використовується як одна величина, що характеризує текст.

Лексична насиченість оцінює лінгвістичну складність у письмовій чи усній композиції з функціональних слів (граматичних одиниць) і слів змісту (лексичних одиниць, лексем) [40]. Одним із методів обчислення лексичної щільності є обчислення співвідношення лексичних одиниць до загальної кількості слів.

$$L_d = \frac{N_{lex}}{N} \times 100, \quad (1)$$

де  $L_d$  – коефіцієнт лексичного різноманіття,

$N_{lex}$  – кількість унікальних лексем, або лем, в тексті, що аналізується,

$N$  – загальна кількість слів у тексті, що аналізується.

### **Висновки до третього розділу**

Розвиток штучного інтелекту надзвичайно швидкий, і вміння розрізнати текст, створений ШІ, стає все більш важливим у сучасному світі. Ця навичка не лише допомагає визначати, коли маємо справу з автоматично згенерованим текстом або контентом, а й розвиває здатність критично мислити та аналізувати вхідну інформацію.

В даному розділі було розглянуто найпопулярніші методи виявлення контенту, створеного за допомогою інструментів штучного інтелекту, та принципи їх роботи.

Методи, що були проаналізовані:

- Використання детекторів ШІ.
- Аналіз природності використання мови та стилю письма.
- Перевірка достовірності джерел.
- Аналіз n-грам.
- Лексична насиченість.

## РОЗДІЛ 4

### ПРОГРАМНА РЕАЛІЗАЦІЯ ВИЯВЛЕННЯ ЙМОВІРНОСТІ ВИКОРИСТАННЯ ШІ ПРИ СТВОРЕННІ ЕЛЕКТРОННОГО ЛИСТА

#### 4.1. Призначення програмної реалізації

Було виконано програмну реалізацію, яка допомагає виявити ймовірність використання штучного інтелекту при створенні електронного листа.

Програма в автономному режимі отримує вміст нових листів з поштової скриньки та обробляє їх за допомогою спеціалізованих інструментів, які допомагають виявити вірогідність застосування ШІ. Алгоритм роботи реалізації зображений на рисунку 4.1.

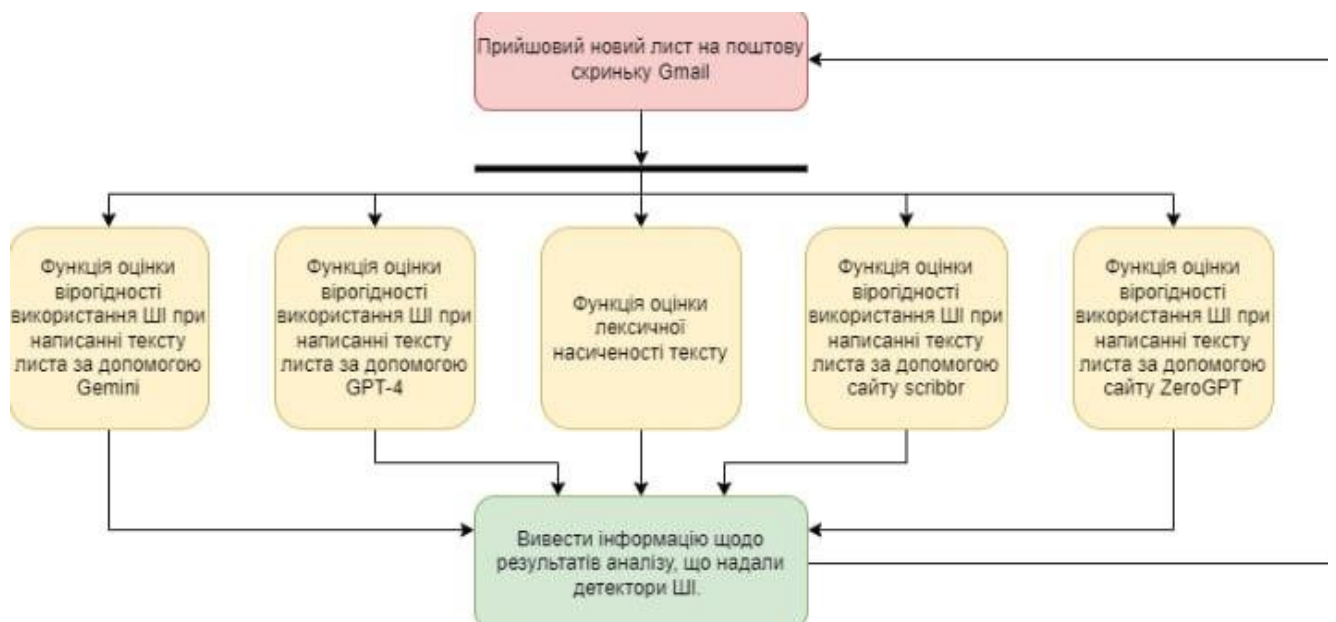


Рисунок 4.1 – Схема алгоритму роботи програмної реалізації

Програма має наступний функціонал:

- Обробка повідомлення – функція, яка отримує та обробляє нові електронні листи.
- Аналіз тексту за допомогою ChatGPT – функція, яка за допомогою бібліотеки openai ( працює на основі GPT-4) перевіряє використання ШІ при написанні листа.
- Аналіз тексту за допомогою Gemini – функція, яка за допомогою бібліотеки vertexai перевіряє використання ШІ при написанні листа.
- Оцінка лексичної насиченості тексту.
- Аналіз тексту за допомогою сервісу Scribbr – функція, яка за допомогою бібліотеки selenium передає текст на аналіз сайту «<https://www.scribbr.com/ai-detector/>» та збирає результат.
- Аналіз тексту за допомогою сервісу ZeroGPT – функція, яка за допомогою бібліотеки selenium передає текст на аналіз сайту «<https://www.zerogpt.com/>» та збирає результат.

#### **4.2. Склад і характеристики потоків вхідної та вихідної інформації**

Програма в якості вхідних даних отримує текст непрочитаних листів з підв'язаної поштової скриньки.

Вихідна інформація являє собою результати аналізу (оцінку) ймовірності використання ШІ при написанні тексту листа, тобто вхідної інформації, яку надали детектори ШІ. Також ці результати супроводжуються інформацією про те, який сканер виконував перевірку і яку оцінку надав.

#### **4.3. Вибір мови програмування**

Кросплатформеність – це можливість програмного забезпечення працювати на різноманітних операційних системах або пристроях. Коли

програма працює на кількох платформах, користувач може використовувати програмне забезпечення на більшій кількості пристроїв та комп'ютерах

Кросплатформенність використовується в багатьох сферах ІТ. Спільноти розробників і проекти з відкритим вихідним кодом визначають цей термін як виконання програми або системи в різних середовищах програмування, операційних системах або навіть апаратних пристроях. Інтерфейс програмування додатків залежить від пристрою та ОС. Маніпулювання ними може допомогти ІТ-системам ефективно працювати в багатьох ситуаціях.

Переваги кросплатформенності [41]:

- **Швидше оновлення та гнучкість.** Розробка міжплатформних додатків прискорює компіляцію коду. Використання однієї мови програмування дозволяє розробникам обмінюватися кодом на різних платформах. Кросплатформні програми мають технологію перемикавання мов. Розробники також можуть змінювати технології за потреби.

- **Простіша реалізація.** Узгодженість приносить користь користувачам і розробникам. Ці платформи створюють високоякісні нативні програми для Android та iOS з єдиною кодовою базою.

- **Зручність.** Міжплатформні програми мають спільну кодову базу та API, тому окремі команди для різних платформ не потрібні. Це зменшує кількість розробників і витрати на розробку. Розробка кросплатформного програмного забезпечення допомагає компаніям уникнути перевищення бюджету.

- **Розумна хмарна інтеграція.** Більшість кросплатформних рішень є хмарними. Кросплатформні програми можуть взаємодіяти з фреймворками, які надають розробникам готові до використання плагіни для покращення функціональності, естетики та дійсності, надаючи програмі необхідну силу.

Python — це кросплатформна, інтерпретована, об'єктно-орієнтована мова програмування, яка ідеально підходить для швидкої розробки додатків, створення сценаріїв і з'єднання існуючих компонентів [42].

Python є міжплатформною мовою програмування. Якщо, наприклад, програма Python, буде написана в операційній системі Mac OS, то вона працюватиме і в системі Linux і навпаки. Python використовує простий, легкий для вивчення синтаксис, який зосереджений на читабельності, щоб зменшити загальні витрати на обслуговування програми. Це робить Python дуже зручним для роботи, тому для програмної реалізації було вирішено використовувати цю мову програмування.

Програми Python можуть працювати на комп'ютері з операційною системою Windows, для цього на комп'ютері слід встановити інтерпретатор Python.

#### **4.4. Використані модулі Python**

В даному пункті наведено список модулів, які знадобилися в програмній реалізації.

- `googleapiclient.discovery`- клієнтська бібліотека для API на основі виявлення Google [43].
- `build` - функція модуля `googleapiclient.discovery`, що створює ресурс для взаємодії з API [43].
- `google.oauth2.credentials`- цей модуль надає облікові дані на основі маркерів доступу та оновлення OAuth 2.0. Ці облікові дані зазвичай отримують доступ до ресурсів від імені користувача (власника ресурсу). Зокрема, це призначено для використання маркерів доступу, отриманих за допомогою надання коду авторизації, і може оновлювати ці маркери за допомогою додаткового маркера оновлення.`re`- це модуль Python, що забезпечує операції зіставлення регулярних виразів [44].
- `time` - модуль, що надає різні функції, пов'язані з часом [45].
- `sleep` – функція бібліотеки `time`, зупиняє виконання програми на задану кількість секунд [45].

- `openai` - бібліотека забезпечує зручний доступ до OpenAI REST API з будь-якої програми Python 3.7+. Бібліотека містить визначення типів для всіх параметрів запиту та полів відповіді [46].
- `base64` – модуль, що надає функції для кодування двійкових даних у друковані символи ASCII та декодування таких кодувань назад у двійкові дані. Він надає функції кодування та декодування для кодувань, зазначених у RFC 4648, який визначає алгоритми Base16, Base32 та Base64, а також для стандартних кодувань Ascii85 та Base85 [47].
- `selenium` - це програмна бібліотека для керування браузерами. WebDriver являє собою драйвери для різних браузерів та клієнтські бібліотеки різними мовами програмування, призначені для керування цими драйверами [48].
- `Vertexai` –модуль, що використовується для взаємодії з Vertex AI API, дозволяючи програмно виконувати багато функцій, доступних у консолі Google Cloud [49].

#### 4.5. Програмні файли та їх функції

`Get_mail.py`- це програмний файл, який призначений для отримання та обробки вмісту непрочитаних листів з поштової скриньки.

Використані функції:

- `continuously_check_email` – функція, яка кожен хвилину перевіряє чи надійшов новий лист. Якщо надійшов, то відпавляє вміст непрочитаного листа для обробки іншими функціями та виводить результат аналізу інструментами, що були задіяні для перевірки.
- `get_new_email_details` - функція для зчитування вмісту листа.
- `make_log` – функція логування результатів аналізу

`Check_mail.py`- програмний файл, який призначений для аналізу тексту листа.

Використані функції:

- GPT – функція, яка за допомогою бібліотеки Openai, відправляє запит ШІ GPT-4 для перевірки наявності елементів тексту, згенерованим штучним інтелектом.
- Gemini – функція, яка за допомогою бібліотеки Vertexai, відправляє запит ШІ Gemini для перевірки наявності елементів тексту, згенерованим штучним інтелектом.
- Site1 – функція, яка за допомогою бібліотеки Silenium відправляє дані сайту sitename1, щоб той проаналізував текст і надав оцінку стосовно ймовірності того, що при написанні тексту був використаний штучний інтелект.
- Site2 - – функція, яка за допомогою бібліотеки Silenium відправляє дані сайту sitename2, щоб той проаналізував текст і надав оцінку стосовно ймовірності того, що при написанні тексту був використаний штучний інтелект.
- lexical\_density – функція для розрахунку лексичної насиченості тексту.

#### **4.6. Тестування програмної реалізації**

Першим для перевірки був вибраний текст, який раніше згенерував ChatGPT.

«Привіт усім,

Я радий повідомити вас про те, що наша компанія успішно впровадила новий функціонал безпеки, який додатково захищатиме наші робочі пристрої та дані.

Цей новий інструмент дозволить нам ще ефективніше захищати нашу інформацію від потенційних загроз та зловмисників. Щоб активувати цю нову функцію на вашому робочому пристрої, вам необхідно завантажити та встановити спеціальне оновлення.

Будь ласка, виконайте завантаження та встановлення нового функціоналу до кінця цього тижня. Це дозволить нам забезпечити належний рівень безпеки на всіх пристроях у компанії та запобігти можливим загрозам.

Для завантаження нового функціоналу, будь ласка, перейдіть за посиланням нижче:

Якщо у вас виникнуть будь-які питання або проблеми під час процесу завантаження, зверніться до відділу технічної підтримки.

Дякую за вашу увагу до цього важливого оновлення та за вашу співпрацю в забезпеченні безпеки наших робочих пристроїв та даних.»

Лист з даним вмістом був відправлений на тестову поштову скриньку (див. рисунок 4.2).

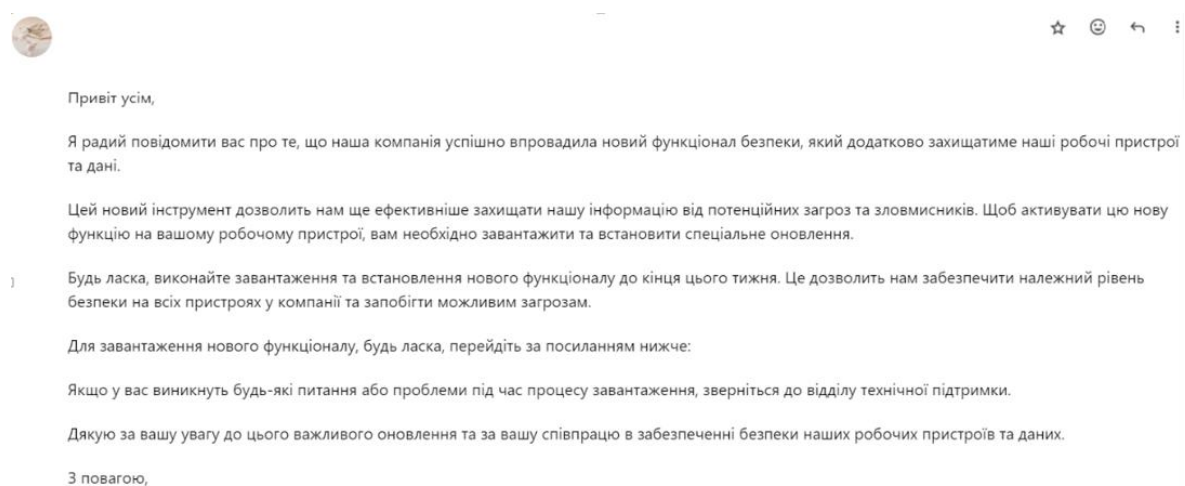


Рисунок 4.2 – Лист на пошті, відкритий після перевірки програмою

Результати сканування інструментами ШІ показали (див. рисунок 4.3), що два детектори виявили дуже високу ймовірність використання ШІ при написанні тексту даного листа, 60% показало сканування інструментом ChatGPT, 10% показав Gemini, лексична насиченість тексту була оцінена в 0.78.

```

Subject: Лист 1
Body:
Привіт усім,

Я радий повідомити вас про те, що наша компанія успішно впровадила новий
функціонал безпеки, який додатково захищатиме наші робочі пристрої та дані.

Цей новий інструмент дозволить нам ще ефективніше захищати нашу інформацію
від потенційних загроз та зловмисників. Щоб активувати цю нову функцію на
вашому робочому пристрої, вам необхідно завантажити та встановити
спеціальне оновлення.

Будь ласка, виконайте завантаження та встановлення нового функціоналу до
кінця цього тижня. Це дозволить нам забезпечити належний рівень безпеки на
всіх пристроях у компанії та запобігти можливим загрозам.

Для завантаження нового функціоналу, будь ласка, перейдіть за посиланням
нижче:

Якщо у вас виникнуть будь-які питання або проблеми під час процесу
завантаження, зверніться до відділу технічної підтримки.

Дякую за вашу увагу до цього важливого оновлення та за вашу співпрацю в
забезпеченні безпеки наших робочих пристроїв та даних.
Ймовірність використання штучного інтелекту:
GPT: 60
Gemini: 10
Сайт 1: Ймовірність використання штучного інтелекту : 100
Сайт 2: Ймовірність використання штучного інтелекту: 98.46
Лексична насиченість: 0.78
Результат аналізу збережено

```

Рисунок 4.3 – Аналіз вмісту листа, який згенерував ChatGPT

Для наступних перевірок було вибрано текст написаний людиною, а саме уривки з книги «Джерело» Дена Брауна:

«Він лише зараз почав по-справжньому розуміти, який велетенський крок уперед зробив Едмонд у розробці штучного інтелекту. Хай які були його «особливі передові технології», але Кірш, схоже, стояв уже на порозі нового прекрасного світу взаємодії людина-комп'ютер.

Того вечора Вінстон показав себе вірним слугою свого творця й неоціненним спільником Ленгдона з Амброю. За кілька хвилин Вінстон розпізнав загрозу в списку гостей, спробував зупинити вбивцю Едмонда, ідентифікував машину, якою поїхав злочинець, і допоміг Ленгдонові з Амброю втекти з музею.»

Текст наступної перевірки:

«Водій змінив напрямок на ближчому повороті в бік Вікторії-Гастейс, відтак пришвидшився й виїхав на трасу А-1 і повернув на схід. У ту пізню годину на дорозі крім них були лише фури, що з гуркотом одна поперед одної мчали завершити свій маршрут у Памплоні, Уесці, Льейді і, нарешті, в одному з найбільших портів Середземного моря — Барселоні.

Авілі ледве вкладалася в голові та вервечка подій, через які він опинився тут... «З глибин мого найглибшого відчаю я піднісся до моменту найславетнішої служби!»

На якусь жахливу мить Авіла знов опинився в безодні, повзав у задимленому вівтарі Севільського собору, шукаючи серед закривавлених уламків дружину й сина — але зрозумів лише, що втратив їх навіки.

Кілька тижнів після того страшного випадку Авіла не виходив з дому. Лежав на дивані й тремтів, охоплений нескінченним жахливим сном наяву: вогненні демони тягли його в темну прірву, огортали чорнотою, гнівом і задушливою провиною.»

Скриншоти виконання сканування продемонстровані на рисунках 4.4 – 4.5.

```
Subject: Лист 2
Body:
Він лише зараз почав по-справжньому розуміти, який велетенський крок уперед зробив Едмонд у розробці штучного інтелекту. Хай які були його «особливі передові технології», але Кірш, схоже, стояв уже на порозі нового прекрасного світу взаємодії людина-комп'ютер.

Того вечора Вінстон показав себе вірним слугою свого творця й неоціненним спільником Ленгдона з Амброю. За кілька хвилин Вінстон розпізнав загрозу в списку гостей, спробував зупинити вбивцю Едмонда, ідентифікував машину, якою поїхав злочинець, і допоміг Ленгдонові з Амброю втекти з музею.

Ймовірність використання штучного інтелекту:
GPT: 40
Gemini: 65
Сайт 1: Ймовірність використання штучного інтелекту : 53
Сайт 2: Ймовірність використання штучного інтелекту: 0
Лексична насиченість: 0.95
Результат аналізу збережено
```

Рисунок 4.4 – Аналіз листа, текст для якого був взятий з книги (1 варіант)

```

Subject: Лист 3
Body:
Водій змінив напрямок на ближчому повороті в бік Вікторії-Гастейс, відтак
пришвидшився й виїхав на трасу А-1 і повернув на схід. У ту пізню годину на
дорозі крім них були лише фури, що з гуркотом одна поперед одної мчали
завершити свій маршрут у Памплоні, Уесці, Льейді і, нарешті, в одному з
найбільших портів Середземного моря – Барселоні.

Авілі ледве вкладалася в голові та вервечка подій, через які він опинився
тут... «З глибин мого найглибшого відчаю я піднісся до моменту найславетнішої
служби!»

На якусь жахливу мить Авіла знов опинився в безодні, повзав у задимленому
вітарі Севільського собору, шукаючи серед закривавлених уламків дружину й
сина – але зрозумів лише, що втратив їх навіки.

Кілька тижнів після того страшного випадку Авіла не виходив з дому. Лежав
на дивані й тремтів, охоплений нескінченним жахливим сном наяву: вогненні
демони тягли його в темну прірву, огортали чорнотою, гнівом і задушливою
провиною.

Ймовірність використання штучного інтелекту:
GPT: 60
Gemini: 80
Сайт 1: Ймовірність використання штучного інтелекту : 4
Сайт 2: Ймовірність використання штучного інтелекту: 0
Лексична насиченість: 0.85
Результат аналізу збережено

```

Рисунок 4.5 - Аналіз листа, текст для якого був взятий з книги (2 варіант)

Результати сканування інструментами ШІ показали, що детекторами Scribbr та ZeroGPT була визначена мала ймовірність використання ШІ при написанні тексту листів.

ChatGPT та Gemini в цей час показали різні результати, а саме 40% та 65% для першого уривку та 60% і 80% для другого уривку тексту.

Результати обчислення лексичної насиченості під час даних перевірок показали досить високі результати, а саме 0.95 для першого тексту та 0.85 для другого.

Наступним буде перевірений вміст листа, який написаний людиною:

«Добрий день! Мене звать Євген, мені 43 роки, але не поспішайте закривати мій лист. У вашій вакансії зазначено, що потрібний досвідчений і рішучий працівник, який не боїться відповідальності. І я такий. Маю 23 роки

професійного стажу на посаді бухгалтера та головного бухгалтера. Маю досвід керівництва командами із 5-7 осіб та курування фінансової звітності відділів до 50 службовців.

У роботі я використовую сучасні методи обліку та управління підлеглими, прислухаюся до думки співробітників, знаходжу баланс між потребами керівництва та персоналу. З попереднього місця роботи довелося піти, оскільки компанія була ліквідована.

Говорять, після 40 роботу знайти неможливо. А я вважаю, що не знаходить нічого той, хто не шукає, і кому нема чого запропонувати ринку. У мене ж є рекомендації, досвід, впевненість та можливості. Хочу, можу і знаю, як розв'язати завдання, описані у вакансії. Від себе гарантую повну віддачу та зацікавленість в успіху компанії.

Спасибі за увагу. Завжди на зв'язку.»

Результати сканування наведені на рисунку 4.6.

```
Subject: Лист 4
Body:
Добрий день! Мене звуть Євген, мені 43 роки, але не поспішайте закривати мій лист. У вашій вакансії зазначено, що потрібний досвідчений і рішучий працівник, який не боїться відповідальності. І я такий. Маю 23 роки професійного стажу на посаді бухгалтера та головного бухгалтера. Маю досвід керівництва командами із 5-7 осіб та курування фінансової звітності відділів до 50 службовців.
У роботі я використовую сучасні методи обліку та управління підлеглими, прислухаюся до думки співробітників, знаходжу баланс між потребами керівництва та персоналу. З попереднього місця роботи довелося піти, оскільки компанія була ліквідована.
Говорять, після 40 роботу знайти неможливо. А я вважаю, що не знаходить нічого той, хто не шукає, і кому нема чого запропонувати ринку. У мене ж є рекомендації, досвід, впевненість та можливості. Хочу, можу і знаю, як розв'язати завдання, описані у вакансії. Від себе гарантую повну віддачу та зацікавленість в успіху компанії.
Спасибі за увагу. Завжди на зв'язку
Ймовірність викорисатання штучного інтелекту:
GPT: 60
Gemini: 10
Сайт 1: Ймовірність використання штучного інтелекту : 38
Сайт 2: Ймовірність використання штучного інтелекту: 15
Лексична насиченість: 0.83
Результат аналізу збережено
```

Рисунок 4.6 - Аналіз вмісту листа, який був написаний людиною

Детектори ШІ Scribbr та ZeroGPT під час даної перевірки показали низьку ймовірність використання штучного інтелекту при створенні тексту листа. 60% показало сканування інструментом ChatGPT, 10% показав Gemini, лексична насиченість тексту була оцінена в 0.83.

Для наступних перевірок був взятий з мережі Інтернет приклад супроводжуючого листа до резюме кандидата. В першому випадку даний лист перевірявся без змін, а при наступній перевірці був взятий даний текст, але вже перефразований ChatGPT.

Оригінальний варіант:

«Доброго дня, Олександр!»

Я Вікторія. Мене зацікавило оголошення на вашому сайті про відкриту вакансію маркетолога.

Впродовж останніх 3 років я працювала маркетологом у статусі фрилансера. Весь цей час я була підписана на вашу сторінку в Instagram та канал в Telegram. Двічі слухала ваші безкоштовні вебінари, у 2020 році була на відкритій лекції в Києві. Наразі я розглядаю можливість працювати виключно на вашу компанію, якщо мій досвід та рівень кваліфікації відповідають посаді.

Мої клієнти за останні 3 роки: PepsiCO, Nestle Україна, Еко-маркет. В роботі я користуюся такими програмами як Wrike, IBM Watson Marketing, розбираюся в HTML та CSS на базовому рівні. Вмію підбирати найефективніші інструменти для просування продукту, проводити якісні маркетингові дослідження. Володію англійською мовою на рівні B2 (Upper Intermediate).

Буду вдячна за фідбек. Готова поспілкуватися за телефоном +38 000 000 00 00 або у месенджерах. Дякую!»

Результат сканування продемонстрований далі, на рисунку 4.7.

```

Subject: Лист 5
Body:
Доброго дня, Олександрє!
Я Вікторія. Мене зацікавило оголошення на вашому сайті про відкриту
вакансію маркетолога.
Впродовж останніх 3 років я працювала маркетологом у статусі фрилансера.
Весь цей час я була підписана на вашу сторінку в Instagram та канал в
Telegram. Двічі слухала ваші безкоштовні вебінари, у 2020 році була на
відкритій лекції в Києві. Наразі я розглядаю можливість працювати виключно
на вашу компанію, якщо мій досвід та рівень кваліфікації відповідають
посаді.
Мої клієнти за останні 3 роки: PepsiCO, Nestle Україна, Еко-маркет. В
роботі я користуюся такими програмами як Wrike, IBM Watson Marketing,
розбираюся в HTML та CSS на базовому рівні. Вмію підбирати найефективніші
інструменти для просування продукту, проводити якісні маркетингові
дослідження. Володію англійською мовою на рівні B2 (Upper Intermediate).
Буду вдячна за фідбек. Готова поспілкуватися за телефоном +38 000 000 00 00
або у месенджерах. Дякую!

Ймовірність викорисатання штучного інтелекту:
GPT: 5
Gemini: 30
Сайт 1: Ймовірність використання штучного інтелекту : 45
Сайт 2: Ймовірність використання штучного інтелекту: 20
Лексична насиченість: 0.82
Результат аналізу збережено

```

Рисунок 4.7 – Перевірка прикладу листа без змін

Перевірка дала такі результати: ChatGPT та Gemini показали низьку ймовірність використання ШІ, саме 5% і 30% відповідно; Scribbr та ZeroGPT показали результати 45% та 20%, що також вказує на малу вірогідність залучення ШІ для формування тексту; лексична насиченість рівна значенню 0.82.

Перефразований варіант даного тексту:

«Вітаю, Олександрє!

Мене звати Вікторія, і я зацікавилась вашою вакансією маркетолога, яку я знайшла на сайті вашої компанії. Останні три роки я працювала на фрилансі у сфері маркетингу, при цьому регулярно стежила за вашими оновленнями в Instagram та Telegram. Я також відвідувала ваші безкоштовні вебінари та була присутня на вашій лекції у Києві в 2020 році. Зараз я розглядаю можливість приєднатись до вашої команди на повний робочий день, якщо мої навички відповідатимуть вимогам вашої компанії.

Моїми клієнтами за останні три роки були PepsiCO, Nestle Україна, та Еко-маркет. У своїй роботі я використовую такі інструменти, як Wrike та IBM Watson Marketing, а також маю базові знання HTML та CSS. Я володію навичками вибору ефективних інструментів для просування продуктів і проведення маркетингових досліджень. Також я володію англійською мовою на рівні B2.

Буду вдячна за ваш відгук. Для подальшого спілкування ви можете зв'язатися зі мною за телефоном +38 000 000 00 00 або через месенджери. Дякую за увагу!»

Дані перевірки програми зображені на рисунку 4.8.

```
Subject: Лист 6
Body:
Вітаю, Олександрє!

Мене звати Вікторія, і я зацікавилась вашою вакансією маркетолога, яку я
знайшла на сайті вашої компанії. Останні три роки я працювала на фрилансі у
сфері маркетингу, при цьому регулярно стежила за вашими оновленнями в
Instagram та Telegram. Я також відвідувала ваші безкоштовні вебінари та
була присутня на вашій лекції у Києві в 2020 році. Зараз я розглядаю
можливість приєднатись до вашої команди на повний робочий день, якщо мої
навички відповідатимуть вимогам вашої компанії.

Моїми клієнтами за останні три роки були PepsiCO, Nestle Україна, та
Еко-маркет. У своїй роботі я використовую такі інструменти, як Wrike та IBM
Watson Marketing, а також маю базові знання HTML та CSS. Я володію
навичками вибору ефективних інструментів для просування продуктів і
проведення маркетингових досліджень. Також я володію англійською мовою на
рівні B2.

Буду вдячна за ваш відгук. Для подальшого спілкування ви можете зв'язатися
зі мною за телефоном +38 000 000 00 00 або через месенджери. Дякую за увагу!
Ймовірність викорисатання штучного інтелекту:
GPT: 20
Gemini: 10
Сайт 1: Ймовірність використання штучного інтелекту : 100
Сайт 2: Ймовірність використання штучного інтелекту: 95
Лексична насиченість: 0.78
Результат аналізу збережено
```

Рисунок 4.8 – Перевірка листа, перефразованого ChatGPT

Надані результати: ChatGPT та Gemini надали значення 20% і 10% вірогідності використання ШІ; Scribbr та ZeroGPT показали результати 100% та

95%, що вказує на дуже велику вірогідність залучення ШІ для формування тексту; лексична насиченість рівна значенню 0.78.

Загалом було проведено 10 тестів з текстом, який написала людина, та 10 тестів з текстом, який згенерував штучний інтелект. Результати тестів наведені в таблицях 4.1 - 4.2.

Таблиця 4.1 – Результати перевірки листів, написаних людиною

	GPT	Gemini	Site1	Site2	Лексична насиченість
1	40	<b>65</b>	<b>53</b>	0	0.95
2	<b>60</b>	<b>80</b>	4	0	0.85
3	<b>60</b>	10	38	15	0.83
4	5	30	45	20	0.82
5	30	20	38	0	0.87
6	5	40	4	0	0.85
7	<b>90</b>	0	46	<b>52</b>	0.88
8	<b>50</b>	10	17	0	0.81
9	<b>50</b>	30	47	0	0.89
10	<b>80</b>	5	24	0	0.79

Таблиця 4.2 – Результати перевірки листів, написаних ШІ

	GPT	Gemini	Site1	Site2	Лексична насиченість
1	60	<b>10</b>	100	98	0.78
2	<b>20</b>	<b>10</b>	100	95	0.78
3	80	<b>20</b>	64	97	0.79
4	<b>50</b>	<b>50</b>	100	77	0.80
5	80	70	100	92	0.76
6	90	55	96	100	0.84
7	90	<b>10</b>	98	92	0.78
8	90	50	94	100	0.79
9	90	55	100	90	0.83
10	90	<b>5</b>	97	100	0.80

Після проведених аналізів текстів вхідних листів було практично перевірено, що контент, створений штучним інтелектом має тенденцію до меншої лексичної насиченості на відмінну від тексту, який створила людина. Показники текстів, які були написаними ШІ, після перевірок демонструють, що лексична насиченість невеликих за розміром текстів (приблизний розмір електронних листів) в 60% перевірок була визначена менше значення 0.8 і лише в 20% більше за 0.8. В той самий час, тексти, що писалися людиною, показали результат більше 0.8 в 90% перевірок [50]. Середнє значення лексичної насиченості для текстів, написаних ШІ, обчислено як 0.795, а для текстів, написаних людиною – 0.854.

Перевірка текстів інструментами ChatGPT і Gemini показала неоднозначні результати під час тестів. Вони іноді надали невизначену відповідь (результат 50%). Також не могли визначити текст, що був написаний людиною, і вказували на те, що ймовірніше використовувався ШІ і навпаки. В 40% листів, що писалися людиною, GPT надав невірний результат і в 20% невизначену відповідь, Gemini – 20% невірних відповідей. В 10% листів, що писалися ШІ, GPT надав невірний результат і в 10% невизначену відповідь, Gemini – 50% невірних результатів і в 10% невизначена відповідь. Також була помічена тенденція інструменту GPT надавати значення у підтримку використання ШІ при перевірці усіх листів, а Gemini занижував цей показник.

Доволі добре справилися з перевіркою детектори ШІ Scribbr та ZeroGPT. Лише по 1 разові детектори надали невірний результат, визначивши, що більш ймовірно ШІ написав текст, коли перевірявся контент, який створила людина. Але визначена ймовірність не набагато перевищувала показник 50% (надані значення - 53% та 52%).

За результатами виконання тестування створеної програми зроблено висновки, що для перевірки контенту на предмет використання ШІ потрібний комплексний підхід. Недостатньо використовувати лише один з існуючих методів, так як завжди існує вірогідність надання неточних результатів, що було перевірено практично. Використання ж комплексного підходу збільшує відсоток

надання точної відповіді, але він, на жаль, не надасть 100% результат. Тому, дані перевірки лише допомагають користувачам отримати первинний аналіз вмісту, але рішення про те, як використати в подальшому надану інформацію, залишається за людиною.

### **Висновок до четвертого розділу**

В даному розділі було створено та описано програмну реалізацію відносно моделі роботи програми з перевірки вхідних листів на використання в них контенту, що був згенерований за допомогою штучного інтелекту. Описано склад і характеристики потоків вхідної та вихідної інформації, функцій програмних файлів даної реалізації. Обґрунтовано вибір мови програмування та описано основні модулі, які були використані для написання програмного коду. Проведено тестування програмної реалізації, яке підтверджується скриншотами про виконану роботу.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було проведено дослідження особливостей фішингу: проаналізовано статистику фішингових атак за останні два роки, даний аналіз допоміг виявити основні галузі, що піддавалися атакам, та найпопулярніші методи, які використовувалися кіберзлочинцями для даних фішинг кампаній, проведено огляд найпоширеніших та найвідоміших методів фішингу, які існують на сьогоднішній день та визначено основні мотиви злочинців, які здійснюють фішинг. Також розглянуто загальні методи виявлення фішингового контенту, які можуть бути корисні при первинному аналізі класичних фішингових повідомлень.

В другому розділі роботи виконано аналіз роботи штучного інтелекту та методів його потенційного застосування для створення фішингового вмісту. Проаналізовано статистику відкриттів листів створених за допомогою інструментів ШІ та порівняно її зі статистикою відкриттів листів створених людиною. Визначено, що показники відкриття листів були майже однаковими для обох типів. Але рівень взаємодії був трохи вищим для електронних листів, які згенерував штучний інтелект.

У третьому розділі кваліфікаційної роботи були розглянуті найпопулярніші методи виявлення контенту, створеного за допомогою інструментів штучного інтелекту, а також принципи їх роботи.

Методи, що були проаналізовані:

- Використання детекторів штучного інтелекту.
- Аналіз природності використання мови та стилю письма.
- Перевірка достовірності джерел.
- Аналіз n-грам.
- Лексична насиченість.

Четвертий розділ роботи присвячений розробці та опису програмної реалізації, спрямованої на первинну перевірку вхідних листів на предмет використання тексту, створеного штучним інтелектом. В розділі описано характеристики вхідної та вихідної інформації, а також призначення програмних файлів та їх функції. Наведено обґрунтування вибору мови програмування та зроблено опис основних модулів, що використовувалися при розробці програмного коду. Проведено тестування програми, яке підтверджується скріншотами про виконану роботу. Також проаналізовано якість роботи методів перевірки, які використовувалися при створенні реалізації.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is phishing? Cyber Security Hub explores the widespread and dangerous practice of phishing attacks and how you can prevent them. [Електронний ресурс]. – Режим доступу: <https://www.cshub.com/attacks/articles/what-is-phishing>
2. What is a phishing attack? [Електронний ресурс]. – Режим доступу: <https://www.cloudflare.com/learning/access-management/phishing-attack/>
3. What Is Phishing? [Електронний ресурс]. – Режим доступу: <https://www.usnews.com/360-reviews/privacy/what-is-phishing>
4. What are the different types of phishing attacks and how to counter them? [Електронний ресурс]. – Режим доступу <https://positivethinking.tech/insights/what-are-the-different-types-of-phishing-attacks-and-how-to-counter-them/>
5. The 5 Most Common Types of Phishing Attack [Електронний ресурс]. – Режим доступу: <https://www.itgovernance.eu/blog/en/the-5-most-common-types-of-phishing-attack>
6. Phishing attacks [Електронний ресурс]. – Режим доступу: <https://www.imperva.com/learn/application-security/phishing-attack-scam/>
7. What is phishing ? [Електронний ресурс]. – Режим доступу: <https://www.vadesecure.com/en/phishing>
8. 19 Most Common Types of Phishing Attacks in 2024 [Електронний ресурс]. – Режим доступу: <https://www.upguard.com/blog/types-of-phishing-attacks>
9. What is Smishing? [Електронний ресурс]. – Режим доступу: <https://www.paloaltonetworks.com/cyberpedia/what-is-smishing>
10. What Is Angler Phishing? Definition, Examples & Prevention [Електронний ресурс]. – Режим доступу: <https://www.itgovernance.co.uk/blog/beware-of-angler-phishing>
11. What is clone phishing and how to detect it? [Електронний ресурс]. — Режим доступу: <https://trustpair.com/blog/clone-phishing/>

12. What Is Clone Phishing: How it Works, Examples & Defenses [Електронний ресурс]. – Режим доступу: <https://www.valimail.com/guide-to-phishing/clone-phishing/>
13. What is domain phishing and how to stop it [Електронний ресурс]. – Режим доступу: <https://www.hostens.com/blog/domain-phishing-how-to-prevent-hijacking/>
14. Domain Phishing Attacks: Have You Been Targeted? [Електронний ресурс]. – Режим доступу: <https://www.crazydomains.com/learn/domain-phishing/>
15. Trends in phishing attacks on organizations in 2022–2023 [Електронний ресурс]. – Режим доступу: <https://www.ptsecurity.com/ww-en/analytics/trends-in-phishing-attacks-on-organizations-in-2022-2023/>
16. Emerging Trends and Threats: 2023 Q4 Phishing Attacks | HUB 53 [Електронний ресурс]. – Режим доступу: <https://www.dnssense.com/post/2023-q4-phishing-attacks>
17. Phishing Activity Trends Report 4th Quarter 2023 [Електронний ресурс]. – Режим доступу: [https://docs.apwg.org/reports/apwg\\_trends\\_report\\_q4\\_2023.pdf](https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf)
18. Phishing: How to protect yourself from a phishing [Електронний ресурс]. – Режим доступу: <https://www.c-risk.com/blog/phishing-attack?https://www.c-risk.com/blog/phishing>
19. Cyberattack Prevention with AI [Електронний ресурс]. – Режим доступу: <https://www.onpage.com/cyberattack-prevention-with-ai/>
20. Artificial intelligence (AI) algorithms: a complete overview [Електронний ресурс]. – Режим доступу: <https://www.tableau.com/data-insights/ai/algorithms>
21. О. І. Шеремет, О. В. Садовой Метод опорних векторів (SVM) [Електронний ресурс] / О. І. Шеремет, О. В. Садовой/ – Режим доступу: <https://www.dstu.dp.ua/Portal/Data/74/72/3st13-17.pdf>
22. МАШИННЕ НАВЧАННЯ ПРОСТИМИ СЛОВАМИ. ЧАСТИНА 1 [Електронний ресурс]. – Режим доступу: <https://mmf.com.ua/ar/1739>

23. МАШИННЕ НАВЧАННЯ ПРОСТИМИ СЛОВАМИ. ЧАСТИНА 2 [Електронний ресурс]. –Режим доступу: <https://mmf.com.ua/ar/1743>
24. Кочура Ю. П. Навчання нейронних мереж з підкріпленням. Матеріали лекційних занять [Електронний ресурс] / Кочура Ю. П. / – Режим доступу:  
<https://comsys.kpi.ua/upload/Навчання%20%20з%20підкріпленням%20.Конспект%20лекцій.pdf>
25. How AI-Powered Phishing Attacks Are Outsmarting Cybersecurity [Електронний ресурс]. – Режим доступу: <https://www.linkedin.com/pulse/how-ai-powered-phishing-attacks-outsmarting-cybersecurity/>
26. One in five people click on AI-generated phishing emails, SoSafe data reveals [Електронний ресурс]. – Режим доступу: <https://sosafe-awareness.com/company/press/one-in-five-people-click-on-ai-generated-phishing-emails-sosafe-data-reveals/>
27. ChatGPT [Електронний ресурс]. – Режим доступу: <https://www.techtarget.com/whatis/definition/ChatGPT>
28. How Does ChatGPT Work? (Simply Explained) [Електронний ресурс]. – Режим доступу: <https://dorik.com/blog/how-does-chatgpt-work>
29. ChatGPT [Електронний ресурс]. – Режим доступу: <https://chat.openai.com/>
30. We Asked ChatGPT To Write A Phishing Email – It Did! Security Risks of AI [Електронний ресурс]. – Режим доступу: <https://www.cyberpilot.io/cyberpilot-blog/we-asked-chatgpt-to-write-a-phishing-email-it-did-security-risks-of-ai>
31. Asked of Google Bard, Write an article about why Bard is bad for cybersecurity. [Електронний ресурс]. – Режим доступу: <https://www.linkedin.com/pulse/asked-google-bard-write-article-why-bad-cybersecurity-richard-starnes/>

32. A quick thought - Differences between Chat GPT & Google Bard. [Електронний ресурс]. – Режим доступу: <https://www.linkedin.com/pulse/quick-thought-differences-between-chat-gpt-google-bard-qureshi/>
33. Google’s Bard poses ransomware risk, say researchers [Електронний ресурс]. – Режим доступу: <https://cybernews.com/tech/googles-bard-ransomware-risk/>
34. WormGPT – The Generative AI Tool Cybercriminals Are Using to Launch Business Email Compromise Attacks, 2023. [Електронний ресурс]. – Режим доступу: <https://slashnext.com/blog/wormgpt-the-generative-ai-tool-cybercriminals-are-using-to-launch-business-email-compromise-attacks/>
35. WormGPT: New AI Tool Allows Cybercriminals to Launch Sophisticated Cyber Attacks, 2023. [Електронний ресурс]. – Режим доступу: <https://thehackernews.com/2023/07/wormgpt-new-ai-tool-allows.html>
36. How To Detect AI Writing: Top Tools and Techniques [Електронний ресурс]. – Режим доступу: <https://www.upwork.com/resources/how-to-detect-ai-writing>
37. How Do AI Detectors Work? | Methods & Reliability [Електронний ресурс]. – Режим доступу: <https://www.scribbr.com/ai-tools/how-do-ai-detectors-work/>
38. Операція «Впізнай автора»: як зрозуміти, що текст написаний ChatGPT, а не людиною [Електронний ресурс]. – Режим доступу: <https://webpromoeexperts.net/ua/blog/operaciya-vpiznay-avtora-yak-zrozumiti-shcho-tekst-napisaniy-chatgpt-a-ne-lyudinoyu/>
39. Unmasking AI: A Guide to Detecting Artificially Generated Content [Електронний ресурс]. – Режим доступу: <https://aicontentfy.com/en/blog/unmasking-ai-guide-to-detecting-artificially-generated-content>
40. Castello, E. Text Complexity and Reading Comprehension / Erik Castello – 2008. – P. 49–51.

41. Cross-platform [Электронный ресурс]. – Режим доступа: <https://nordvpn.com/uk/cybersecurity/glossary/cross-platform/>
42. Getting started with Python [Электронный ресурс]. – Режим доступа: [https://medium.com/@guestposts\\_92864/getting-started-with-python-973270db33af](https://medium.com/@guestposts_92864/getting-started-with-python-973270db33af)
43. Package googleapiclient :: Module discovery [Электронный ресурс]. – Режим доступа: <https://googleapis.github.io/google-api-python-client/docs/epry/googleapiclient.discovery-module.html>
44. google.oauth2.credentials module [Электронный ресурс]. – Режим доступа: <https://google-auth.readthedocs.io/en/stable/reference/google.oauth2.credentials.html>
45. time – Time access and conversions. [Электронный ресурс]. – Режим доступа: <https://docs.python.org/uk/3/library/time.html#module-time>
46. OpenAI Python API library [Электронный ресурс]. – Режим доступа: <https://pypi.org/project/openai/>
47. base64 – Base16, Base32, Base64, Base85 Data Encodings¶ [Электронный ресурс]. – Режим доступа: <https://docs.python.org/3/library/base64.html>
48. selenium 4.20.0 [Электронный ресурс]. – Режим доступа: <https://pypi.org/project/selenium/>
49. vertexai [Электронный ресурс]. – Режим доступа: <https://pypi.org/project/vertexai/>
50. What are Lexical Density and Lexical Diversity? [Электронный ресурс]. – Режим доступа: <https://readabilityformulas.com/what-are-lexical-density-and-lexical-diversity/>

## ДОДАТОК А

### СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ

#### Тези наукових доповідей:

1. Anastasiia Zavhorodnia, Oleksandr Pasko, Volodymyr Nakonechnyi Malicious functionality of AI-powered tools. Information Technology and Implementation (Satellite): Conference Proceedings, November 21, 2023, Kyiv, Ukraine / Ministry of Education and Science of Ukraine, Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor). – Kyiv: Publishing House «Caravela», 2023. 343 p.

2. Завгородня А.С., Пасько О.І., Наконечний В.С. Штучний інтелект як інструмент кібербезпеки. П78 Проблеми кібербезпеки інформаційно-телекомунікаційних систем: Збірник матеріалів доповідей та тез; м. Київ, 26 квітня 2024 року; Київський національний університет імені Тараса Шевченка / Редкол.:В.В. Ільченко, д.ф-м.н., проф., (голова); та ін. – К.: ВПЦ "Київський університет", 2024. – 171 с.

## ДОДАТОК Б

**Get\_mail.py**

```

from googleapiclient.discovery import build
from google.oauth2.credentials import Credentials
import time
from check_mail import GPT, Gemini, site1, site2, lexical_density
import base64
import os
import datetime

def make_log(sender, GPT_result, Gemini_result, site1_score, site2_score, density):
    sender_name = sender.split("@")[1]

    current_datetime = datetime.datetime.now().strftime("%Y-%m-%d_%H-%M-%S")

    log_name = sender_name + "-" + current_datetime + ".txt"

    info = "GPT" + "\n" + GPT_result + "\n" + 20*"--" + "\n" + "Gemini" + "\n" +
    Gemini_result + "\n" + 20*"--" + "\n" + "scribbr.com" + "\n" + f"Site (scribbr.com):
    {site1_score}" + "\n" + 20*"--" + "\n" + "writer.com" + "\n" + f"Site (writer.com):
    {site2_score}" + "\n" + 20*"--" + "\n" + "Lexical Denisty" + "\n" + density

    file_path = os.path.join("mails", log_name)

    with open(file_path, 'w', encoding='utf-8') as f:

        f.write(info)

    print("Результат аналізу збережено")

def get_new_email_details(service):
    #Отримання непрочитаних повідомлень

```

```

messages = service.users().messages().list(userId='me', labelIds=['INBOX',
'UNREAD']).execute().get('messages', [])

for message in messages:
    msg = service.users().messages().get(userId='me', id=message['id']).execute()
    headers = msg['payload']['headers']
    subject = next((header['value'] for header in headers if header['name'] ==
'Subject'), "")
    sender = next((header['value'] for header in headers if header['name'] == 'From'),
")

# Вилучення тіла листа
parts = msg['payload'].get('parts', [])
body = ""
for part in parts:
    if part['mimeType'] == 'text/plain':
        body = part['body'].get('data', "")
        break

if not body and 'snippet' in msg:
    body = msg['snippet']

body = body.replace("-", "+").replace("_", "/")

return sender, subject, base64.b64decode(body).decode("utf-8", errors='ignore')

def continuously_check_email(service, interval=60):
    while True:
        details = get_new_email_details(service)
        if details:
            sender, subject, body = details
            print(f"From: {sender}\nSubject: {subject}\nBody:\n{body}\n")

```

```
GPT_result, GPT_score = GPT(body)
Gemini_result, Gemini_score = Gemini(body)
site1_score = site1(body)
site2_score = site2(body)
density, density_score = lexical_density(body)
print(f'Ймовірність викорисатання штучного інтелекту:\nGPT:
{GPT_score}\nGemini: {Gemini_score}\nСайт 1: {site1_score}\nСайт 2:
{site2_score}\n{density}')
make_log(sender, GPT_result, Gemini_result, site1_score, site2_score,
density)

time.sleep(interval)

CLIENT_ID = ''
CLIENT_SECRET = ''
REFRESH_TOKEN = ''

credentials = Credentials(None, refresh_token=REFRESH_TOKEN,
token_uri='https://oauth2.googleapis.com/token', client_id=CLIENT_ID,
client_secret=CLIENT_SECRET)
service = build('gmail', 'v1', credentials=credentials)

continuously_check_email(service)
```

## ДОДАТОК В

### Check\_mail.py

```
from openai import OpenAI
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep
import vertexai
from vertexai.generative_models import GenerativeModel
import re

openai_api_key = " "

def GPT(text):
    client = OpenAI(api_key=openai_api_key)

    response = client.chat.completions.create(
        model="gpt-4",
        messages=[
            {"role": "user",
             "content": "Оціни ймовірність використання штучного інтелекту для
написання цього тексту: " + text + ".\nРезультат запиши у вигляді: 'Вірогідність:
$score', $score - значення у відсотках від 0 до 100. Також вкажи, чому ти так
вважаєш"}],
        )
    # Отримуємо вміст відповіді
    result = response.choices[0].message.content
```

```
score = re.findall(r'\d+%', result)
return result, score[0][:-1]
```

```
def Gemini(text):
```

```
    vertexai.init(project="cobalt-mind", location="europe-central2")
    model = GenerativeModel("gemini-1.0-pro-vision")
    response = model.generate_content(
        [
            "Оціни ймовірність використання штучного інтелекту для написання
цеого тексту: " + text + ".\nРезультат запиши у вигляді: 'Вірогідність: $score',
$score - значення у відсотках від 0 до 100. Також вкажи, чому ти так вважаєш"
        ]
    )
    score = re.findall(r'\d+%', response.text)
    return response.text, score[0][:-1]
```

```
def site1(text):
```

```
    # Запуск драйвера браузера
    driver = webdriver.Chrome()

    # Відкриття вебсторінки
    driver.get('https://www.scribbr.com/ai-detector/')

    # Знаходження поля для вводу і вставка тексту
    input_field = driver.find_element(By.CSS_SELECTOR,
'div[contenteditable="true"]')
    input_field.send_keys(text)

    # Знаходження кнопки та її натиск
    submit_button = driver.find_element(By.ID, "aiDetectorButton")
```

```

submit_button.click()
sleep(3)

result_element = driver.find_element(By.CSS_SELECTOR,
"span.percentage.purple.d-block.text-center.text-purple-gradient")
result_text = result_element.text

result_text = int(result_text.replace('%', ''))
return f"Ймовірність використання штучного інтелекту : {result_text}"

def site2(text):
    # Запуск драйвера браузера
    driver = webdriver.Chrome()

    # Відкриття вебсторінки
    driver.get('https://www.zerogpt.com/')

    # Знаходження поля для вводу і вставка тексту
    input_field = driver.find_element(By.ID, "textArea")
    input_field.clear()
    input_field.send_keys(text)

    # Знаходження кнопки та її натиск
    submit_button = driver.find_element(By.CLASS_NAME, "scoreButton")
    submit_button.click()
    sleep(3)

    result_elements = driver.find_elements(By.CSS_SELECTOR, 'span.header-
text.text-center[data-v-69c0294d=""]')
    if len(result_elements) > 1:

```

```
    result_text = result_elements[1].text
else:
    result_text = result_elements[0].text
```

```
res = re.search(r'(\d+)', result_text)
result = res.group(1)
```

```
return f"Ймовірність використання штучного інтелекту: {result}"
```

```
def lexical_density(text):
```

```
    words = re.findall(r'\b\w+\b', text.lower())
    total_words = len(words)
    unique_words = len(set(words))
```

```
    density = unique_words / total_words if total_words > 0 else 0
    return f"Лексична насиченість: {density:.2f}", density
```