Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: " Розробка системи для виявлення фішингових атак на основі технологій штучного інтелекту"

Виконав: студент IV курсу, групи СБ-41

Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Малик Едуард Юрійович
підпис (прізвище та ініціали)

Керівник Загородна Н.В.
підпис (прізвище та ініціали)

Нормоконтроль Дроздова Т.В.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)
за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)
Студенту Малику Едуарду Юрійовичу
(прізвище, ім'я, по батькові)
1. Тема роботи Розробка систему для виявлення фішингових атак на основі технологій штучного інтелекту

Керівник роботи к.т.н., доц. Загородна Наталія Володимирівна,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
Затверджені наказом ректора від « 02 » 05 2025 року № 4/7-361
2. Термін подання студентом завершеної роботи 17.06.2025
3. Вихідні дані до роботи Система виявлення фішингових атак

4. Зміст роботи (перелік питань, які потрібно розробити)
ВСТУП
ФІШИНГОВІ АТАКИ ТА ЇХ ОСОБЛИВОСТІ
МАТЕМАТИЧНІ МОДЕЛІ АЛГОРИТМІВ ЦИФРОВОГО ПІДПИСУ
ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ
БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ
Висновки
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Мариненко С.Ю., к.т.н., доцент кафедри МТ		

7. Дата видачі завдання 29.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		
2.	Підбір джерел на тему кваліфікаційної роботи		
3.	Науково-технічне пояснення		
4.	Проведення аналізу існуючих методів ШІ для виявлення фішингових атак		
5.	Розробка методології виявлення фішингових атак		
6.	Реалізація програмного інструменту для виявлення фішингових атак		
7.	Тестування розробленої системи		
8.	Оформлення розділу «ФІШИНГОВІ АТАКИ ТА ЇХ ОСОБЛИВОСТІ»		
9.	Оформлення розділу «МАТЕМАТИЧНІ МОДЕЛІ АЛГОРИТМІВ ЦИФРОВОГО ПІДПИСУ»		
10.	Оформлення розділу «ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ»		
11.	Оформлення кваліфікаційної роботи		
12.	Нормоконтроль		
13.	Перевірка на плагіат		
14.	Попередній захист кваліфікаційної роботи		
15.	Захист кваліфікаційної роботи		

Студент _____ Малик Е.Ю.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Загородна Н.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка системи для виявлення фішингових атак на основі технологій штучного інтелекту // Кваліфікаційна робота ОР «Бакалавр» // Малик Едуард Юрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБ-41 // Тернопіль, 2025 // С - 74 , рис. – 35, табл. – 5 –, кресл. – , додат. – 1

Ключові слова: фішинг, штучний інтелект, машинне навчання, безпека, URL-аналіз.

Кваліфікаційна робота присвячена розробці методу та програмного інструменту для виявлення фішингових атак із використанням сучасних технологій штучного інтелекту. У роботі розглянуто основні особливості фішингових атак – їх типи, способи реалізації, технічні характеристики, а також наслідки як для окремих користувачів, так і для організацій та суспільства в цілому. У рамках дослідження була розроблена власна методика аналізу URL-адрес із застосуванням алгоритмів машинного навчання для виявлення підозрілих вебресурсів. Реалізовано програмний інструмент, що дозволяє автоматизувати процес збору, обробки, аналізу та класифікації даних. Проведені тестування на реальних та штучно підготовлених вибірках показали високу ефективність розробленої системи у виявленні фішингових ресурсів.

Окремо у роботі розглянуто вимоги до апаратного та програмного забезпечення, а також умови розгортання інструменту у середовищі Windows. Важливу увагу приділено питанням безпеки життєдіяльності та основам охорони праці, що необхідні для забезпечення безпечної роботи фахівців, які працюють із такими системами.

У підсумку наведено результати проведеного дослідження, сформульовано висновки та пропозиції щодо удосконалення створеної системи, а також можливості її застосування на практиці. Робота може стати у пригоді

спеціалістам з інформаційної безпеки, розробникам захисних систем та дослідникам, які займаються використанням штучного інтелекту для посилення кіберзахисту.

ABSTRACT

Development of a System for Detecting Phishing Attacks Based on Artificial Intelligence Technologies // Bachelor's Qualification Thesis // Malyk Eduard // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, Group СБ-41 // Ternopil, 2025 // p. 74 , fig. – 35 , tables – 5 , drawings – , appendices – 1.

Keywords: phishing, AI, machine learning, safety, URL analysis.

This bachelor's qualification paper focuses on the development of a method and software tool for detecting phishing attacks using modern ai technologies. The work examines the main advantages of phishing attacks — their types, methods of execution, technical characteristics, and the consequences for individual users, organizations, and society as a whole.

As part of the research, a custom methodology for analyzing URLs was developed, applying machine learning algorithms to identify suspicious web resources. A software tool was implemented to automate the collection, preprocessing, analysis, and classification of data. Testing conducted on real and artificially generated datasets demonstrated the high effectiveness of the developed system in detecting phishing resources.

The paper also addresses hardware and software requirements, as well as deployment conditions in a Windows environment. Special attention is given to occupational safety and health issues, which are essential for ensuring the secure work of cybersecurity specialists operating such systems.

The conclusions summarize the research results and provide suggestions for improving the developed system, expanding its functionality, and applying it in practice. This work can be useful for information security specialists, developers of protective systems, and researchers engaged in applying artificial intelligence to strengthen cybersecurity.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI — artificial intelligence

ML — machine learning

URL — uniform resource locator

HTTP — hypertext transfer protocol

HTTPS — hypertext transfer protocol secure

DNS — domain name system

IP — internet protocol

CSV — comma-separated values

API — application programming interface

GUI — graphical user interface

SSL — secure sockets layer

TLS — transport layer security

HTML — hypertext markup language

JSON — javascript object notation

SSH — secure shell

PHISHING — фішинг

BLACKLISTING — чорний список

WHITELISTING — білий список

ВСТУП

Актуальність теми

Сучасний розвиток цифрових технологій і збільшення кількості користувачів інтернету призводять до того, що кіберзагрози, зокрема фішингові атаки, набувають все більшого масштабу й складності. Фішинг став одним з основних методів соціальної інженерії, що використовуються зловмисниками для викрадення конфіденційних даних користувачів, таких як паролі, банківські реквізити та особисті відомості. Традиційні способи захисту, зокрема фільтрація контенту та чорні списки, часто не встигають за швидкоплинними атаками. У цьому контексті впровадження технологій штучного інтелекту та машинного навчання представляється перспективним шляхом для підвищення ефективності виявлення й блокування фішингових ресурсів.

Мета і завдання дослідження.

Метою даного дослідження є розробка методу та програмного забезпечення для виявлення фішингових атак, ґрунтуючись на алгоритмах машинного навчання. Для досягнення цієї мети були визначені такі задачі:

- Провести аналіз різновидів і характеристик фішингових атак.
- Дослідити сучасні способи захисту від фішингу.
- Розробити методику аналізу url-адрес за допомогою алгоритмів машинного навчання.
- Зреалізувати програмний інструмент, який автоматизує аналіз та класифікацію даних.
- Провести тестування розробленої системи на різноманітних вибірках.
- Сформувати рекомендації щодо покращення та розширення функціональності системи.

Об'єкт дослідження.

Процес виявлення фішингових атак в інформаційно-комунікаційних мережах.

Предмет дослідження.

Методи, алгоритми та програмні засоби виявлення фішингових атак, що ґрунтуються на аналізі url-адрес та використанні технологій машинного навчання.

Практичне значення отриманих результатів.

Розроблений метод та програмний інструмент можуть бути застосовані для підвищення рівня кібербезпеки в організаціях та окремих інформаційних системах. Одержані результати мають практичну цінність для впровадження у службах безпеки, центрах обробки даних, а також у процесі розробки антивірусного й антифішингового програмного забезпечення. Крім того, створений інструмент може стати основою для подальших досліджень і вдосконалення систем кіберзахисту на основі штучного інтелекту.

РОЗДІЛ 1 ФІШИНГОВІ АТАКИ ТА ЇХ ОСОБЛИВОСТІ

1.1 Актуальність роботи

Фішингові атаки продовжують бути одним із найчастіших та найнебезпечніших видів кіберзлочинів. Вони створюють суттєву загрозу як для простих користувачів, так і для великих підприємств, оскільки дають зловмисникам можливість отримати доступ до чутливих даних, фінансової інформації та облікових записів. З удосконаленням інформаційних технологій фішинг стає дедалі більш витонченим – кіберзлочинці використовують соціальну інженерію, фальшиві веб-сайти, електронні листи та навіть SMS і месенджери для обману жертв.

Особливої актуальності ця тема набула в умовах військового конфлікту між Україною та російською федерацією, оскільки фішингові кампанії все частіше застосовуються як інструмент інформаційно-психологічного впливу, збору розвідувальних даних, атак на державні органи та критичну інфраструктуру. Від початку повномасштабного вторгнення кількість кіберінцидентів в Україні значно збільшилась. Згідно з аналізом Державної служби спеціального зв'язку та захисту інформації України (Держспецзв'язок) та інших дослідницьких організацій, фішинг залишається однією з найпоширеніших форм кібератак в Україні. Протягом 2024 року було зафіксовано 4315 кіберінцидентів, що на 69,8% більше, ніж у 2023 році, коли було зареєстровано 2541 атаку. Основні типи атак були трьох категорій:

- фішингові атаки;
- спроби використання вразливостей;
- з'єднання зі шкідливим кодом.

Традиційні методи захисту від фішингу, засновані на аналізі сигнатур або ручному моделюванні загроз, більше не забезпечують необхідного рівня захисту. У зв'язку з цим виникає потреба у використанні сучасних технологій, таких як штучний інтелект (ШІ) та машинне навчання (МН), які дозволяють

автоматизувати процес виявлення фішингових ресурсів, оперативно реагувати на загрози та зменшувати ймовірність хибних спрацьовувань.

За повідомленнями, основними цілями російських хакерів були:

• органи державної та місцевої влади;

• інформаційні ресурси сектору безпеки та оборони;

• енергетичний сектор;

• фінансовий сектор;

• комерційний сектор;

• телекомунікаційний сектор та розробники програмного забезпечення.

У 2023 році в Україні спостерігалося значне зростання кіберзлочинності, зокрема у сфері шахрайства з платіжними системами. За даними Української міжбанківської асоціації членів платіжних систем «ЕМА», загальна сума збитків, завданих громадянам, перевищила 3 мільярди гривень, що втричі більше, ніж у попередньому році.

Ключові тенденції та загрози включають:

• Соціальна інженерія: схеми соціальної інженерії залишаються найпоширенішим методом шахрайства. Середні втрати на одну транзакцію зросли до 8500 гривень, що на 20% більше, ніж у 2022 році.

• Фішингові атаки: значне збільшення кількості фішингових веб-сайтів, спрямованих на українських користувачів. У 2022 році було заблоковано 568 таких ресурсів, що у 2,5 рази більше, ніж у 2021 році.

• Фейкові додатки та чат-боти: збільшення кількості шахрайських мобільних додатків та чат-ботів, які маскуються під офіційні сервіси для надання соціальної допомоги або продажу товарів за зниженими цінами, з метою збору особистої інформації та даних облікових записів користувачів.

• Грошові мули: значне збільшення кількості осіб, які надають свої банківські рахунки шахраям за грошову винагороду. У 2023 році було виявлено понад 10 800 таких випадків, що в шість разів більше, ніж у 2021 році.

За даними Української міжбанківської асоціації учасників платіжних систем, як показано на рисунку 1.1, кіберзлочинці найчастіше використовують назви відомих українських брендів та сервісів для крадіжки конфіденційної інформації.

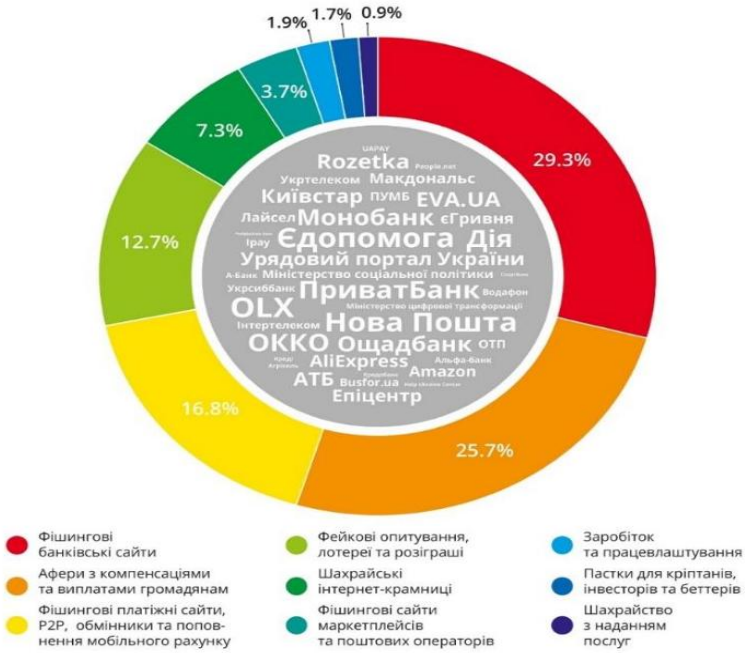


Рисунок 1.1 – Бренди, що використовувалися під час фішингових атак

На рисунку 1.2 представлено аналіз походження реєстраторів фішингових веб-сайтів. Найбільша частка припадає на реєстраторів зі Сполучених Штатів Америки (49,1%) та Росії (23,6%). Це корелює із загальними світовими тенденціями кіберзлочинності. Значна частина шахрайських ресурсів (57,8%) використовує сервіс зворотного проксі Cloudflare, розроблений у США, для захисту веб-ресурсів. У той час як легітимні компанії використовують його для захисту від атак, зловмисники вдаються до цього інструменту, щоб приховати реальну адресу своїх сайтів та ускладнити їх блокування.

Відповідно, актуальність цього дослідження зумовлена необхідністю підвищення ефективності кіберзахисту шляхом розробки інтелектуальної системи виявлення фішингових атак. Така система повинна мати здатність адаптуватися до нових загроз та забезпечувати надійний захист користувачів у цифровому середовищі.

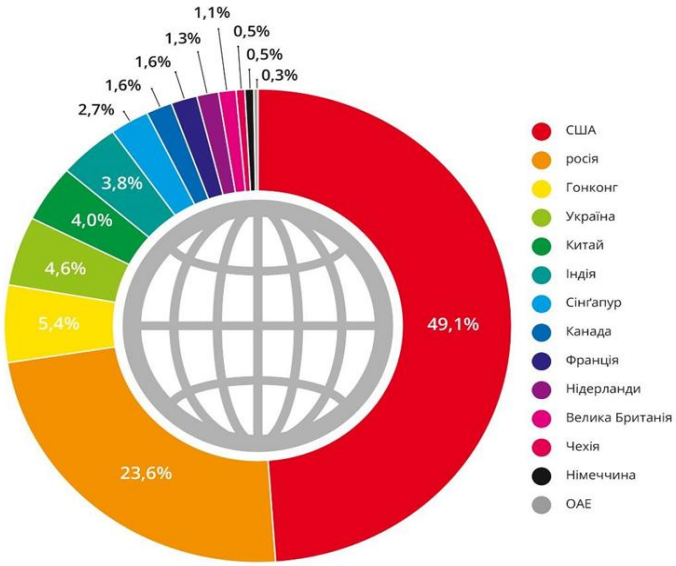


Рисунок 1.2 – Розподіл за походженням реєстраторів фішингових сайтів

У цьому розділі було розглянуто актуальність фішингових атак як одного з найпоширеніших інструментів кіберзлочинності в сучасному цифровому середовищі. Фішинг продовжує залишатися серйозною загрозою як для окремих користувачів, так і для організацій, особливо в умовах збройного конфлікту між Україною та російською федерацією, де кіберзброя дедалі активніше використовується в інформаційній війні.

Наводячи аналітичні дані за останні роки, було показано, що кількість фішингових атак суттєво зростає, що вимагає посиленої уваги до методів їх виявлення та запобігання. Звичайні засоби боротьби з фішингом вже не забезпечують достатнього рівня захисту, тому постає необхідність у впровадженні сучасних підходів – зокрема, використанні методів штучного інтелекту та машинного навчання.

З урахуванням статистичних даних про вектори атак, популярні платформи для розповсюдження фішингових посилань, а також приклади зловживань авторитетом державних і комерційних брендів, можна зробити висновок, що ефективна боротьба з фішингом потребує комплексного підходу. Це має включати не лише технічні засоби, але й підвищення обізнаності користувачів, державне регулювання, міжнародну співпрацю та постійне оновлення систем виявлення загроз.

Таким чином, проблема фішингу є вкрай актуальною як на глобальному рівні, так і в контексті українських реалій. Саме тому дослідження інтелектуальних методів виявлення фішингових атак є важливим кроком до підвищення національної кібербезпеки.

1.2 Поняття та види фішингових атак

Поняття «фішинг» розкривається перед нами крізь призму думок експертів, науковців та фахівців з кібербезпеки. Через постійну еволюцію тактик атак, визначення цього явища в єдиному та загальноприйнятому вигляді є складним завданням. Однак більшість інтерпретацій сходяться на тому, що фішинг – це спроба спонукати жертву до дій, вигідних для зловмисників. Це може служити відправною точкою для розуміння сутності фішингових схем.

У деяких випадках фішинг обмежується атаками з використанням підроблених веб-ресурсів. Коротко кажучи, фішинг – це вид шахрайства, заснований на імітації реального сайту з метою введення користувача в оману та отримання доступу до його конфіденційних даних: особистої інформації, фінансової інформації або паролів для авторизації.

Головна мета фішингової атаки полягає у отриманні від жертви особистої інформації: логінів, паролів, банківських реквізитів, номерів кредитних карток та інших даних. Зловмисники нерідко вдаються до методів соціальної інженерії, щоб викликати в жертви довіру або змусити її прийняти рішення поспіхом.

Відповідно до даних, викладених у розділі з кібербезпеки на офіційному сайті Microsoft, можна виділити шість ключових типів фішингових атак (див. рисунок 1.3).

Рисунок 1.3 – Види фішингу за класифікацією Microsoft

Фішинг через електронну пошту – один з найпоширеніших методів обману, що передбачає розсилку підроблених листів із прихованими гіперпосиланнями, які ведуть на шахрайські ресурси. Мета таких листів – спонукати адресата розкрити особисті дані. Зловмисники часто маскуються під відомі компанії: Microsoft, Google або сервіси банківських установ, електронної комерції та хостингу.

Фішинг із використанням шкідливого ПЗ. Цей тип атаки полягає у надсиланні користувачеві зловмисних файлів, замаскованих під нібито безпечні документи, наприклад, резюме або виписки з банку. Відкриття таких вкладень може призвести до зараження комп'ютера або порушення функціонування всієї мережі.

Цільовий фішинг. На відміну від масових атак, цей метод спрямований на конкретних осіб або організації, про які зловмисники вже зібрали інформацію. Такі фішингові листи, як правило, надзвичайно переконливі та можуть обходити стандартні засоби кіберзахисту.

Атаки на "велику здобич". В цьому випадку об'єктами фішингу стають впливові особи: керівники компаній, бізнесмени, політики чи публічні діячі. Атакуючі ретельно вивчають свої цілі, щоб у потрібний момент отримати доступ до важливої або конфіденційної інформації.

Смс-фішинг. Метод, що базується на розсиланні підроблених текстових повідомлень (SMS), які імітують повідомлення від відомих брендів, наприклад, "Нова Пошта" або "Rozetka". Через простоту та персоналізований підхід ці повідомлення виглядають правдоподібно, що збільшує шанси на ошукування користувачів.

Вішинг. Атака, що здійснюється телефоном. Зловмисники представляються працівниками банків, служб підтримки або інших офіційних структур, щоб переконати жертву повідомити важливу інформацію або встановити шкідливе програмне забезпечення. У таких випадках активно використовуються методи соціальної інженерії.

Розгляньмо докладніше одну з ключових схем фішингових атак. Незмінною деталлю таких атак є фішинговий веб-сайт, який і є головним знаряддям для збору секретної інформації користувача. Один з найпростіших способів реалізації атаки передбачає використання проксі-сервера, що управляється зловмисником. Схематичне зображення цього процесу наведено на рисунку 1.4.

Рисунок 1.4 – Структура найпростішої фішинг-атаки

Механіка атаки розгортається так: жертва отримує фальшиве посилання, приховане під виглядом адреси надійного ресурсу – скажімо, відомого інтернет-магазину чи онлайн-сервісу. Клікнувши на це посилання, користувач

переноситься на фішинговий проксі-сервер, котрий візуально дублює дизайн справжнього веб-сайту.

На сторінці, як правило, розміщено форму входу чи реєстрації, де вимагається ввести особисті або фінансові відомості. Введена інформація одразу зберігається зловмисником. Після цього жертву можуть автоматично переадресувати на справжній сайт, де з'являється повідомлення про помилку авторизації або інша технічна неполадка. Це формує уявлення про те, що проблема виникла випадково, а сам користувач часто навіть не здогадується, що його дані вже перехоплені. Розглянемо класичний приклад фішингу. Його узагальнена схема наведена на рисунку 1.5.

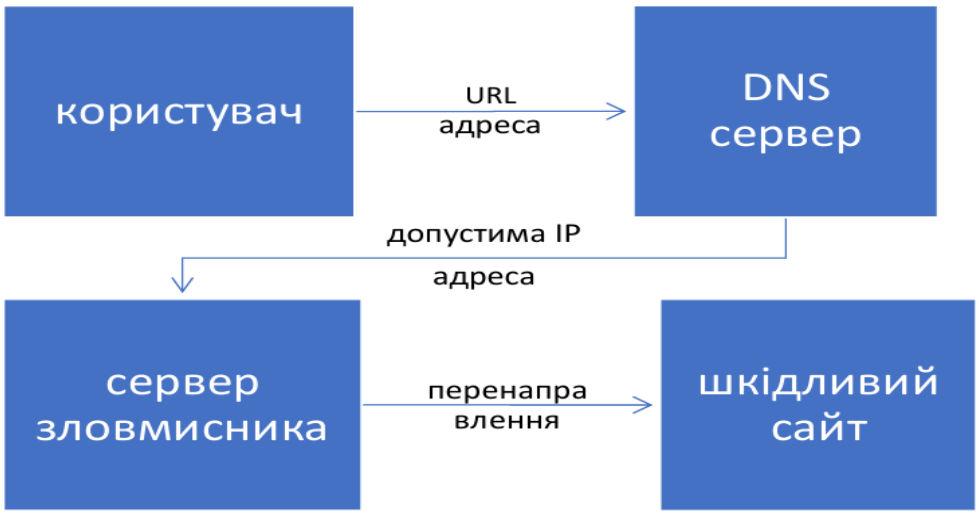


Рисунок 1.5 – Традиційний вид атаки

Розглянемо просунуту техніку фішингу – стратегію rock-phish (див. рисунок 1.6).

Цей метод дозволяє зловмисникам значно знизити ймовірність швидкого виявлення системами кіберзахисту, зберігаючи при цьому високу ефективність атаки завдяки складній інфраструктурі. Ключовою особливістю rock-phish є використання розгалуженої мережі проміжних ланок, таких як проксі-боти, які значно ускладнюють та затримують процес блокування фішингових ресурсів.

На відміну від звичайних атак з використанням прямого проксі-сервера, ця схема використовує ботнет – групу заражених пристроїв, заражених спеціальним

шкідливим програмним забезпеченням. Ці пристрої, не підозрюючи про свою роль у фішинговій кампанії, служать проміжною ланкою між користувачем та сервером зловмисника. Це дозволяє не тільки продовжувати існування фішингового сайту, але й ускладнювати та знижувати ефективність його блокування з технічної точки зору.

Таким чином, стратегія rock-phish є прикладом динамічного та адаптивного підходу до реалізації фішингових загроз, демонструючи високий рівень організованості з боку кіберзлочинців.

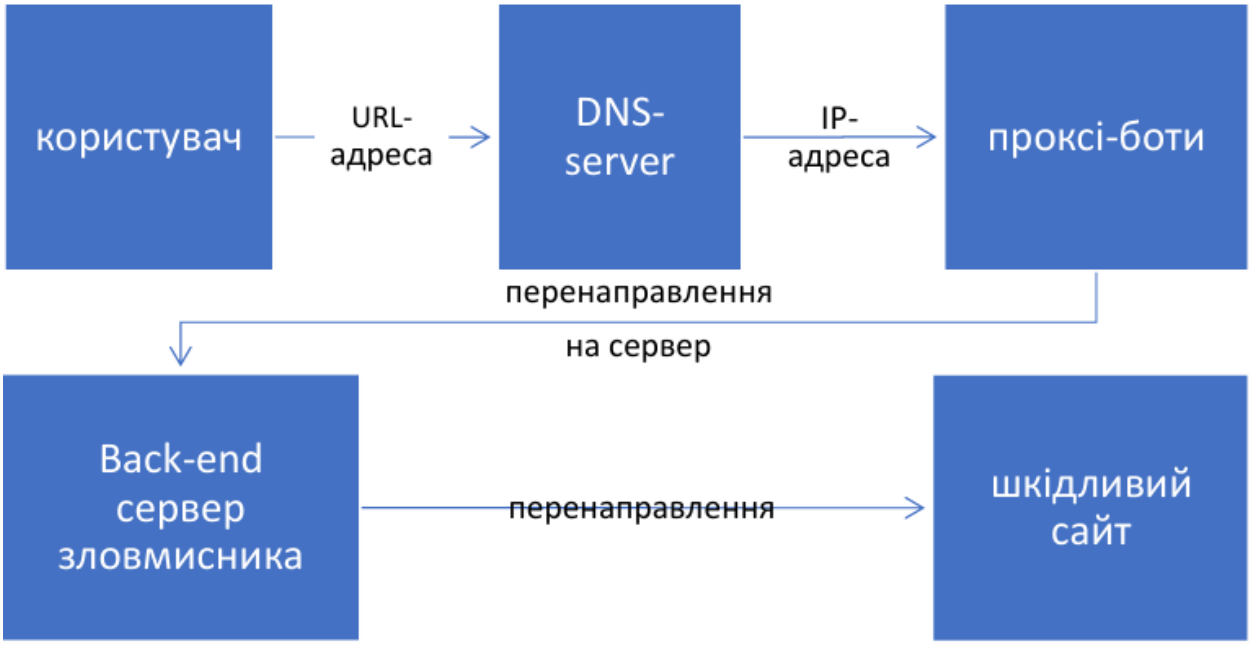


Рисунок 1.6 – Стратегія Rock-phish

Одним із найсерйозніших негативних наслідків Rock-phish-атак є не лише потенційна крадіжка персональних даних чи фінансів. Під час взаємодії з такими фішинговими ресурсами пристрій користувача ризикує бути зараженим шкідливим програмним забезпеченням, що перетворить його на невід'ємну частину ботнету – мережі комп'ютерів, що використовуються для подальших кібератак, без інформування їхніх власників.

Наступний етап еволюції цих загроз демонструє стратегія Fast-Flux, яка візуалізована на рисунку 1.7. Ця методика базується на складному механізмі

приховування джерела шкідливого трафіку, який полягає в динамічній зміні DNS-записів.

Використовуючи можливості Fast-Flux DNS, зловмисники отримують можливість швидко змінювати IP-адреси, пов'язані з певним доменом. Це дозволяє їм створювати розгалужену мережу взаємозамінних ботів, через яких забезпечується доступ до фішингового сайту або шкідливих сервісів. Така організація значно ускладнює процес відстеження фактичного місцезнаходження сервера управління, а також унеможливлює швидке блокування ресурсу правоохоронними органами чи аналітичними підрозділами.

Таким чином, Fast-Flux – це ефективна технологія обходу традиційних методів кіберзахисту, яка забезпечує високу стабільність фішингової інфраструктури та гнучкість в її управлінні.

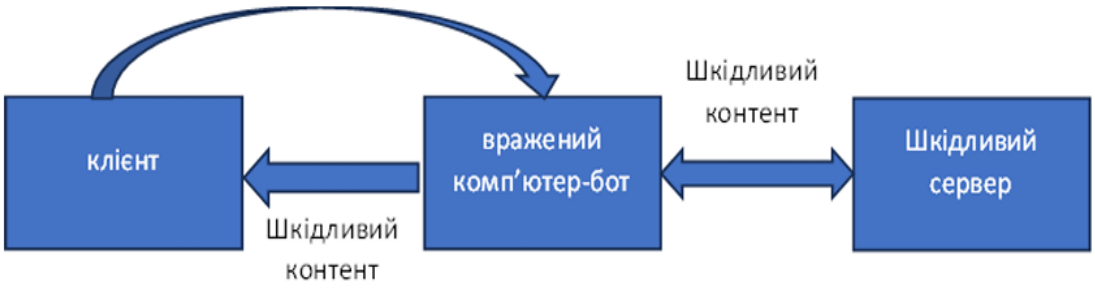


Рисунок 1.7 – Стратегія фішингової атаки fast-flux

Fast flux DNS – це одна з найхитромудріших технік, які використовують зловмисники, аби сховати справжнє розташування своїх серверів. В основі цього лежить цілком законна функціональність системи доменних імен, що дає можливість одному домену відповідати кількох IP-адресам одночасно. Зазвичай це застосовується для розподілу навантаження на сервери. Але саме в цьому зловмисники розгледіли зручний метод для маніпуляцій.

На відміну від звичної практики, де IP-адреси залишаються незмінними, у fast flux вони змінюються миттєво – буквально кожні декілька хвилин. Така висока частота змін досягається через встановлення вкрай малого TTL (time to live) у DNS-записах. В результаті – коли жертва переходить на фішинговий сайт,

насправді вона може потрапляти на різні сервери, що ускладнює виявлення та блокування традиційними засобами захисту.

Оскільки ця схема потребує співпраці з реєстраторами доменів, які часто не звертають уваги на зловживання або просто не мають можливості їх виявляти, переважна частина fast flux-інфраструктур реєструється в юрисдикціях зі слабким кіберзаконодавством – зазвичай у країнах, що розвиваються. Це дозволяє зловмисникам діяти майже безкарно.

У свою чергу, загальна структура фішингової атаки, представлена на рисунку 1.8, лише підкреслює складність і продуманість цих загроз. Від самого початку, коли зловмисник лише збирає інформацію про майбутню жертву – її поведінку, уподобання, професійну сферу, – до моменту, коли дані вже скомпрометовано та використовується, все функціонує як добре налагоджений механізм. Кожна деталь ретельно продумана: вибір способу атаки, маскування, і навіть психологічні прийоми, що змушують людину втрачати пильність.

Фішинг більше не виглядає як просте підроблення листа з проханням "надішліть пароль". Це складна багатоступенева кампанія, де за кожною дією криється стратегія, за кожною сторінкою – аналіз, а за кожною атакою – чітке розуміння людської психології та слабкостей. І найбільша небезпека полягає в тому, що зловмисники майже завжди на крок попереду.

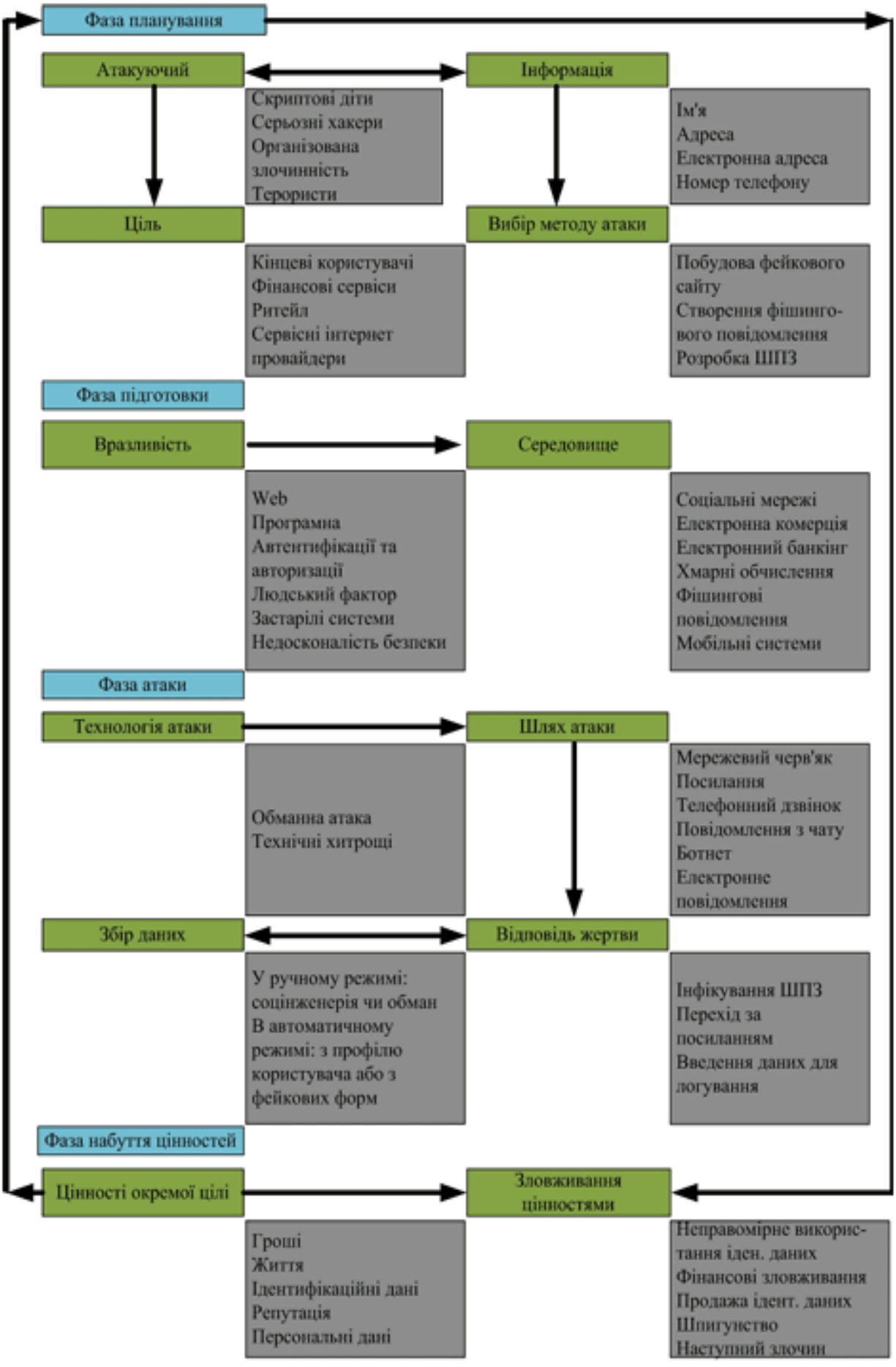


Рисунок 1.8 – Фази фішингової атаки

Ця структура фішингової атаки дає можливість глибше зануритися у суть проблеми, а не обмежуватися лише тим, що "користувач помилково клацнув". Вона демонструє, що фішинг – це не випадкова помилка, а ретельно спланована стратегія, яка включає в себе різні етапи, від збору інформації до використання вкрадених даних. І саме усвідомлення цього циклу атаки відкриває перспективи для зупинення її не лише на завершальному етапі, коли вже запізно, а ще до того, як вона досягне кінцевого споживача.

Більше того, цей підхід ставить під сумнів розповсюджене переконання, що користувач – головний та єдиний чинник проблеми. По суті, якщо фішинговий лист потрапив до скриньки отримувача, це означає, що щось дало збій на етапі технічного захисту. Відтак, задача розробників полягає не лише в тому, щоб навчити користувачів "не натискати на підозрілі посилання", а й створити системи, які мінімізують ймовірність того, що ці посилання взагалі з'являться на екрані.

Захист має починатися на ранніх етапах – на рівні мереж, хостинг-провайдерів, поштових фільтрів, DNS та веб-браузерів. Слід враховувати, що кожна атака проходить декілька рівнів, перш ніж досягне цілі: вона використовує недоліки протоколів, експлуатує вразливості програмного забезпечення та маскується під виглядом соціальної інженерії. Саме тому боротьба з фішингом не повинна зводитися лише до рекомендацій для кінцевих користувачів – вона повинна охоплювати всю інфраструктуру, через яку потенційна загроза може проникнути. І чим більше зусиль буде спрямовано на запобігання проникненню, тим менше відповідальності покладатиметься на того, хто, зрештою, просто відкрив електронний лист.

1.3 Системи виявлення та запобігання фішингу

Фішинг, як явище, глибоко вкоренився у світовому цифровому просторі. Його історія бере початок ще з 1990-х, але сьогодні він надзвичайно актуальний та постійно еволюціонує. Протидія цьому виду шахрайства не обмежується лише

розробкою політичних і навчальних інструментів для користувачів, а й включає технічні, а також запатентовані рішення.

Вивчення патентної бази показує сталий інтерес до автоматизованих систем боротьби з фішингом у всьому світі. Наприклад, патент US2021099484A1 пропонує новаторський спосіб ідентифікації фішингових ресурсів. Замість звичайних маркерів у коді чи URL, використовується візуальний аналіз – "відбиток" веб-сторінки. Браузер робить "знімок" сторінки, перетворюючи його у хеш, який порівнюється з базою відомих фішингових шаблонів. Цей підхід нагадує людське сприйняття: ми розпізнаємо сайт за його зовнішнім виглядом, а не за IP-адресою.

Інші патенти, такі як US2023344866, зосереджуються на мережевій поведінці, свідчачи про зміщення акцентів у бік поведінкового аналізу. У світі, де фішинг легко маскується під легальні сервіси, поведінка користувача та мережева активність стають важливішими індикаторами загроз, ніж доменне ім'я або електронна адреса.

Ці підходи доповнюються перевірками електронної пошти (US2023291767), де акцент зміщено з оцінки "підозрілості тексту" на перевірку справжності домену відправника. Тут реалізується технічна перевірка ідентифікаторів, вбудованих у протоколи e-mail.

Ще один патент – US2023344868 – комбінує ідеї: використання хеш-відбитків веб-сторінок разом з перевіркою IP-адрес, що дозволяє швидко реагувати на спроби використання нетипових шляхів доставки шкідливого контенту.

Ці патенти демонструють не лише зростаючу складність атак, але й розуміння необхідності багатошарового захисту. Сучасна система має бути багатоаспектною, гнучкою та здатною до адаптації. Компанія Cisco реалізувала саме таку стратегію у вигляді продукту Cisco Email Security, котрий є цілісним програмно-апаратним рішенням, що не просто перевіряє вхідну пошту, а проводить глибокий аналіз, враховуючи шаблони, політики та взаємозв’язки.

Ці розробки – це не лише відповідь на минулі виклики. Це спроба випередити розвиток фішингу, який змінюється швидше, ніж оновлюються класичні антивірусні програми. Поки користувачі клікають на листи, що виглядають правдоподібно, єдине, що може зупинити атаку – це розумна, своєчасна автоматична система, яка бачить те, що не помічаємо ми.

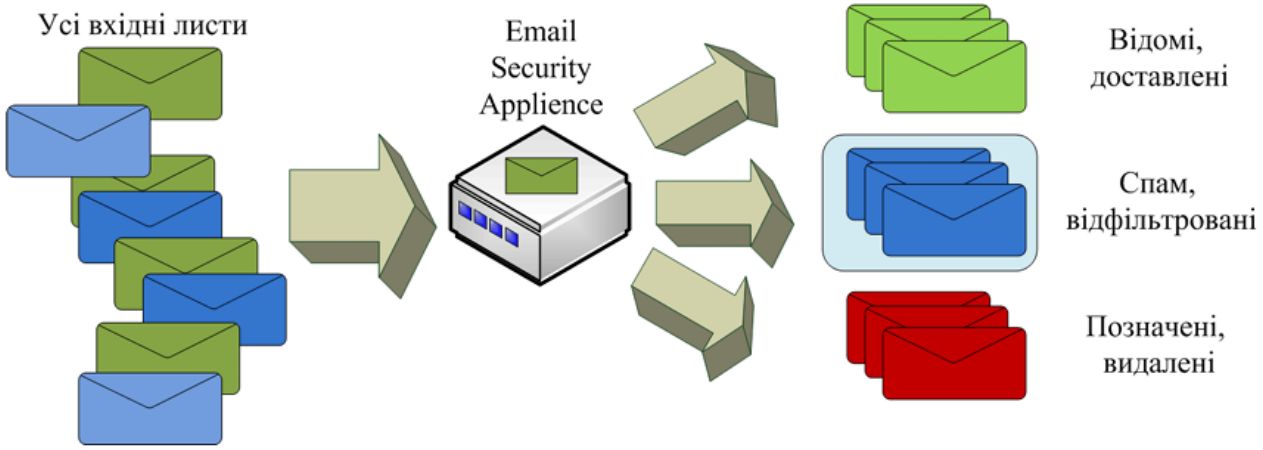


Рисунок 1.9 – Механізм роботи системи безпеки електронної пошти Cisco

Дана система на рисунку 1.9 (Cisco Email Security Appliance – CESA) – це цілісний комплекс, який поєднує набір інструментів та технологій. Їх мета – виявляти, блокувати й аналізувати фішингові атаки та пов’язані з ними небезпеки. Функціонал базується не лише на статичних фільтрах, але й на підході, що адаптується та реагує на загрози в режимі реального часу.

Однією з ключових особливостей є використання глобального інтелекту щодо загроз. Платформа Cisco Talos постійно відстежує інтернет-трафік, виявляючи нові шаблони поведінки та аномалії. Перевага в тому, що нові правила безпеки, розроблені на основі цього аналізу, надходять до системи практично миттєво, з оновленнями кожні 3-5 хвилин. Це означає, що користувачі захищені не тільки від відомих атак, але й від тих, що тільки починають розповсюджуватися.

Значну частку загроз складає спам, і тут CESA застосовує багатошарову фільтрацію: зовнішню, засновану на репутації відправника, та внутрішню, яка аналізує зміст. Це дозволяє блокувати більшість непотрібної пошти ще до того,

як вона досягне вхідних повідомлень користувача. Аналіз враховує тему, вміст, відправника та інші фактори.

Ще одним важливим компонентом є фільтрування так званої graymail (масових розсилок, які технічно не є спамом, але можуть дратувати або відволікати). Тут система дозволяє безпечно керувати такими повідомленнями та генерувати звіти.

Коли йдеться про справжні загрози, такі як шкідливе програмне забезпечення, CESA не обмежується базовим антивірусним аналізом. Використовується комбінований підхід: перевірка репутації файлів, запуск у "пісочниці" для аналізу поведінки та ретроспективне відстеження загроз навіть після доставки файлу. Це важливо, бо деяке шкідливе ПЗ може активуватися із запізненням, і лише в такому режимі можна визначити його справжню сутність.

Мережеві фільтри дозволяють блокувати складні атаки, які поєднують кілька способів зараження. Додатково, інтерактивне веб-відстеження дозволяє адміністраторам бачити, які вебсайти відвідують користувачі та чи не потрапляють вони на небезпечні ресурси.

Окремої уваги заслуговує контроль вихідної пошти. CESA дозволяє виявляти витоки конфіденційної інформації за допомогою модулів DLP (Data Loss Prevention), забезпечувати шифрування повідомлень та інтеграцію з іншими корпоративними системами безпеки, наприклад, RSA Enterprise Manager. Це гарантує, що навіть повідомлення, відправлені ззовні, не порушують вимоги безпеки організації.

Загалом, CESA – це потужне та багатофункціональне рішення. Водночас, як і будь-яка система, вона не є абсолютно захищеною – кіберзагрози швидко розвиваються, і жоден продукт не може дати стовідсоткової гарантії. Але використання таких систем значно зменшує вірогідність успішної атаки та дозволяє організаціям залишатися в безпеці навіть при появі нових, ще недостатньо вивчених загроз.

1.4 Постановка задачі

Огляд, здійснений у цьому дослідженні, демонструє, що проблема кібербезпекових загроз, зокрема фішингу, зберігає високу актуальність, хоча існує значний арсенал методів протидії. Зловмисники безперестанку поліпшують свої техніки, розробляючи нові шляхи викрадення секретних даних. Під загрозою стають навіть професіонали у сфері кібербезпеки, що лише акцентує масштабність та важкість проблеми. Відтак, створення нових методик виявлення фішингових атак залишається критичним та багатообіцяючим напрямком наукових досліджень.

Відповідно до поставленої мети, в рамках цієї кваліфікаційної роботи планується реалізація наступних задач:

- Проаналізувати сучасні стратегії протидії фішингу, застосовуючи технології штучного інтелекту.
- Вдосконалити вже наявний спосіб виявлення фішингових атак через інтеграцію інструментів штучного інтелекту.
- Провести експериментальне випробування різних класифікаторів на основі тестових прикладів, здійснити порівняльний аналіз отриманих результатів та визначити найбільш ефективне рішення.

2 РОЗРОБКА СИСТЕМИ ВИЯВЛЕННЯ ФІШИНГОВИХ АТАК

2.1 Огляд технологій штучного інтелекту

У сучасну епоху інформаційні технології розвиваються з надзвичайною швидкістю, що призводить до збільшення як кількості, так і складності кіберзагроз. Це, у свою чергу, вимагає вдосконалення механізмів інформаційної безпеки. Останні роки характеризуються значним зростанням кібератак, таких як фішинг, шкідливе програмне забезпечення та атаки з використанням методів соціальної інженерії. Кіберзлочинці використовують дедалі складніші методи, що вимагає впровадження передових технологій для забезпечення надійного захисту даних. Штучний інтелект (ШІ) став потужним інструментом як для посилення кібербезпеки, так і для планування та реалізації кібератак. Технології ШІ, зокрема машинне навчання та глибоке навчання, досягли значного прогресу та знайшли широке застосування в різних галузях, включаючи кібербезпеку. Штучний інтелект здатний обробляти значні масиви інформації та виявляти невидимі на перший погляд взаємозв’язки, що робить його важливим інструментом у протидії кіберзагрозам.

Постановка проблеми. Зростання кількості та складності кіберзагроз у сучасному світі створює нагальну потребу в удосконаленні методів захисту інформації. Сучасні кіберзлочинці використовують дедалі складніші та витонченіші методи, що диктує необхідність впровадження передових технологій для забезпечення ефективного захисту інформаційних систем. Тому існує потреба у вивченні потенціалу використання ШІ для підвищення кібербезпеки.

У контексті зростаючих кіберзагроз вивчення методів і підходів до кібербезпеки із залученням штучного інтелекту набуває особливого значення та актуальності. Використання технологій штучного інтелекту дозволяє значно підвищити ефективність захисту інформації, забезпечуючи оперативне виявлення та нейтралізацію кіберзагроз. Існує нагальна потреба в проведенні

нових наукових досліджень, здатних забезпечити більш ефективні рішення, що відповідатимуть актуальним викликам у сфері кібербезпеки. Тому дослідження в цій галузі мають першорядне значення через низку ключових факторів:

- З кожним роком кіберзагрози стають дедалі складнішими та масштабнішими, що знижує ефективність традиційних методів захисту. У таких умовах штучний інтелект і машинне навчання відіграють ключову роль, оскільки здатні швидко аналізувати великі обсяги даних, виявляти складні закономірності та оперативно адаптуватися до нових типів атак.
- Штучний інтелект дає змогу автоматизувати численні процеси в сфері кібербезпеки – від ідентифікації загроз до оперативного реагування на інциденти, що суттєво підвищує ефективність і швидкість дій, знижуючи навантаження на персонал та мінімізуючи ризик людських помилок.
- Застосування аналізу аномальної поведінки в режимі реального часу сприяє більш точному та оперативному виявленню загроз, що дає змогу зменшити кількість помилкових спрацьовувань і підвищити загальний рівень кібербезпеки.
- Системи зі штучним інтелектом мають здатність до безперервного навчання та адаптації, що забезпечує їхню релевантність і ефективність у динамічному середовищі кібербезпеки, яке постійно змінюється.
- Поєднання штучного інтелекту з передовими технологіями, зокрема Інтернетом речей і великими даними, створює нові перспективи для зміцнення кібербезпеки. Така синергія забезпечує глибший аналіз інформації та сприяє формуванню більш всеосяжних і ефективних стратегій захисту.

Дослідження у сфері застосування штучного інтелекту в кібербезпеці охоплюють також критично важливі етичні та правові аспекти, зокрема питання конфіденційності, прозорості прийняття рішень і відповідальності за дії, вчинені на основі рішень ІІ-систем. Вивчення цієї теми відкриває нові горизонти для стратегічного розвитку та інновацій у галузі кіберзахисту, надаючи фахівцям новітні підходи та засоби для забезпечення стійкості інформаційних систем. В умовах стрімкої еволюції кіберзагроз дослідження ролі ШІ є не лише

актуальним, а й критично необхідним – адже воно підвищує загальну ефективність захисту та сприяє створенню сучасних стратегій боротьби з цифровими загрозами.

Сучасна статистика свідчить про стрімке розширення ринку рішень на основі штучного інтелекту у сфері кібербезпеки. Згідно з даними MarketsandMarkets, у період з 2019 по 2026 рік прогнозується щорічне зростання цього ринку на 23,3%, – від $8,8 млрд до $38,2 млрд. Така динаміка підкреслює зростаючу зацікавленість у використанні ШІ для підвищення ефективності та надійності захисту інформаційних систем.

Згідно з даними звіту IBM X-Force Threat Intelligence Index 2023, технології штучного інтелекту та машинного навчання були задіяні у 30% усіх кіберінцидентів. Водночас компанії, що впроваджують ШІ у свої системи безпеки, досягли скорочення часу на виявлення й реагування на інциденти на 27%.

Аналітики Gartner прогнозують, що до 2025 року 60% організацій активно застосовуватимуть технології штучного інтелекту для виявлення й реагування на кіберзагрози. Згідно з їх висновками, впровадження ШІ у стратегії кібербезпеки суттєво підвищує ефективність і точність виявлення загроз.

За даними Statista, у 2022 році обсяг інвестицій у технології штучного інтелекту для забезпечення кібербезпеки становив близько $12,5 млрд, а до 2025 року прогнозується його зростання до $25 млрд. Така динаміка свідчить про зростання значущості й актуальності ШІ як ключового інструменту в галузі кіберзахисту.

Національний інститут стандартів і технологій США (NIST) приділяє значну увагу дослідженню можливостей застосування штучного інтелекту у сфері кібербезпеки. Організація розробляє рекомендації та стандарти, які сприяють впровадженню ефективних стратегій захисту із залученням ІІ. Особливий акцент робиться на інтеграцію штучного інтелекту в системи виявлення та реагування на загрози з метою підвищення загального рівня інформаційної безпеки.

Отже, аналіз даних з авторитетних джерел підтверджує значний потенціал та актуальність інтеграції технологій штучного інтелекту у сферу кібербезпеки. Завдяки можливостям автоматизації процесів моніторингу, виявлення та реагування на інциденти, ШІ сприяє скороченню часу реагування та зміцненню загального захисту інформаційних систем. Динамічне зростання інвестицій у цю сферу ще більше підкреслює її стратегічне значення у боротьбі з кіберзагрозами. Сучасні підходи – машинне та глибоке навчання, а також аналіз великих даних – ефективно впроваджуються у системи кіберзахисту для забезпечення оперативності та точності реагування. Водночас залишаються актуальними і низка викликів, зокрема: забезпечення прозорості рішень, ухвалених ШІ, та захист від потенційного зловживання цими ж технологіями. Наприклад, генеративні моделі на кшталт GPT-3 можуть бути використані для створення досконаліших фішингових атак або кампаній дезінформації.

2.2 Застосування методів машинного навчання для ідентифікації фішингових загроз

Метод передбачає використання автоматизованих класифікаторів, побудованих на основі алгоритмів машинного навчання та аналізу даних. Вони працюють на серверній стороні, здійснюючи категоризацію отриманих вебсторінок за визначеним набором характеристик. У таблиці 2.1 наведено порівняльний аналіз різних підходів до автоматичного розпізнавання.

Таблиця 2.1 – Оцінка ефективності різних класифікаторів

Алгоритм	**Основний принцип**	**Потрібність навчання**	**Переваги**	**Недоліки**
Naive Bayes	Кожен атрибут у вибірці розглядається незалежно від інших ознак класу. Метод ґрунтується на використанні байєсівської теореми. Дозволяє здійснити передбачення класу на основі обчислення ймовірностей.	Цей підхід вимагає попереднього навчання на основі розміченого набору даних, який використовується для формування класифікаційної таблиці.	Алгоритм базується на елементарних арифметичних операціях, Забезпечує високу швидкість обчислень. Ефективно масштабується при роботі з великими обсягами даних.	Метод базується на припущенні незалежності ознак, і його ефективність суттєво знижується у випадках, коли ця умова не виконується.
LR	Для прогнозування класу використовується лінійне рівняння з незалежними змінними. Основна мета методу — оцінити ймовірність настання певної події.	Для роботи цього методу необхідне попереднє навчання на тренувальній вибірці	Результати моделі легко піддаються інтерпретації. Алгоритм ефективно працює для прогнозування двійкових (бінарних) класів.	Метод потребує формулювання більшої кількості статистичних припущень перед його застосуванням. Найкраще працює зі змінними, між якими існує лінійна залежність;
J48	Класифікація здійснюється шляхом побудови дерева рішень, де кожне розгалуження відповідає перевірці певної ознаки. На кожному вузлі дерева проводиться оцінка важливості ознаки, що дозволяє класифікувати об'єкти за заданими критеріями.	Метод передбачає навчання на розміченому тренувальному наборі даних, у якому кожен приклад віднесений до певного класу.	Модель відзначається простотою інтерпретації результатів. Характеризується високою швидкістю обробки даних. Результати моделі є легкими для сприйняття людиною.	Існує ризик перенавчання моделі, особливо за наявності шумових або надмірно специфічних даних.

Продовження таблиці 2.1

Neural Network	Модель організована у вигляді мережі взаємопов'язаних елементів (нейронів), кожен з яких самостійно не виконує розпізнавання. З'єднання між нейронами забезпечують передачу сигналів, що дозволяє поширювати інформацію між шарами мережі.	Для роботи цього методу необхідне попереднє навчання на тренувальній вибірці	Модель має високу гнучкість і придатна як для класифікаційних, так і для регресійних задач. Може бути навчена на наборах даних будь-якого розміру, включно з великими або неоднорідними.	Потребують значних обчислювальних ресурсів для навчання та застосування моделі.Характеризуються підвищеною тривалістю обробки, особливо при роботі з великими наборами даних.Мають природу "чорного ящика", що ускладнює інтерпретацію внутрішніх механізмів прийняття рішень.
kNN	Метод є неконтролюваним за параметрами (не параметризований) і належить до контрольованого машинного навчання. Класифікація нових прикладів здійснюється шляхом аналізу належності до класу, домінуючого серед *k* найближчих сусідів у навчальній вибірці.	Для роботи цього методу необхідне попереднє навчання на тренувальній вибірці	Метод є інтуїтивно зрозумілим і простим для реалізації. Його впровадження не потребує складної математичної моделі. Точність класифікації значною мірою залежить від вибраної метрики відстані, що дозволяє досягати високих результатів за правильного налаштування.	Може потребувати значних обчислювальних ресурсів, особливо при обробці великих обсягів даних. Наявність шуму у вхідних даних може суттєво погіршити точність класифікації за методом kNN. Ознаки з великим діапазоном значень можуть непропорційно впливати на обчислення відстані, зменшуючи внесок ознак з меншим діапазоном. Метод вимагає більше пам'яті порівняно з активними класифікаторами, оскільки зберігає всі навчальні дані.

Продовження таблиці 2.1

SMO	Алгоритм належить до групи методів Support Vector Machine (SVM). Класифікація здійснюється шляхом побудови гіперплощини, яка розділяє дані на два класи.На концептуальному рівні SVM виконує подібні функції до алгоритму J48, однак застосовує інший принцип розділення. Модель самостійно визначає функцію гіперплощини для досягнення найкращого розмежування між класами.	Для роботи цього методу необхідне попереднє навчання на тренувальній вибірці	Здатний розв'язувати задачі квадратичного програмування (QP), що лежать в основі його оптимізаційного процесу. У випадках лінійного розподілу даних демонструє поведінку, подібну до логістичної регресії. Забезпечує ефективну класифікацію при наявності нелінійного розподілу завдяки вибору відповідного ядра (kernel).	Потребує попереднього вибору ядрової функції, що впливає на результат класифікації. Має низький рівень інтерпретованості, що ускладнює пояснення прийнятих рішень.

Розглянуті методи машинного навчання демонструють різний рівень ефективності, залежно від особливостей вхідних даних та умов задачі. Алгоритм Naive Bayes відзначається швидкістю і простотою, тоді як нейронні мережі мають високу гнучкість, але потребують значних ресурсів. Дерево рішень J48 та метод опорних векторів (SMO) забезпечують хорошу точність, але мають певні обмеження щодо інтерпретованості та схильності до перенавчання. Вибір класифікатора залежить від конкретних потреб системи виявлення фішингових атак та доступних обчислювальних можливостей.

2.3 Вивчення типових ознак фішингових повідомлень

Для виявлення фішингової складової певного повідомлення важливо встановити чіткі критерії прийняття рішень. Це передбачає визначення набору ключових показників, які потребують уваги під час аналізу таких повідомлень, що дозволить їх подальше використання для навчання нейронної мережі.

Перший показник, який потребує пильної уваги, – це довжина URL-адреси. Згідно з інформацією від Microsoft, при використанні методу GET максимальна кількість символів у фактичному шляху URL-адреси обмежена 2048 символами. У випадку методу POST формальних обмежень на довжину URL-адреси немає, оскільки дані передаються в заголовку. Однак браузери зазвичай мають внутрішні обмеження. Наприклад, максимальна довжина URL-адреси в Microsoft Internet Explorer становить 2083 символи.

Деякі дослідники радять обмежувати прийнятну довжину URL-адреси 75 символами, але обґрунтування такого обмеження часто відсутнє. Варто пам'ятати, що коротка адреса не є гарантією безпеки. Наприклад, через сервіс «PhishTank», який спеціалізується на боротьбі з фішинговими атаками, було виявлено фішинговий веб-сайт з URL-адресою лише з 21 символу (рисунок 2.1). Водночас, приклади довгих фішингових URL-адрес наведено на рисунок 2.2 а, б – їхня довжина становить 126 та 145 символів відповідно. У деяких випадках, наприклад, під час активації облікового запису або запрошень до проектів, URL-адреса може перевищувати 600 символів (рисунок 2.3). Таким чином, довжину адреси слід враховувати як один із факторів

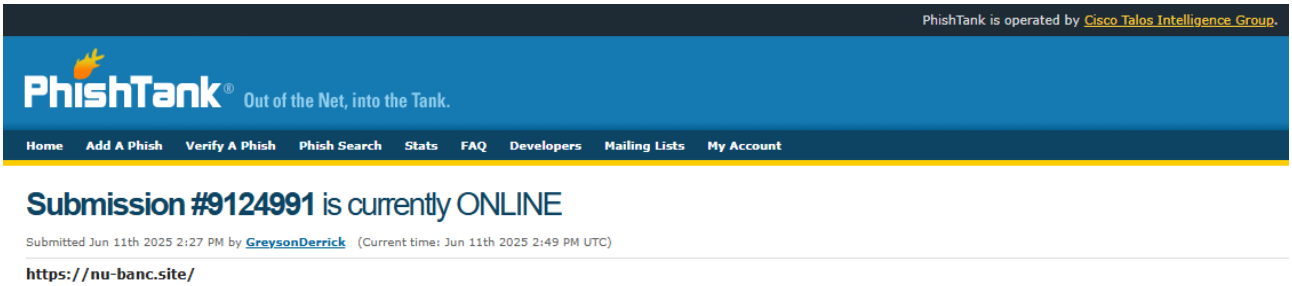
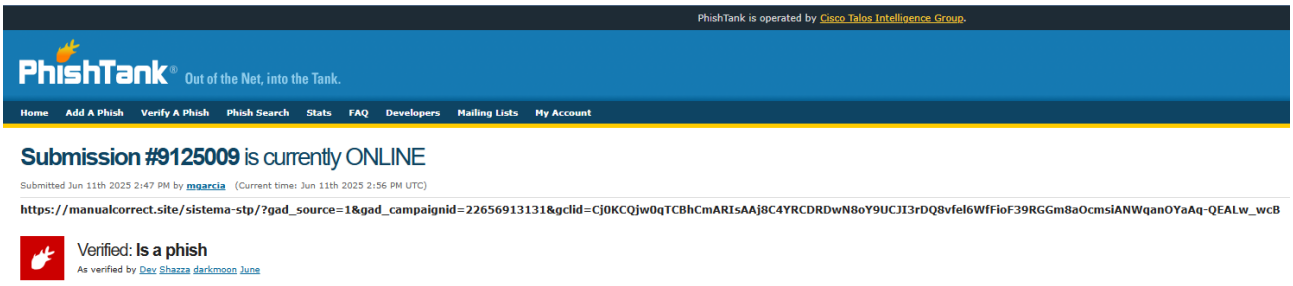


Рисунок 2.1 – Фішинговий сайт, замаскований під коротку URL-адресу

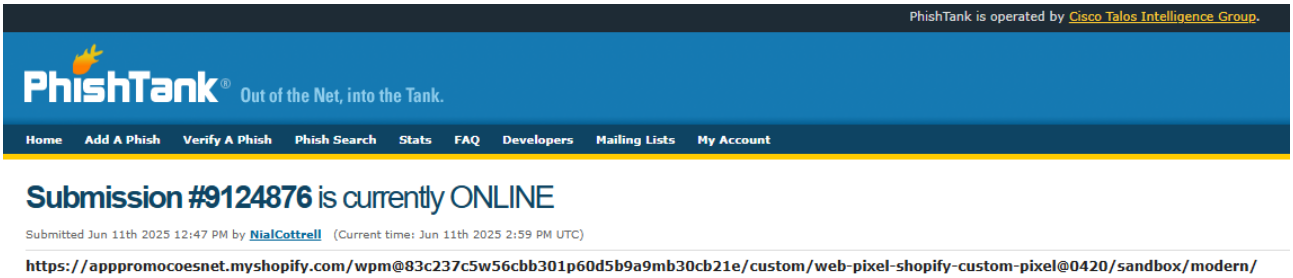
Окрім довжини URL-адреси, надзвичайно важливо ретельно проаналізувати її компоненти. Поширеною помилкою є те, що наявність HTTPS в адресі автоматично засвідчує безпеку. Насправді, фішингові ресурси часто використовують шифрування, оскільки отримати SSL/TLS-сертифікат сьогодні легко завдяки безкоштовним сервісам, таким як Let's Encrypt. Тому сама по собі наявність HTTPS не може служити незаперечним доказом надійності.

Також слід вивчити структуру доменного імені, особливо використання субдоменів або навмисне копіювання відомих брендів. Наприклад, фішингове посилання може виглядати так: https://secure-login.paypa1.com/ (де «paypa1» замінює «L» на «1») або https://paypal.com.security-check.info/ – у цьому випадку домен security-check.info, а paypal.com є частиною субдомену, призначеного для введення користувача в оману.

Таким чином, під час визначення потенційно небезпечних посилань необхідно враховувати не лише довжину, але й семантичну складність доменного імені, наявність незвичайних символів, кількість піддоменів, використання сервісів скорочення URL-адрес та інші точки зору.



а)



б)

Рисунок 2.2 – Зразки фішингових посилань довжиною а) 178 та б) 136 символів

Як видно з прикладу на рисунку 2.3, URL-адреса має довжину 154 символи та включає шість субдоменів: firebasestorage, googleapis, com, pppelwiieur, appspot і ooooommmwnneuuu. Це підкреслює важливість аналізу кількості крапок (.) у структурі веб-адреси, оскільки цей параметр може слугувати індикатором фішингової активності.

Рисунок 2.3 – URL-адреса із великою кількістю субдоменів

Наступний аспект, на який слід звернути увагу у разі підозри, – це використання фрагментованих посилань у тексті. Наразі існує багато онлайн-інструментів, які дозволяють значно скорочувати довгі адреси веб-сторінок, роблячи їх коротшими та зручнішими для розміщення в тексті, наприклад, в електронних листах або месенджерах. Однак саме ця практичність часто використовується шахраями для приховування фішингових посилань.

На рисунку 2.4 показано приклад скороченого фішингового посилання, створеного за допомогою відповідного онлайн-сервісу. Варто зазначити, що не всі такі сервіси дозволяють скорочувати шкідливі посилання – деякі з них перевіряють URL-адресу перед скороченням і навіть можуть видавати попередження про потенційну загрозу.

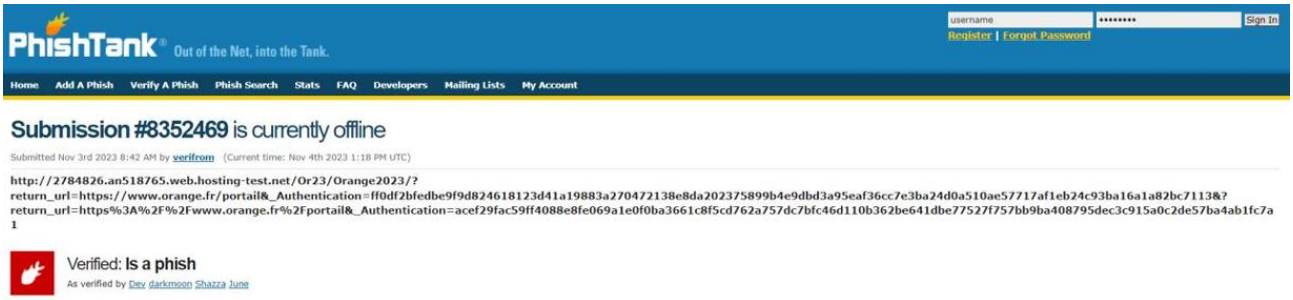


Рисунок 2.4 – Скорочене посилання, яке веде на сайт із ознаками фішингу

Аналогічно, фішингове посилання може використовувати символ «@», який здебільшого ігнорується браузерами. Будь-що перед символом «@» не впливає на фактичну адресу веб-сайту – за ним йде фактична URL-адреса, яку браузер використовуватиме для підключення. Це дозволяє зловмисникам приховувати небезпечні адреси під виглядом нібито легітимних.

Перейдемо до розгляду ознак текстової частини фішингових повідомлень, які можуть свідчити про шахрайський характер звернення:

- Загальні або неформальні звернення до користувача. Зловмисники рідко знають конкретного одержувача, тому використовують типові фрази, такі як «Шановний клієнте», без будь-якої персоналізації. Іноді зустрічається псевдоперсоналізація з використанням вигаданих імен або випадкових даних.
- Вимоги оновити особисту інформацію. Найчастіше фішингові повідомлення містять запити або вимоги оновити особисту інформацію, перейшовши за посиланням. Офіційні установи, включаючи банки, не використовують такі методи – вони пропонують зробити це через офіційні програми або особистий візит до відділення.
- Граматичні та стилістичні помилки. Наявність помилок у тексті, дивні звороти фраз або неприродний стиль письма є поширеною ознакою фішингу. Однак навіть добре написане повідомлення не гарантує його безпеки.
- Несподівана активність з боку установи. Якщо ви отримуєте повідомлення від фінансової чи іншої установи, з якою ви останнім часом не мали справ, це привід для роздумів. Офіційні повідомлення не повинні вимагати введення особистих даних у відповідь.
- Заклики до термінових дій. Фішингові повідомлення часто апелюють до страху або терміновості: «Підтвердьте негайно», «Ваш обліковий запис буде заблоковано» тощо. Це робиться для того, щоб викликати емоційну реакцію та змусити користувача діяти поспіхом.

- Підозріло вигідні пропозиції. Обіцянки подарунків, призів або продуктів за нереально низькими цінами – типовий психологічний трюк. Щоб отримати «подарунок», вас часто просять перейти за посиланням, зареєструватися або надати особисту інформацію. На рисунку 2.5 наведено приклад такого повідомлення.

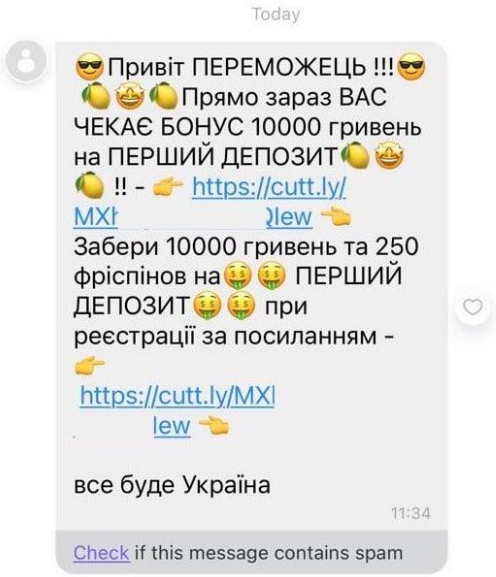


Рисунок 2.5 – Приклад спам-фішинг повідомлення

Виявлення фішингових повідомлень ґрунтується на сукупності технічних і семантичних ознак. До ключових критеріїв належать довжина та структура URL-адреси, використання субдоменів, наявність скорочених або фрагментованих посилань, а також характерні особливості тексту повідомлення – від граматичних помилок до закликів до термінових дій. Урахування цих ознак дозволяє ефективно формувати вхідні параметри для навчання моделей штучного інтелекту та підвищує точність виявлення фішингових атак.

2.4 Архітектура та процес навчання моделі ChatGPT

Аби збагнути, як відбувається навчання ChatGPT, слід заглибитися у механізм його роботи та структуру його внутрішньої логіки. Найперше, що важливо усвідомити, це ключовий принцип його діяльності: ChatGPT постійно

прагне створити "інтелектуальне продовження" тексту, отриманого на вхід. Під терміном "інтелектуальне" мається на увазі таке продовження, яке виглядає найбільш вірогідним – базуючись на досвіді того, як люди писали в аналогічних ситуаціях, що було взято з мільярдів веб-сторінок, книг та інших текстових джерел.

Для прикладу розглянемо речення: "Найкраще в штучному інтелекті – це його здатність до…". Уявимо, що алгоритм проводить сканування величезної кількості людських текстів (наприклад, з мережі Інтернет або оцифрованих книжок) та знаходить усі випадки, де зустрічається такий самий початок. Далі здійснюється аналіз, яке слово найчастіше з'являється після цієї фрази та з якою ймовірністю.

ChatGPT функціонує за аналогічним принципом, але працює не лише з буквальним текстом – він оцінює семантичну відповідність, тобто намагається передбачити таке слово або фразу, що найкраще підходить за змістом у контексті попереднього тексту. В результаті, модель створює впорядкований перелік можливих наступних слів, кожному з яких відповідає певне значення ймовірності. Приклад такого процесу показано на рисунку 2.6.

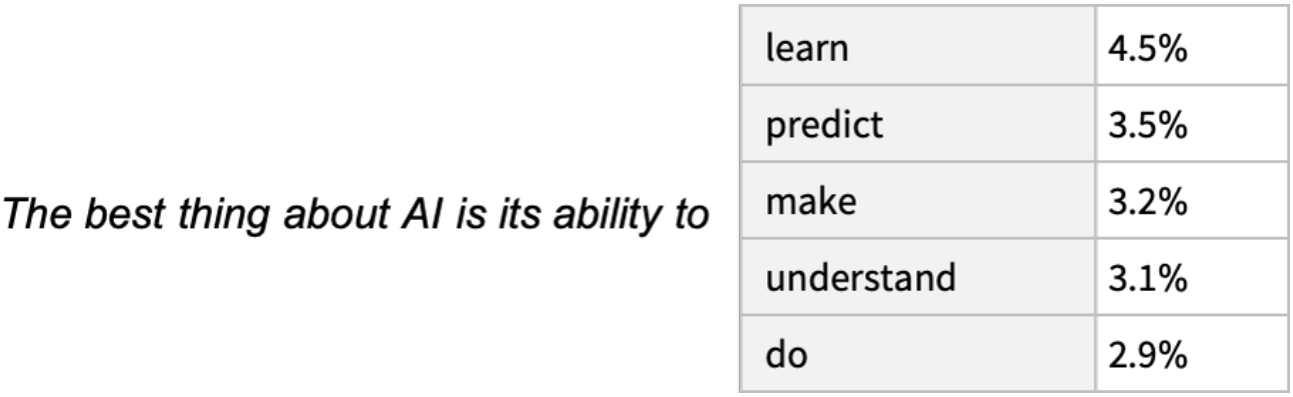


Рисунок 2.6 – Прогнозований набір слів, які можуть бути використані штучним інтелектом

Отже, коли ChatGPT складає речення, він, грунтуючись на своїх внутрішніх "знаннях", обирає слова, які найкраще відповідають поточному контексту. Точніше, модель додає не цілі слова, а специфічні маркери (токени),

котрі можуть бути частинами слів. Саме тому іноді ChatGPT може генерувати нові, раніше невідомі слова.

На кожному етапі створення тексту модель отримує низку можливих варіантів продовження (словосполучень або частин слів), кожний з яких має певну ймовірність появи. Виникає цілком резонне питання: яке саме слово обрати?

На перший погляд, найпростішим рішенням є завжди вибирати слово з найбільшою ймовірністю. Однак, цей метод може призвести до надмірно «сухого» та передбачуваного тексту, який часто дублюватиме існуючі вирази. Щоб уникнути цього, модель використовує певний ступінь випадковості – вона може вибирати не лише найімовірніше слово, але й таке, яке має меншу ймовірність, але водночас добре вписується в контекст. Завдяки цьому результат виглядає більш природним, гнучким та привабливим для читача.

Цей ступінь випадковості контролюється параметром під назвою «температура». Чим вища температура, тим більше варіативності та «креативності» в тексті; чим вона нижча, тим більш передбачуваним та логічно організованим він буде. Експериментально встановлено, що оптимальним значенням температури для генерації текстів, які є одночасно логічними та різноманітними, є значення 0,8.

Щоб краще зрозуміти механізм формування такого тексту, розглянемо простіший приклад з використанням моделі GPT-2, яку можна запустити навіть на звичайному персональному комп’ютері. Для цього ми скористаємося мовою програмування Wolfram Language та проаналізуємо, як саме відбувається вибір слів для продовження речення, що проілюстровано на рисунку 2.7.

```
In[ ]:= model =
          NetModel[{"GPT2 Transformer Trained on WebText Data",
            "Task" → "LanguageModeling"}]

Out[ ]= NetChain[ Input port:  string
                  Output port: class ]
```

Рисунок 2.7 – Команда та підключення моделі GPT-2 у Wolfram Mathematica

Далі ми передаємо йому вхідні дані: речення, яке потрібно розширити, та об'єкт керування, який визначає потрібну кількість слів, як показано на рисунку 2.8.

```
In[ ]:= model["The best thing about AI is its ability to", {"TopProbabilities", 5}]

Out[ ]= {do → 0.0288508, understand → 0.0307805,
         make → 0.0319072, predict → 0.0349748, learn → 0.0445305}
```

Рисунок 2.8 – Код для автоматичного добору п’яти слів у реченні

Згідно з рисунку 2.8, вихідні дані показують 5 слів, які найбільше підходять для завершення речення, враховуючи їх найвищі числові значення. На рисунку 2.9 представлено результат багаторазового використання моделі та наслідки цього процесу.

```
In[ ]:= NestList[StringJoin[#, model[#, "Decision"]] &,
          "The best thing about AI is its ability to", 7]

Out[ ]= {The best thing about AI is its ability to,
         The best thing about AI is its ability to learn,
         The best thing about AI is its ability to learn from,
         The best thing about AI is its ability to learn from experience,
         The best thing about AI is its ability to learn from experience.,
         The best thing about AI is its ability to learn from experience. It,
         The best thing about AI is its ability to learn from experience. It's,
         The best thing about AI is its ability to learn from experience. It's not}
```

Рисунок 2.9 – Результат генерування тексту після 7 проходів

Однак, під час створення великих блоків тексту модель схильна до повторень, що негативно впливає на загальну якість виводу. Це особливо помітно, коли використовується дуже точний алгоритм вибору наступного слова – наприклад, коли завжди вибирається слово з найбільшою ймовірністю (що відповідає параметру температури temperatura = 0). Приклад результату такого підходу показано на рисунку 2.10.

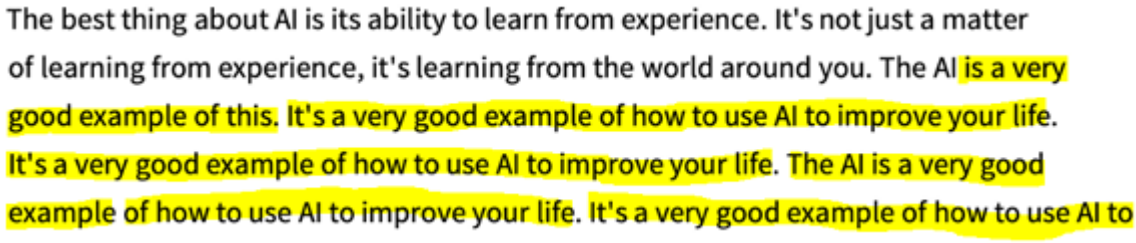


Рисунок 2.10 – Вихід тексту при майже нульовому значенні температури генерації

Однак, якщо вибір наступного слова є дещо випадковим, наприклад, при температурі = 0,8, то кожен окремий прогін моделі може генерувати унікальний варіант тексту. Як показано на рисунку 2.11, після п'яти послідовних прогонів з тим самим вступним фрагментом було отримано п'ять різних розширень.

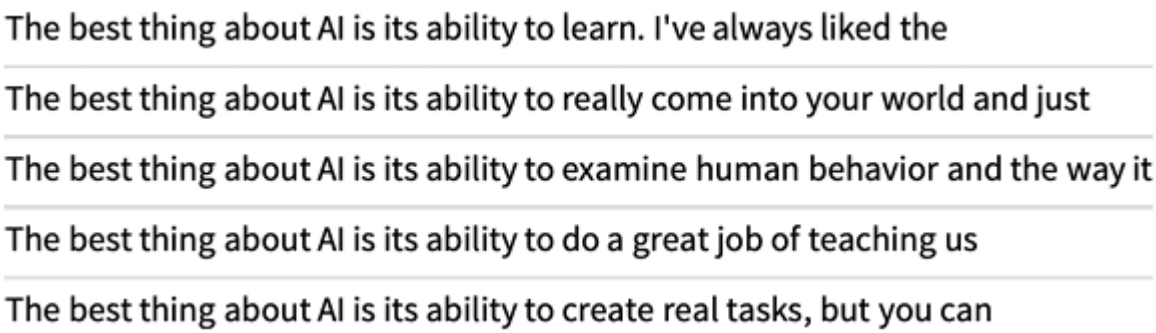


Рисунок 2.12 – Результат генерування тексту після 5 спроб з випадковим вибором наступного слова

Слід також наголосити, що навіть на першому етапі генерації за температури 0,8 є можливість вибору з широкого діапазону слів, хоча

ймовірності швидко зменшуються для менш підходящих варіантів. Це явище чітко продемонстровано на рисунку 2.13.

Підсумовуючи, було проведено експеримент з генерацією продовжень речень за різних температур (температура = 0 та температура = 0,8). Отримані результати, представлені на рисунку 2.14, показують вплив зміни температури на різноманітність та якість згенерованого тексту.

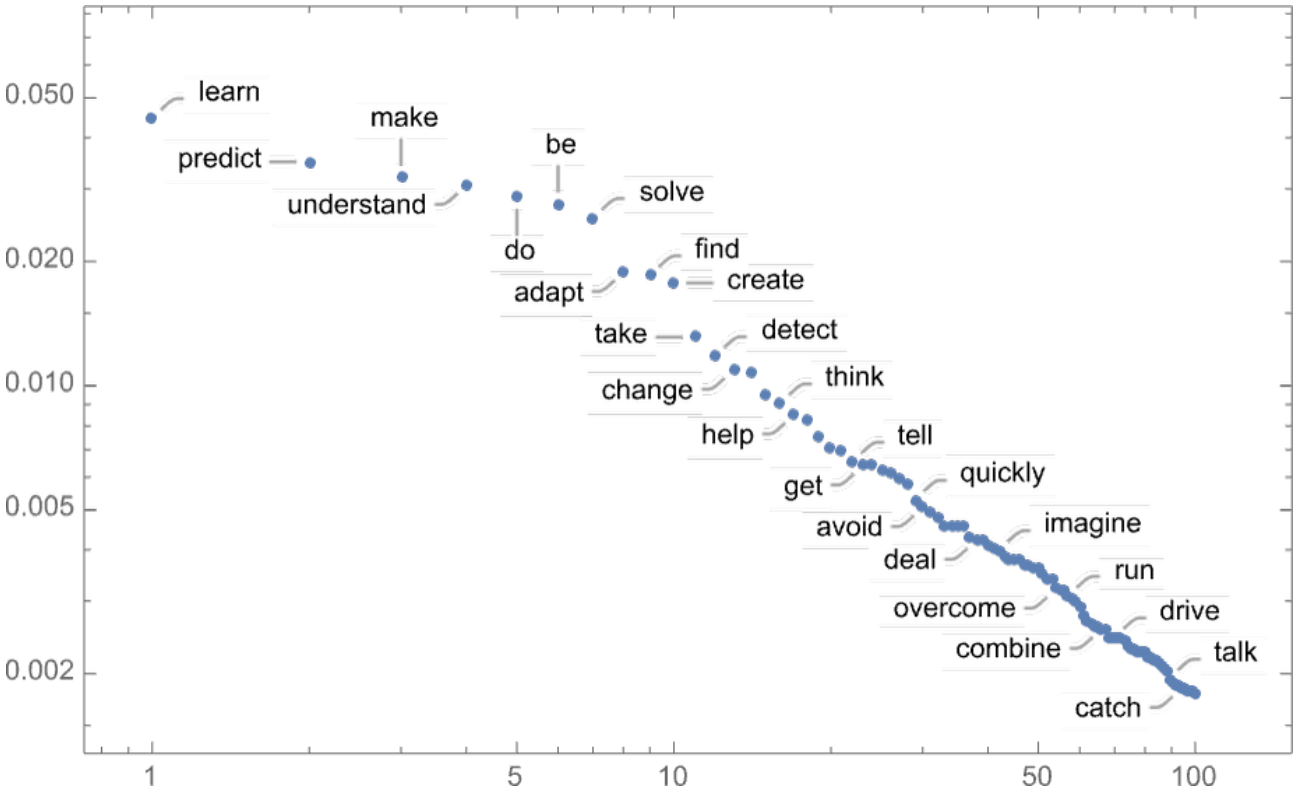


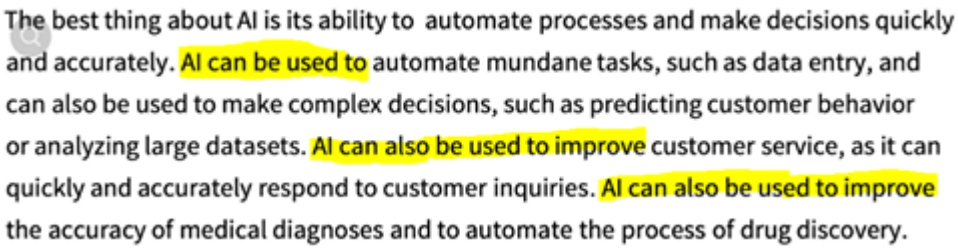
Рисунок 2.13 – Графічне зображення розподілу слів за ймовірністю їх використання

Навіть якщо ми використовуємо потужнішу модель, таку як GPT-3, як показано на рисунку 2.14 (а), загальна якість буде вищою, ніж у GPT-2, але деякі повторення все ще можна спостерігати. Наприклад, на рисунку 2.14 (б) чітко видно, що друге та третє речення мають схожий початок, що свідчить про обмеження різноманітності навіть за вищих температурних налаштувань.

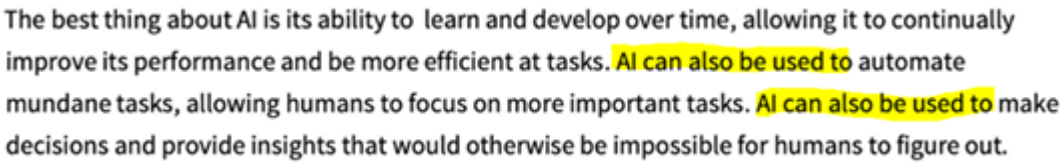
Щоб краще зрозуміти процес вибору слова в цих моделях, необхідно звернутися до аналізу ймовірностей, які визначають вибір наступного слова.

Варто врахувати, зокрема, частоту появи літер у певному тексті, створивши такі діаграми:

- Ймовірність появи символів у тексті (рисунок 2.14, а).
- Ймовірність появи другої літери в словах, якщо перша відома (рисунок 2.15, б). На особливу увагу заслуговує літера «q», за якою майже завжди йде «u» (інші поєднання практично відсутні, що позначено білими квадратиками – брак даних).

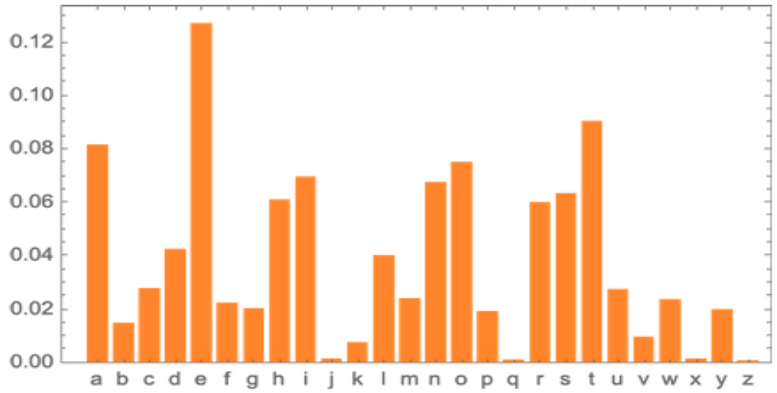


а)

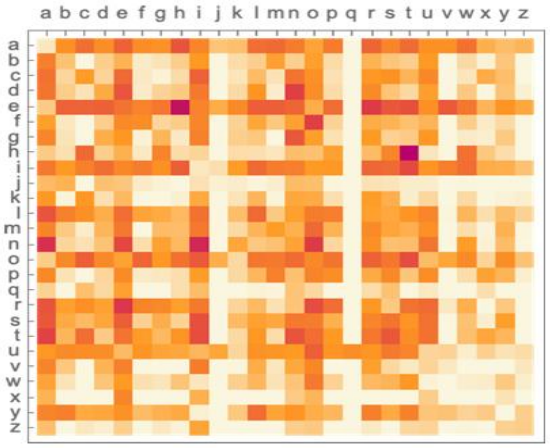


б)

Рисунок 2.14 – Текст, створений GPT-3 при температурі 0 (а) та 0,8 (б)



а)



б)

Рисунок 2.15 – Аналіз частоти літер англійської мови у вигляді графіка

Якщо у вас достатньо великий навчальний набір тексту, ви можете отримати ймовірнісні оцінки не лише для окремих літер або їх комбінацій, але й для довгих послідовностей. При генерації слів з поступовим урахуванням ширших контекстів (наприклад, трійки, четвірки тощо) результати стають все точнішими. Але ChatGPT використовує інший підхід: модель працює не на рівні символів, а на рівні слів або частин слова (токенів).

Для англійської мови найчастіше використовується близько 40 000 слів, тому, якщо обсяг даних достатній, цілком можливо побудувати ймовірнісну модель як окремих слів, так і їх комбінацій. Однак існує проблема масштабованості: кількість можливих комбінацій 40 000 слів зростає експоненціально. Наприклад:

• для пар слів: $40\,000^2$=1,6 мільярда комбінацій;

• для триплетів: $40\,000^3$=64 трильйони комбінацій.

Очевидно, що зберігання та використання таких обсягів інформації є надзвичайно ресурсомістким та практично недоцільним. Для вирішення цієї проблеми ChatGPT використовує підхід, заснований на моделі великої мови (LLM), який замість зберігання комбінацій використовує параметричні аналітичні функції для оцінки ймовірностей появи слів або фраз.

Таким чином, замість таблиць з мільярдами значень, модель використовує навчену нейронну мережу, яка дозволяє узагальнення, прогнозування та оцінку ймовірностей на основі попереднього контексту.

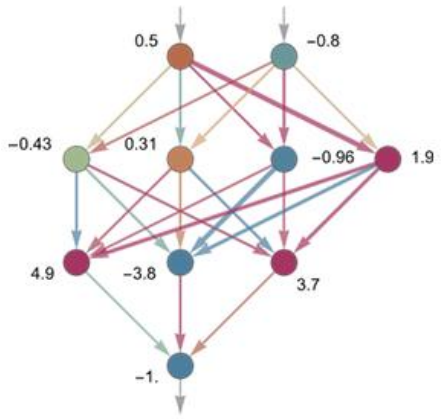
По суті, ChatGPT – це складна багатошарова нейронна мережа. Будь-яка нейронна мережа складається зі штучних нейронів, з'єднаних один з одним, які організовані у вигляді шарів (рисунок 2.16, а).

Кожен штучний нейрон виконує обчислення простої числової функції. Для використання мережі вхідні значення (наприклад, координати xxx, yyy або токени слів) подаються на вхід першого шару. Потім кожен нейрон обчислює своє власне значення, передає його наступному шару, і в результаті на виході отримується оброблене значення (рисунок 2.16, б).

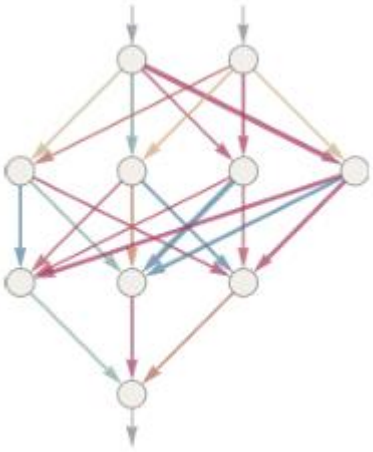
У біологічній метафорі кожен нейрон отримує вхідні сигнали через певну кількість з'єднань (синапсів), кожен з яких характеризується певною вагою. Щоб обчислити вихідне значення нейрона:

- Помножте кожне вхідне значення на відповідну вагу.
- Підсумуйте всі отримані результати.
- Додайте значення зсуву.
- Застосуйте функцію активації, яка вводить нелінійність.

Нелінійність дозволяє мережі відображати та обробляти складні взаємозв'язки та функції. Процес навчання моделі полягає в систематичному коригуванні ваг на основі великої кількості навчальних прикладів з використанням методів машинного навчання.



а)



б)

Рисунок 2.16 – Умовне зображення нейронів

Враховуючи результати, отримані в попередньому розділі, основним джерелом ризику є URL-адреса, отже, вона є центральним об'єктом дослідження. Для оцінки безпеки електронних листів рекомендується використовувати методологію, візуалізовану у вигляді алгоритму на рисунку 2.17. Отже, процес починається з моменту надходження нового електронного листа. Як правило, кожен електронний лист містить: текстовий контент, одне або кілька посилань, вкладені файли. Вкладення також можуть становити небезпеку, але більшість поштових платформ блокують перегляд електронних листів зі шкідливими вкладеннями.

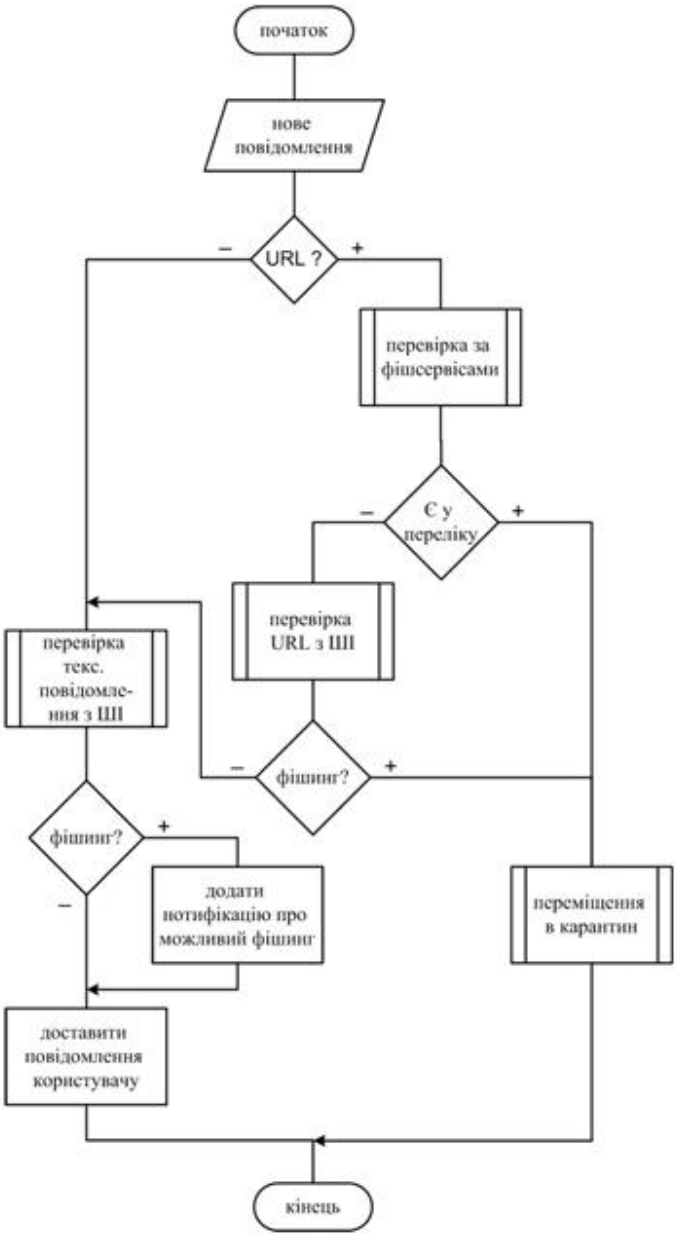


Рисунок 2.17 – Послідовність дій для виявлення фішингової загрози

У процесі пошуку можливих фішингових загроз у вхідних повідомленнях основна увага приділяється аналізу текстового вмісту та веб-адрес, що містяться в них. Алгоритм такого аналізу визначається структурою отриманого повідомлення та складається з кількох етапів, які виконуються один за одним.

На початковому етапі, після отримання повідомлення, виконується перевірка на наявність веб-адрес. Якщо адреси не знайдено, запускається процес аналізу текстового вмісту на наявність ознак фішингу. Якщо такі ознаки виявлено, користувач отримує повідомлення з попередженням про можливу

загрозу. Якщо підозрілих ознак не виявлено, повідомлення вважається безпечним та передається одержувачу без додаткових перевірок.

У випадку, коли повідомлення містить одну або кілька веб-адрес, кожна з них перевіряється за допомогою бази даних відомих фішингових ресурсів. Якщо в цій базі даних знайдено хоча б одну адресу, повідомлення негайно видаляється та переміщується до зони карантину, а подальша обробка припиняється. Важливо розуміти, що відсутність адреси в базі даних не гарантує її безпеки. Тому в цьому випадку запускається додаткова перевірка за допомогою інструментів штучного інтелекту. Якщо результати цієї перевірки вказують на потенційну загрозу, повідомлення також переміщується до карантину. Якщо перевірка не виявляє підозрілої активності, запускається процес аналізу текстового вмісту повідомлення, аналогічно ситуації з повною відсутністю веб-адрес.

Використання описаного алгоритму дозволяє не лише виявляти відомі загрози, але й ефективно виявляти нові, які ще не включені до баз даних фішингових ресурсів. Це, у свою чергу, значно покращує захист користувачів та дозволяє швидко реагувати на нові загрози інформаційній безпеці.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Алгоритм навчання мережі для аналізу текстової інформації

Навчання нейронних мереж – це процес пошуку оптимальних ваг та параметрів моделі, необхідних для вирішення конкретної задачі. Зазвичай навчання поділяється на такі етапи:

- Визначення мети. Важливо чітко сформулювати, яке завдання має вирішувати нейронна мережа. Це може бути класифікація, регресія, генерація тексту, розпізнавання зображень тощо.
- Збір та підготовка навчальних даних. Дані необхідно звести до формату, придатного для введення в нейронну мережу. Це може включати масштабування, кодування за категоріями та інші перетворення.
- Конфігурація моделі полягає у виборі архітектури штучної нейронної мережі з уточненням її внутрішньої структури та параметрів, зокрема кількості шарів, нейронів, функцій активації та алгоритмів навчання.
- Функція втрат – це метрика, що оцінює відхилення прогнозу моделі від фактичних значень.
- Вибір оптимізатора є критичним етапом у процесі навчання моделі, оскільки саме він відповідає за оновлення ваг на основі похибки. Найбільш поширені методи включають стохастичний градієнтний спуск (SGD), Adam, RMSprop та інші варіації, кожен з яких має свої переваги залежно від задачі, структури моделі та характеристик даних.
- Процес навчання моделі. На цьому етапі дані вводяться в модель, і безпосередньо ініціюється процес навчання. Значення ваг, що відповідають коефіцієнтам моделі, коригуються після обробки кожної частини даних.
- Після завершення етапу навчання проводиться оцінка продуктивності моделі на тестовому наборі даних, який не використовувався під час навчання, з метою перевірки здатності моделі узагальнювати нову інформацію.

- Точне налаштування моделі здійснюється на основі результатів попередньої оцінки, шляхом корекції її параметрів – таких як архітектура, швидкість навчання та кількість епох – з метою підвищення ефективності та точності моделі.
- Після підтвердження відповідної продуктивності модель розгортається для практичного використання – її застосовують для формування прогнозів на основі нових, раніше невідомих даних.

3.2 Набір даних для навчання

Для навчання моделі штучного інтелекту використовувалася база даних URL-адрес фішингових сайтів, отримана з порталу PhishTank – сервісу, який агрегує інформацію про фішингові ресурси та надає допоміжні дані. Як показано на рисунку 3.1, платформа також надає додаткову інформацію про кожен ресурс. Незважаючи на те, що реєстрація користувачів на момент написання статті була недоступна, функція отримання даних через API залишалася активною. Для цього необхідно було перейти до розділу «Розробники» та вибрати один із запропонованих способів взаємодії з API, як проілюстровано на рисунку 3.2.

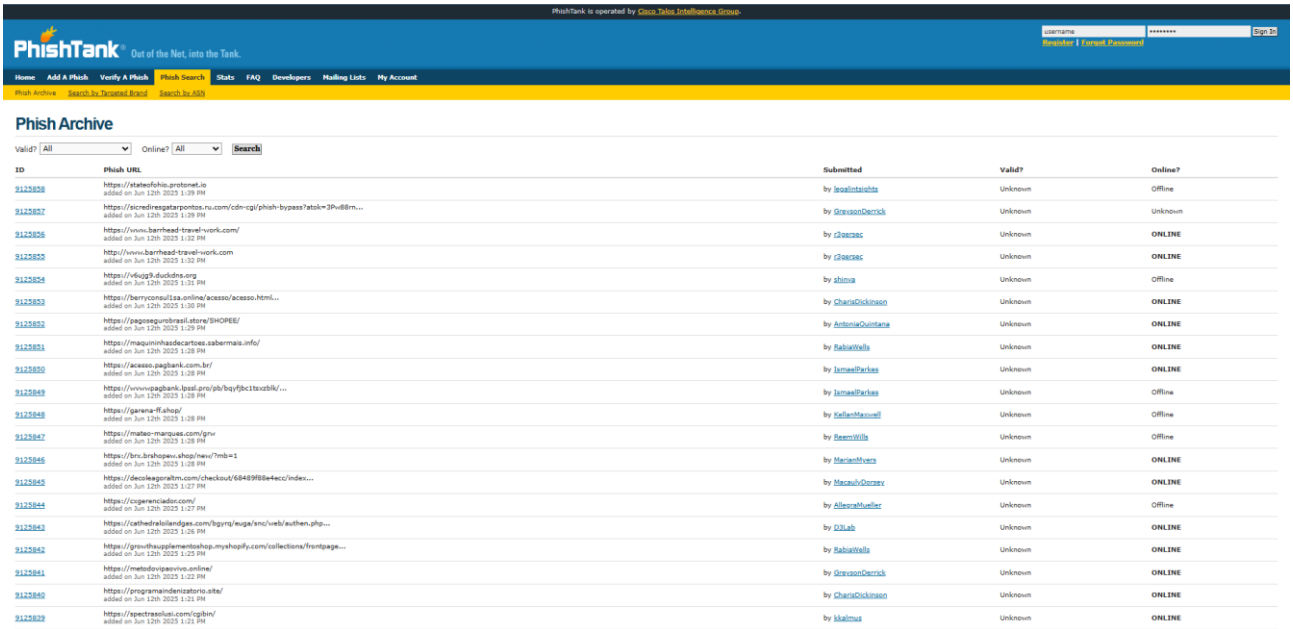


Рисунок 3.1 – Інструмент “Phish Search” для перевірки URL-адрес на PhishTank

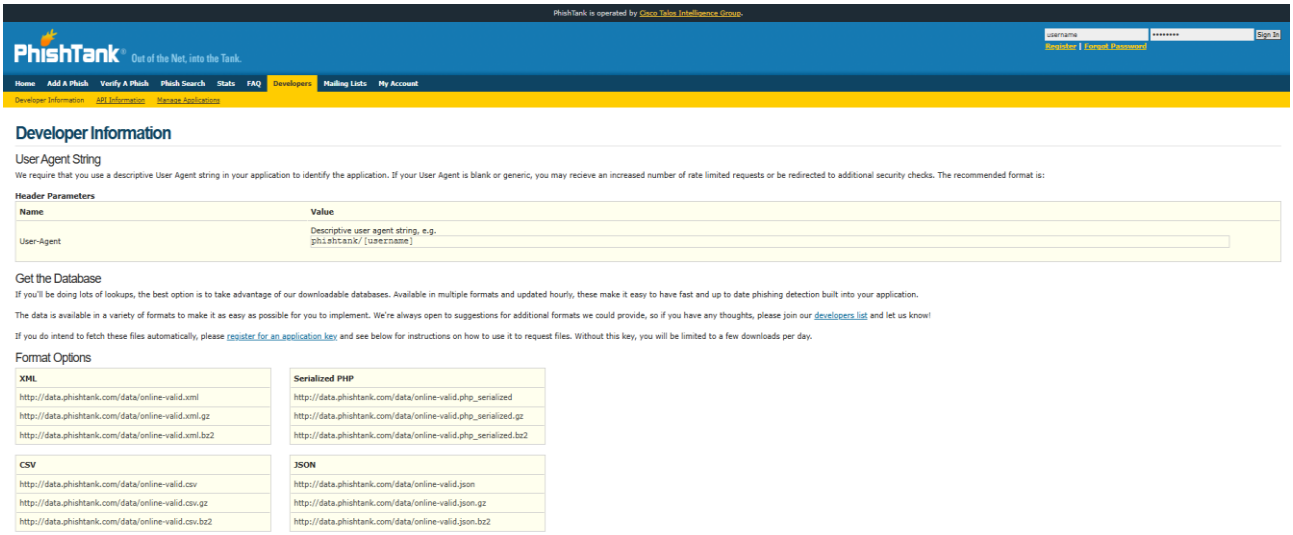
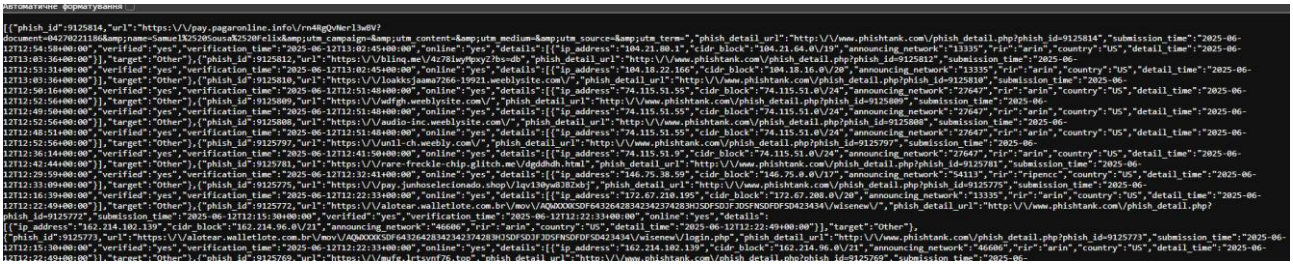


Рисунок 3.2 – Розділ “Developer” платформи PhishTank, призначений для інтеграції API

Згідно з рисунком 3.2, інформацію про джерела фішингу можна представити кількома способами:

- XML;
- CSV;
- Серіалізований PHP;
- JSON.

Варто підкреслити, що обсяг інформації залежить від обраного формату даних. Наприклад, формат CSV надає доступ лише до 8 інформаційних полів (див. рисунок 3.3, а), тоді як формат JSON дозволяє отримати до 14 полів (див. рисунок 3.3, б).

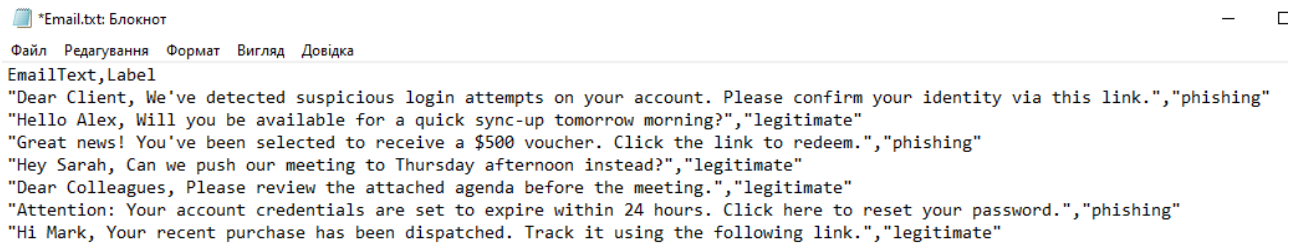


а)

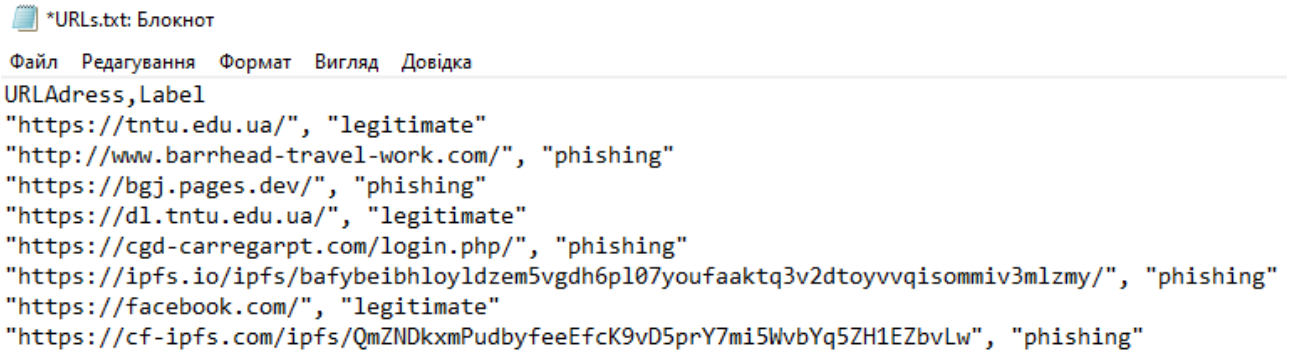
б)

Рисунок 3.3 – Формати, доступні для завантаження з PhishTank API: а) CSV, б) JSON

Використовуючи набір даних, отриманий з PhishTank.org, ми сформуємо набори даних для навчання нейронної мережі. Формат представлено на рисунку 3.4 а), а дані для URL-адрес відповідатимуть прикладу, показаному на рисунку 3.4 б).



а)



б)

Рисунок 3.4 – Типи файлів, що використовуються для тренування нейронних мереж

3.3 Створення Python-скриптів для виявлення фішингових листів і вебпосилань

Для взаємодії з нейронною мережею ChatGPT було використано спеціальний інтерфейс прикладного програмування (API), розроблений авторами мережі. На офіційному веб-сайті API є приклади застосувань та короткий посібник для розробників. Доступ до API забезпечується за допомогою різних технологій, зокрема:

• cURL

• Python

• Node.js

У цій роботі Python було обрано основною мовою програмування, враховуючи його великий потенціал для обробки даних, підтримку штучного інтелекту та наявність широкого спектру математичних бібліотек. Це значно полегшує процес розробки та тестування моделей машинного навчання.

Перший крок включає налаштування середовища Python та встановлення необхідних бібліотек. Для цього використовується менеджер пакетів pip. Нижче наведено список необхідних бібліотек та команди для їх встановлення:

Лістинг 3.1 – Встановлення необхідних бібліотек для обробки даних та машинного навчання

```
pip install pandas
pip install openai
pip install sklearn
```

Завантаження CSV-файлів здійснюється за допомогою pandas, тому спершу потрібно імпортувати цю бібліотеку.

```
import pandas as pd
```

Додатково імпортуються модулі з бібліотеки sklearn, необхідні для побудови моделей машинного навчання.

Лістинг 3.2 – Імпорт необхідних модулів з бібліотеки Scikit-learn для побудови моделі класифікації

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix
```

Зчитування даних із CSV-файлу (див. рисунок 3.4 а) здійснюється за допомогою такого рядка коду.

```
data = pd.read_csv('emails.csv')
```

Після завантаження даних текстові повідомлення слід трансформувати у числове представлення за допомогою

Лістинг 3.3 – Побудова векторного представлення текстових даних за допомогою CountVectorizer

```
CountVectorizer.vectorizer = CountVectorizer()
counts = vectorizer.fit_transform(data['EmailText'].values)
```

Цей код створює словник із текстових повідомлень і будує матрицю частот слів, після чого розділяє дані на навчальну та тестову вибірки.

Лістинг 3.4 – Розподіл даних на навчальну та тестову вибірки

```
X_train, X_test, y_train, y_test = train_test_split(counts, data['Label'],
 test_size=0.2, random_state=42)
```

Параметри функції:

- counts – векторизовані текстові дані
- data['Label'] – мітки ("phishing" або "legitimate")
- test_size – частка тестових даних (20%)
- random_state – фіксація результату поділу

Для класифікації використовується модель наївного баєсівського класифікатора MultinomialNB, який добре показує себе при роботі з текстом, зокрема в завданнях фільтрації спаму.

```
model = MultinomialNB()
model.fit(X_train, y_train)
```

Окрім MultinomialNB, можна також використовувати BernoulliNB або ComplementNB. Перший краще підходить для роботи з бінарними ознаками, другий показує більшу ефективність при роботі з незбалансованими вибірками.

Після навчання моделі необхідно отримати прогнози на основі тестових даних:

```
predictions = model.predict(X_test)
```

Для оцінки ефективності моделі використовується метрика точності:

```
print("Точність: ", accuracy_score(y_test, predictions))
```

Крім того, створюється матриця помилок (матриця плутанини), яка дозволяє побачити кількість правильних та неправильних класифікацій:

```
print("Confusion Matrix: \n", confusion_matrix(y_test, predictions))
```

На рисунку 3.5 зображено матрицю невідповідності в узагальненому вигляді, яка дозволяє оцінити ефективність роботи класифікатора шляхом порівняння передбачених і фактичних значень міток.

		Прогнозований клас	
	Загальна кількість = P + N	Predicted Positive (PP)	Predicted Negative (PN)
Справжній клас	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Рисунок 3.5 – Матриця невідповідності в узагальненому вигляді

На основі аналізу точності та матриці розбіжностей можна визначити ефективність моделі з реальними даними. Для отримання переконливих результатів, як правило, точність моделі повинна перевищувати 92%. За таких обставин її можна використовувати для практичного розпізнавання фішингу та корекції електронних повідомлень.

3.4 Тестування та аналіз результатів

Для навчання буде використано набір даних, що містить 24 250 фішингових та 24 500 легітимних URL-адрес, тобто загалом 48 750 навчальних

записів [19] . Ці дані були отримані з вихідного скрипта, проаналізованого в попередньому розділі, з використанням трьох подібних класифікаторів:

- MultinomialNB;
- BernoulliNB;
- ComplementNB.

Варто підкреслити, що дані для аналізу були згенеровані для всіх трьох класифікаторів з використанням різних тестових вибірок (test_size): 0,2, 0,3, 0,4, 0,5. Тому для кожного значення розміру тестової вибірки та для кожного класифікатора окремо була створена матриця плутанини за допомогою функції confusion_matrix(y_test, predictions) та розраховані показники точності класифікації. Розглянемо детальніше позначення, що використовуються для демонстрації отриманих результатів:

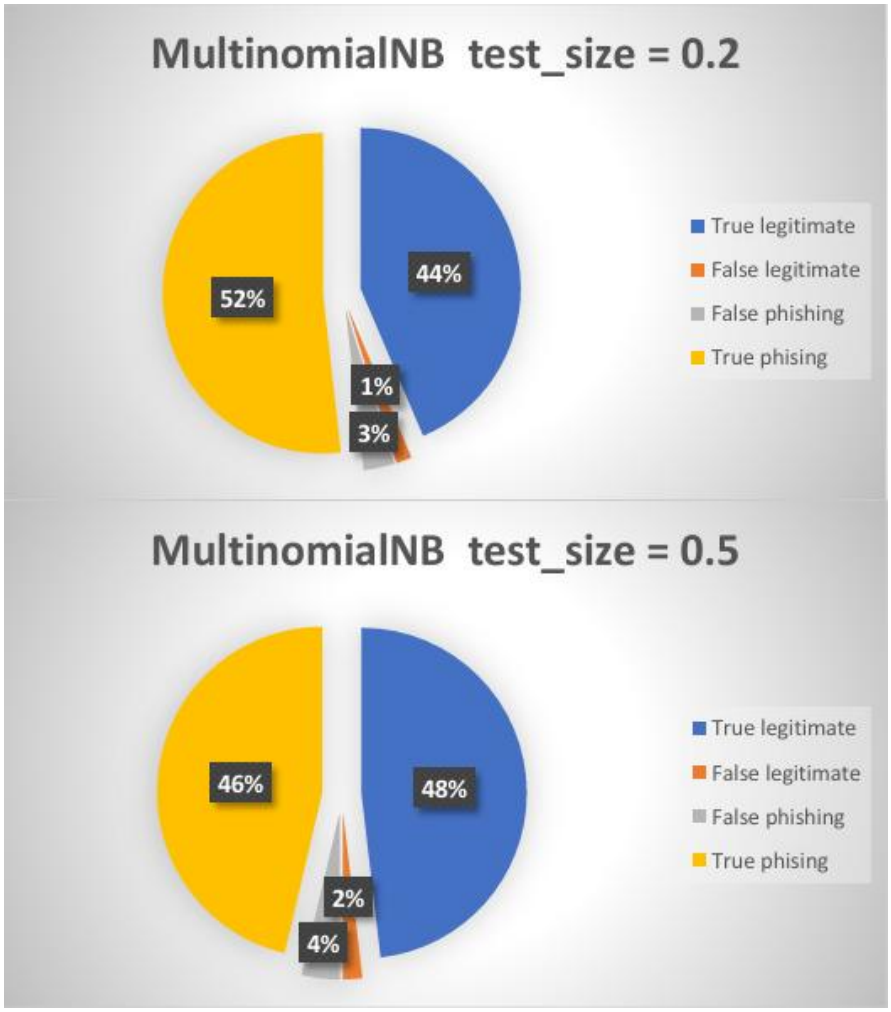
- True positive – правильно класифікована легітимна URL-адреса;
- False positive – неправильно класифікована легітимна URL-адреса;
- False negative – неправильно класифікована фішингова URL-адреса;
- True negative – правильно класифікована фішингова URL-адреса;
- Precision score legitimate – точність класифікації легітимних URL-адрес;
- Precision score phising – точність класифікації фішингових URL-адрес;
- Total score – загальна точність класифікації.

У контексті класифікатора MultinomialNB результати узагальнено в таблиці 3.1. Однак вважається корисним візуалізувати ці дані, як представлено на рисунку 3.6.

Таблиця 3.1 – Оцінка ефективності класифікатора MultinomialNB на тестових даних

Test size	True positive	False positive	False negative	True negative	Precision score legitimate	Precision score phishing	Total score
0.2	4325	162	290	4987	0.964	0.948	0.956
0.3	6712	149	632	7210	0.980	0.920	0.950
0.4	8820	369	745	9723	0.960	0.930	0.946
0.5	11683	460	915	11240	0.965	0.927	0.947

Рисунок 3.6 – Матриця невідповідності для різних обсягів тестової вибірки



Згідно з представленими графіками, стає очевидним, що найвищі показники точності були досягнуті на тестовій вибірці 0,2. Слід зазначити, що

саме при цьому значенні відсоток правильно визначених фішингових адрес перевищував 50%, що свідчить про різницю в розподілі даних між тестовою та навчальною вибірками, спричинену "перетасовуванням". Результати, отримані при використанні класифікатора BernoulliNB, наведено в таблиці 3.2.

Дані, отримані при використанні класифікатора BernoulliNB, наведено в таблиці 3.2.

Таблиця 3.2 – Оцінка ефективності класифікатора BernoulliNB на тестових даних

Test size	True positive	False positive	False negative	True negative	Precision score legitimate	Precision score phishing	Total score
0.2	4520	301	85	4934	0.955	0.947	0.951
0.3	6825	410	190	7240	0.943	0.935	0.939
0.4	9040	605	288	9525	0.937	0.920	0.929
0.5	11380	920	380	11800	0.930	0.915	0.923

На рисунку 3.7 графічно представлено матрицю невідповідності для таблиці, наведеної вище.

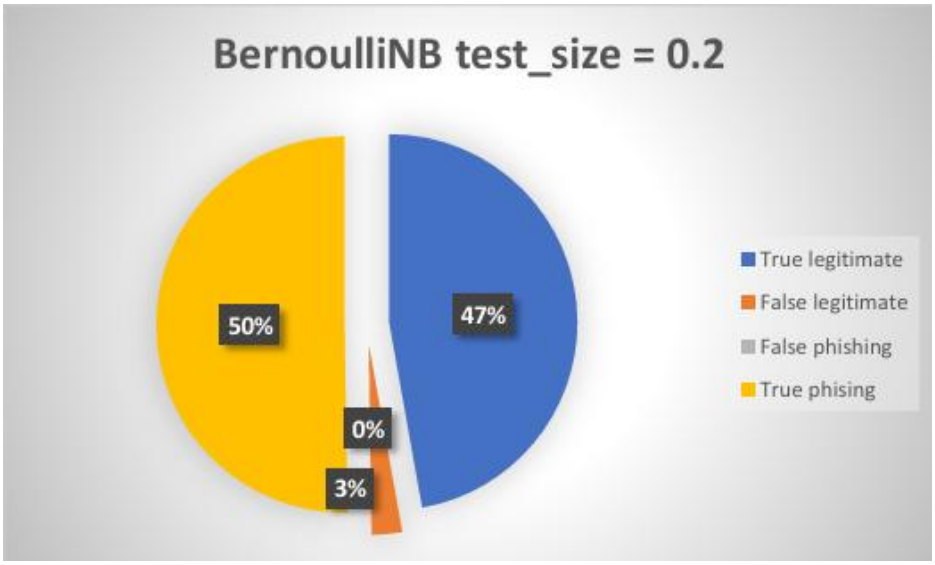


Рисунок 3.7 – Матриця невідповідності для різних обсягів тестової вибірки BernoulliNB

Отже, на противагу від класифікатора BernoulliNB, поточна модель демонструє кращі результати при виявленні легітимних URL-адрес, хоча точність класифікації цих адрес лише незначно перевищує відповідний показник для фішингових. Крім того, було проведено тестування за допомогою класифікатора ComplementNB, результати якого наведено в таблиці 3.3.

Таблиця 3.3 – Оцінка ефективності класифікатора ComplementNB на тестових даних

Test size	True positive	False positive	False negative	True negative	Precision score legitimate	Precision score phishing	Total score
0.2	4825	128	189	4675	0.964	0.961	0.962
0.3	6980	225	365	7075	0.941	0.951	0.946
0.4	9512	312	520	9156	0.923	0.945	0.934
0.5	11735	514	689	11289	0.921	0.936	0.928

Рисунок 3.8 демонструє графічне представлення матриці помилок для класифікатора ComplementNB..

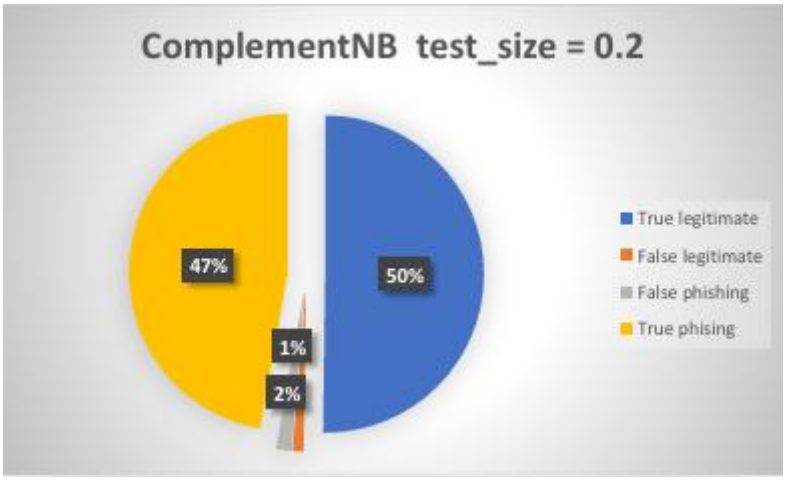


Рисунок 3.8 – Матриця невідповідності класифікатора ComplementNB

У таблиці 3.4 подано результати дослідження класифікаторів, що дозволяє наочно зіставити їхню ефективність.

Таблиця 3.4 – Результати класифікації (тестовий розмір 0.2)

Класифіка-тор	True positive	False positive	False negative	True negative	Precision score legitimate	Precision score phishing	Total score
MultinomialNB	4325	162	290	4987	0.964	0.948	0.956
BernoulliNB	4520	301	85	4934	0.955	0.947	0.951
ComplementNB	4825	128	189	4675	0.964	0.961	0.962

Зіставивши інформацію з таблиці, стає очевидним, що ComplementNB та BernoulliNB демонструють майже ідентичні показники загальної точності, хоча їх класифікаційна здатність трохи відрізняється.

Зокрема, ComplementNB відзначається ефективнішою роботою в ідентифікації фішингових адрес, про що свідчить вищий показник precision для категорії "phishing". Це робить його оптимальним вибором для систем, де

ключовим завданням є виявлення шкідливих URL-адрес та мінімізація помилок типу False Negative (тобто, пропусків фішингових посилань).

Натомість, BernoulliNB демонструє кращі значення precision для легітимних URL-адрес, підкреслюючи його перевагу в правильному розпізнаванні безпечних веб-адрес. Це робить його більш відповідним для випадків, коли критичним є збереження високої точності при визначенні безпечних URL, зменшуючи кількість помилок типу False Positive (хибного розпізнавання як фішингових).

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Заходи щодо захисту від ураження електричним струмом

Фахівець з кібербезпеки здебільшого працює в офісних умовах або в середовищі серверних приміщень, де постійно використовується велика кількість електронного обладнання: комп'ютери, сервери, джерела безперебійного живлення, системи відеоспостереження, мережеве обладнання тощо. Через це ризик ураження електричним струмом є суттєвим і потребує ретельного контролю з боку як працівника, так і роботодавця.

Основні джерела ризику:

- Робота з несправними або саморобними подовжувачами, блоками живлення.
- Надмірне навантаження на електромережу (включення великої кількості пристроїв в одну розетку).
- Вологе або сире середовище в серверних кімнатах.
- Відсутність заземлення або несправність елементів електрозахисту.
- Самостійне втручання у схеми живлення (наприклад, під час підключення нових серверів або мережевого обладнання).

Види уражень електричним струмом.

Залежно від сили струму, тривалості дії та шляху проходження через тіло, електричне ураження може призвести до:

- електричного шоку;
- порушення серцевого ритму;
- зупинки дихання;
- опіків тканин;
- летальних випадків.

Особливо небезпечним є змінний струм частотою 50 Гц, який використовується в електромережах України – саме він має найбільший уражаючий ефект на серцевий м'яз.

Основні заходи електробезпеки.

Для зменшення ризику ураження електричним струмом у середовищі ІТ та кібербезпеки мають застосовуватись наступні заходи:

- Організаційні заходи.

Проведення регулярного інструктажу з охорони праці та електробезпеки (первинного, повторного, цільового).Забезпечення доступу до електроустановок лише кваліфікованим особам із групою допуску не нижче II (згідно з Правилами безпечної експлуатації електроустановок споживачів).Заборона самостійного підключення або ремонту обладнання без відповідного дозволу. Розміщення схем електропостачання офісу/серверної в доступних місцях.

- Технічні заходи.

Обов'язкове використання заземлення для всього електрообладнання. Використання пристроїв захисного відключення (УЗО, диференційних автоматів). Перевірка цілісності ізоляції кабелів, а також справності вилок, розеток, подовжувачів. Організація системи резервного живлення (UPS), яка захищає не лише дані, але й працівника від стрибків напруги. Забезпечення відповідності навантаження потужності електромережі.

- Індивідуальні заходи.

Заборона роботи з обладнанням мокрими руками або у вологому одязі. Забезпечення використання діелектричних килимків та рукавиць для обслуговуючого персоналу. Своєчасне повідомлення про виявлені несправності відповідальній особі. Недопущення розташування кабелів у місцях, де можливе їхнє механічне пошкодження.

У випадку електротравми важливо діяти швидко та грамотно:

- Негайно припинити дію струму – відключити електроживлення або усунути постраждалого від джерела струму (за допомогою сухої дерев'яної палиці, діелектричних рукавиць).
- Викликати швидку допомогу – 103.
- Перевірити життєві функції – дихання, пульс, свідомість.
- Надати домедичну допомогу – у разі зупинки дихання або серцебиття розпочати СЛР (серцево-легеневу реанімацію).

- Забезпечити спокій та спостереження до приїзду медиків.
- Незважаючи на те, що фахівці з кібербезпеки зазвичай не працюють безпосередньо з високовольтним обладнанням, їхня професійна діяльність пов’язана із значною кількістю електронних пристроїв. Тому дотримання правил електробезпеки, а також наявність чітких дій у разі нещасного випадку є обов’язковими складовими безпечної професійної діяльності.

4.2. Дії працівників офісу у разі загрози терористичного акту або повідомлення про замінування

У сучасних умовах роботи, особливо в галузі інформаційних технологій, значна частина фахівців працює в офісах, розташованих у багатоповерхових будівлях або бізнес-центрах. Через загальну напружену ситуацію у світі, а також воєнний стан в Україні, не можна виключати ймовірність терористичних загроз, зокрема — повідомлень про замінування чи виявлення підозрілих предметів. Знання правильного алгоритму дій у таких випадках є важливим фактором безпеки життєдіяльності працівників.

Правове регулювання

Відповідно до статті 153 Кодексу законів про працю України, роботодавець зобов’язаний забезпечити безпечні та нешкідливі умови праці. Крім того, згідно з вимогами Закону України "Про охорону праці" (ст. 13, 14), роботодавець зобов’язаний розробити та впровадити заходи щодо запобігання надзвичайним ситуаціям, у тому числі терористичним загрозам.

У випадку виникнення терористичної загрози діють норми Кримінального кодексу України, а саме стаття 258 "Терористичний акт" та Закон України "Про боротьбу з тероризмом". Алгоритм дій працівників офісу формується з урахуванням Наказу МВС України № 649 від 28.07.2017, який містить інструкції щодо поведінки у разі виявлення підозрілих предметів.

Типові загрози

У контексті офісної роботи можуть виникати такі види терористичних загроз:

- Повідомлення про замінування будівлі (анонімне або від імені конкретної особи);
- Виявлення підозрілого предмету (сумка, коробка, пакунок) у приміщенні або біля нього;
- Загроза через електронну пошту або телефонний дзвінок;
- Зовнішнє проникнення озброєних осіб до будівлі.

Алгоритм дій працівників у разі загрози.

У випадку терористичної загрози всі працівники зобов'язані діяти згідно з попередньо затвердженим Планом евакуації та інструкціями з безпеки, розробленими службою охорони праці та затвердженими адміністрацією.

При отриманні повідомлення про замінування:

- Негайно повідомити керівника або відповідального за охорону праці.
- Оповістити всіх працівників офісу (без створення паніки).
- Не торкатися жодних підозрілих предметів.
- Викликати поліцію та ДСНС (телефон 101 або 102).
- Розпочати організовану евакуацію відповідно до плану.

При виявленні підозрілого предмета:

- Не наближатися, не чіпати, не відкривати.
- Зафіксувати місцезнаходження, описати його (колір, розмір, форма).
- Огородити місце виявлення.
- Повідомити відповідальні служби.
- Евакуювати людей на безпечну відстань (не менше 100 метрів).

Під час евакуації:

- Не користуватися ліфтами.
- Уникати скупчення та паніки.
- Допомагати колегам, які мають труднощі з пересуванням.
- Зібратися в попередньо визначеному безпечному місці.

Профілактичні заходи:

- Проведення регулярних інструктажів з охорони праці та цивільного захисту.
- Розміщення схем евакуації на видимих місцях.
- Наявність "тривожної кнопки" чи внутрішньої системи оповіщення.
- Періодичні навчання персоналу на тему дій при НС (1-2 рази на рік).
- Залучення фахівців з МВС або СБУ для консультацій.

Терористичні загрози є реальними і потенційно небезпечними для будь-якого офісного середовища. Працівники ІТ-компаній повинні бути не лише фахівцями у своїй галузі, але й знати, як діяти в критичних ситуаціях. Завчасна підготовка, інструктажі, наявність плану евакуації та чітке виконання інструкцій – ключ до збереження життя і здоров'я колективу. Відповідальність за створення таких умов лежить як на роботодавцеві, так і на кожному працівнику окремо.

ВИСНОВКИ

У межах виконання дипломної роботи було здійснено комплексне дослідження проблеми виявлення фішингових URL-адрес з використанням алгоритмів машинного навчання. Основна мета роботи полягала у створенні скрипта для автоматизованої класифікації інтернет-посилань на фішингові та легітимні, а також в оцінці ефективності обраних алгоритмів. У результаті було реалізовано програмну систему на мові Python, яка виконує обробку вхідних URL-адрес, перетворює їх на векторні ознаки та класифікує за допомогою трьох алгоритмів: MultinomialNB, BernoulliNB та ComplementNB.

Було використано навчальну вибірку, що містила 48 750 записів, з яких 24 250 є фішинговими, а 24 500 – легітимними. Для перевірки якості роботи моделей було проведено серію експериментів із різними обсягами тестової вибірки (20%, 30%, 40% та 50%). На основі отриманих результатів обчислювались матриці невідповідностей, а також ключові метрики класифікації: точність, повнота та F1-міра. Порівняння моделей показало, що алгоритм ComplementNB продемонстрував вищу ефективність у більшості випадків, зокрема при роботі з нерівномірно збалансованими вибірками. Це дозволяє рекомендувати його для практичного застосування в системах кібербезпеки, орієнтованих на виявлення фішингових загроз.

Окрім технічної частини, дипломна робота містить розділ, присвячений питанням безпеки життєдіяльності та охорони праці працівника ІТ-сфери. Розглянуто потенційні загрози, що можуть виникати під час роботи за комп’ютером в офісному середовищі, зокрема ризики ураження електричним струмом, а також методи очищення повітря від шкідливих виділень, які утворюються внаслідок роботи комп’ютерної техніки. Запропоновано комплекс заходів профілактики та дій у разі виникнення небезпечної ситуації. Усі рекомендації базуються на положеннях чинного законодавства України та відповідних нормативно-правових актах у сфері охорони праці.

Загалом, отримані результати можуть бути використані як для практичної реалізації систем моніторингу URL-адрес з метою запобігання фішинговим атакам, так і для подальших наукових досліджень у галузі машинного навчання, обробки природної мови та інформаційної безпеки. Поєднання технічного аналізу з практичними рекомендаціями з безпеки праці забезпечує комплексний підхід до проблеми, що відповідає сучасним вимогам до фахівців з кібербезпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kovalchuk, O., Karpinski, M., Banakh, S., Kasianchuk, M., Shevchuk, R., & Zagorodna, N. (2023). Prediction machine learning models on propensity convicts to criminal recidivism. Information, 14(3), art. no. 161, 1-15. doi: 10.3390/info14030161.

2. Karpinski, M., Korchenko, A., Vikulov, P., Kochan, R., Balyk, A., & Kozak, R. (2017, September). The etalon models of linguistic variables for sniffing-attack detection. In 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems : Technology and Applications (IDAACS) (Vol. 1, pp. 258-264). IEEE.

3. ZAGORODNA, N., STADNYK, M., LYPA, B., GAVRYLOV, M., & KOZAK, R. (2022). Network Attack Detection Using Machine Learning Methods. Challenges to national defence in contemporary geopolitical situation, 2022(1), 55-61.

4. Tymoshchuk, D., Yasniy, O., Mytnyk, M., Zagorodna, N., Tymoshchuk, V., (2024). Detection and classification of DDoS flooding attacks by machine learning methods. CEUR Workshop Proceedings, 3842, pp. 184 - 195.

5. Boltov, Y., Skarga-Bandurova, I., & Derkach, M. (2023, September). A Comparative Analysis of Deep Learning-Based Object Detectors for Embedded Systems. In 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) (Vol. 1, pp. 1156-1160). IEEE.

6. Lechachenko, T., Gancarczyk, T., Lobur, T., & Postoliuk, A. (2023). Cybersecurity Assessments Based on Combining TODIM Method and STRIDE Model for Learning Management Systems. In CITI (pp. 250-256).

7. Sedinkin , O., Derkach , M., Skarga-Bandurova , I., & Matiuk , D. (2024). Eye tracking system based on machine learning. COMPUTER-INTEGRATED TECHNOLOGIES: EDUCATION, SCIENCE, PRODUCTION, (55), 199-205.

8. PhishTank. URL-адреси, визнані фішинговими [Електронний ресурс]. – Режим доступу: http://data.phishtank.com/data/online-valid.csv

9. Nazario, J. Phishing Corpus [Електронний ресурс] // Мережевий архів фішингових URL. – Режим доступу: https://monkey.org/~jose/phishing/

10. Костюк, О.І. Методи виявлення фішингових атак в електронному листуванні // Збірник наукових праць. – Київ: НТУУ "КПІ", 2020. – №2. – С. 52–59.

11. Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. – MIT Press, 2016. – 800 с.

12. Провоторов, В.М. Системи штучного інтелекту: Навчальний посібник. – Харків: ХНУРЕ, 2019. – 188 с.

13. Kaggle Datasets: Phishing Websites Dataset [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/datasets

14. Secure information system for Chinese Image medicine knowledge consolidation; CEUR Workshop Proceedings. Vol. Vol-3896. 2024. S. Lupenko, O. Orobchuk, I. Kateryniuk, R. Kozak, H. Lypak. https://ceur-ws.org/Vol-3896/

15. Федоров, С.В. Методи інтелектуального аналізу даних: класифікація та прогнозування. – Львів: Видавництво ЛНУ, 2020. – 216 с.

16. Котенко,Д., & Хлапонін,Ю.(2024). Штучний інтелект у системах виявлення і запобігання кібератакам: перспективи та виклики. Підводні технології: промислова та цивільна інженерія, 1(14),48–55.

17. Бабкін,А.А., & Кудін,О.В. (2020). Огляд нейромережевих моделей систем виявлення вторгнень. Вчені записки Таврійського національного університету ім. В.І. Вернадського. Серія: Технічні науки, 31(70), 77–82.

18. Sarker, I. H., Furhad, M. H., Nowrozy, R. (2021). AI-driven cybersecurity: an overview, security intelligence modeling and research directions. SN Computer Science, 2(3)

19. Магденко,А.Р., Бучацький,І.О., & Бондаренко,І.О. (2024). Штучний інтелект: нова зброя у руках кіберзлочинців та шахраїв.

20. Закон України «Про охорону праці» №2694-XII від 14.10.1992.

21. Правила безпечної експлуатації електроустановок споживачів (затверджені наказом Міністерства праці та соціальної політики України №135 від 09.10.1998).

22. ДСТУ EN 50110-1:2014. Експлуатація електроустановок.

23. Наказ МОЗ України №400 від 28.12.1993 «Про затвердження Державних санітарних норм мікроклімату виробничих приміщень».

24. ДБН В.2.2-3:2018 «Будинки і споруди. Заклади освіти. Загальні положення» (актуальні вимоги до вентиляції, освітлення, електробезпеки).

25. Підручник: Небезпечні фактори виробничого середовища: навчальний посібник / За ред. І.М. Ткачука. — К.: Ліра-К, 2020.

26. Підручник: Охорона праці в галузі інформаційних технологій / В.М. Тарасенко, А.М. Литвин. — К.: КНЕУ, 2019.

Додаток А

Повний Python-код для тренування моделей (MultinomialNB, BernoulliNB, ComplementNB)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB, BernoulliNB,
ComplementNB
from sklearn.metrics import confusion_matrix, precision_score,
accuracy_score
df = pd.read_csv('phishing_dataset_balanced.csv')
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df['url'])
y = df['label']
models = {
    "MultinomialNB": MultinomialNB(),
    "BernoulliNB": BernoulliNB(),
    "ComplementNB": ComplementNB()
}
test_sizes = [0.2, 0.3, 0.4, 0.5]
for name, model in models.items():
    print(f"\n===== {name} =====")
    for size in test_sizes:
        print(f"\nTest size = {size}")

        X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=size, random_state=42)
                model.fit(X_train, y_train)
                predictions = model.predict(X_test)

        cm = confusion_matrix(y_test, predictions)
        print("Confusion matrix:")
        print(cm)

        precision_legit = precision_score(y_test, predictions,
pos_label=0)
        precision_phish = precision_score(y_test, predictions,
pos_label=1)
        total_accuracy = accuracy_score(y_test, predictions)

        print(f"Precision (Legitimate):
{precision_legit:.4f}")
        print(f"Precision (Phishing):
{precision_phish:.4f}")
        print(f"Total Accuracy:         {total_accuracy:.4f}")
```