ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка методики виявлення фішингових електронних листів за допомогою алгоритмів машинного навчання»

на здобуття освітнього ступеня магістра

зі спеціальності 121 Інженерія програмного забезпечення
(код, найменування спеціальності)

освітньо-професійної програми «Інженерія програмного забезпечення»
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Артур ГАЙШУК
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-62

Артур ГАЙШУК

Керівник: Богдан ХУДІК
к.т.н., доцент

Рецензент: _____

Київ 2026

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
_____Ірина ЗАМРІЙ
«_____»_____2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Гайшуку Артуру Олеговичу

1. Тема кваліфікаційної роботи: «Розробка методики виявлення фішингових електронних листів за допомогою алгоритмів машинного навчання»

керівник кваліфікаційної роботи Богдан ХУДІК к.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2025 р. № 320.

2. Строк подання кваліфікаційної роботи «17» грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, дані про існуючі методи та підходи виявлення фішингових листів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз предметної галузі.
2. Аналіз існуючих інструментів та технологій.
3. Практична реалізація оптимізованої архітектури системи.
4. Оцінка ефективності впроваджених оптимізацій.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз існуючих методів та підходів виявлення фішингових листів.
2. Порівняння стандартизованого методу та розробленого.
3. Метрики оцінки класифікації.
4. Вхідні дані.
5. Результат оцінки ефективності методів.
6. Порівняння результатів ефективності методів.
7. Єкранні форми.

6. Дата видачі завдання «16» жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	31.10-04.11.25	
2	Вивчення технологій API	04.11-10.11.25	
3	Аналіз існуючих методів та підходів виявлення фішингових листів	11.11-16.11.25	
4	Розробка методу виявлення фішингових листів	18.11-24.11.25	
5	Тестування різних методів виявлення фішингових листів	25.11-28.11.25	
6	Оформлення роботи: вступ, висновки, реферат	29.11-02.12.25	
7	Розробка демонстраційних матеріалів	13.12-19.12.25	

Здобувач вищої освіти _____ Артур ГАЙШУК

(підпис)

Керівник
кваліфікаційної роботи _____ Богдан ХУДІК

(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 80 стор., 6 табл., 9 рис., 63 джерел.

Мета роботи – підвищення ефективності виявлення фішингових електронних листів за допомогою алгоритмів машинного навчання та уніфікованого інтерфейсу AMSI.

Об'єкт дослідження – процес вдосконалення методики виявлення фішингових електронних листів.

Предмет дослідження – методи та алгоритми оптимізації обробки даних та взаємодії API для покращення здатності виявляти фішингові електронні листи.

Короткий зміст роботи: у роботі проведено аналіз існуючих методів та підходів до виявленя фішінгових листів та їх недоліки. На основі запропонованих методик розроблено методику для виявлення фішінгових листів з використанням машинного навчання ML та уніфікованого інтерфейсу AMCI. Проведено порівння ефективності різних моделей виявлення фішінгових листів. Проведено тестування та порівняння різних методів виявлення фішиногвих листів.

КЛЮЧОВІ СЛОВА: ФІШИНГОВІ ЛИСТИ, АЛГОРИТМИ, МАШИННЕ НАВЧАННЯ, КЕШУВАННЯ ДАНИХ, АРХІТЕКТУРА ЗАПРОПОНОВАНОГО МЕТОДУ, КЛАСИФІКАТОР.

ABSTRACT

Text part of the master's qualification work: 80 pages, 9 pictures, 6 tables, 63 sources.

The purpose of the work – improving the performance of a cryptocurrency market monitoring service by optimizing data processing and API interaction processes..

Object of research – the process of data processing and interaction with APIs in a cryptocurrency market monitoring service.

Subject of research – methods and algorithms for optimizing data processing and API interactions to enhance the performance of the service.

Summary of the work: the work analyzes existing methods and approaches to detecting phishing emails and their shortcomings. Based on the proposed methods, a method for detecting phishing emails using ML machine learning and the unified AMCI interface was developed. The effectiveness of the methods was compared. The effectiveness of different models for detecting phishing emails was compared. Different methods for detecting phishing emails were tested and compared.

KEYWORDS: PHISHING LETTERS, ALGORITHMS, MACHINE LEARNING, DATA CACHE, ARCHITECTURE OF THE PROPOSED METHOD, CLASSIFIER.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД	-	База Даних
API	-	Application Programming Interface
SMTP	-	Simple Mail Transfer Protocol
AMQP	-	Advanced Message Queuing Protocol
REST	-	Representational State Transfer
HTTP	-	HyperText Transfer Protocol
CPU	-	Central Processing Unit
RAM	-	Random Access Memory
CLI	-	Command Line Interface
UI	-	User Interface
UX	-	User Experience
SVM/SV	-	Support Vector Machine/Support Vector Classifier

ВСТУП

У сучасній архітектурі глобальних цифрових комунікацій електронна пошта існує для ділової та особистої взаємодії, але водночас вона є головним вектором поширення загроз у кіберпросторі. Серед різноманіття атак особливе місце посідає фішинг — метод соціальної інженерії, спрямований на викрадення конфіденційних даних (паролів, фінансових реквізитів, інтелектуальної власності) шляхом маніпуляції психологією користувача.

Станом на 2025 рік характер фішингових атак зазнав фундаментальної трансформації. Поява потужних великих мовних моделей (LLM) дозволила зловмисникам автоматизувати створення високореалістичних, персоналізованих повідомлень, які не містять традиційних ознак спаму, таких як граматичні помилки або підозрілі вкладення. Класичні системи захисту, що базуються на чорних списках (blacklists) та сигнатурному аналізі, демонструють дедалі меншу ефективність, оскільки вони неспроможні адаптуватися до атак «нульового дня» (Zero-day phishing).

Актуальність розробки методики обумовлена потребою у переході від статичної фільтрації до динамічного семантичного аналізу. Машинне навчання дозволяє системі захисту самостійно виявляти приховані закономірності в текстах, метаданих та структурі листів, ідентифікуючи аномалії, які є невидимими для людського ока або простих алгоритмів.

Робота присвячена підвищенню ефективності виявлення фішингових електронних листів за допомогою алгоритмів машинного навчання та уніфікованого інтерфейсу AMSI.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Огляд сучасних методів та підходів фішингових листів

З розвитком інформаційних технологій та підвищенням інтересу до цифрового середовища виникла потреба у створенні ефективної методики виявлення фішингових листів за допомогою алгоритмів машинного навчання. Яка би виявляла, запобігала та захищала користувачів від відповіді на фішингові електронні листи, що містять шкідливі посилання та вкладення, тим самим допомагаючи цільовим користувачам зменшити кількість фішингових електронних листів.

Електронна пошта є однією з найпоширеніших функцій Інтернету, поряд з веб-сторінками. Вона дозволяє надсилати та отримувати повідомлення будь-кому, хто має адресу електронної пошти, будь-де у світі. Електронна пошта використовує кілька протоколів у рамках набору TCP/IP. Електронна пошта вже досить давно є надзвичайно важливим засобом комунікації, що дозволяє майже миттєво отримати доступ до будь-якої частини світу з підключенням до Інтернету [1].

Незважаючи на свої переваги, електронна пошта має кілька недоліків. Найпоширенішими з них є фішингові та спам-листи. Хоча як фішингові листи, так і спам можуть засмічувати вашу поштову скриньку, лише фішинг спеціально розроблений для крадіжки паролів для входу та іншої важливої інформації. Спам – це маркетингова стратегія, яка передбачає надсилання небажаних електронних листів великим групам людей з метою просування продуктів та послуг.

Фішинговий електронний лист – це електронний лист, що виглядає справжнім, який має на меті обдурити користувачів, змусивши їх думати, що це законний електронний лист, а потім або розкрити конфіденційну інформацію, або завантажити шкідливе програмне забезпечення, натиснувши на шкідливі посилання, що містяться в тілі листа. Фішинг є більш шкідливим у цьому аспекті, оскільки він завдав величезних фінансових втрат користувачам домену.

Тому існує нагальна потреба у виявленні фішингових електронних листів з

високою точністю. Банківська інформація, кредитні звіти, дані для входу та інша конфіденційна та особиста інформація часто передаються електронною поштою. Це робить їх цінними для кіберзлочинців, які можуть використовувати ці знання для власної вигоди.

У контексті фільтрації електронної пошти використовуються різні методи фільтрації небажаної пошти, такі як методи, засновані на знаннях, методи кластеризації, методи, засновані на навчанні, евристичні процеси тощо, але проблема полягає в тому, що вони не здатні контролювати атаки обходу [3].

Для підвищення точності системи виявлення фішингової електронної пошти та для контролю обходу важливо вивчити проблеми сучасного методу виявлення спаму, оскільки зловмисник швидко впроваджує нові методи, вміст фішингового посилання та шкідливих вкладень час від часу змінюється.

Розробка спам-фільтрів продовжуватиме бути активною галуззю досліджень для науковців та фахівців галузі, які досліджують методи машинного навчання для ефективної фільтрації спаму [4]. Поточні функції рішень здебільшого відстають від інновацій, які постійно впроваджують спамери, що значною мірою виправдовує появу антиспам-пропозицій на основі машинного навчання.

Впровадження фільтрації електронної пошти (тобто фішинг проти фальсифікації) надзвичайно важливе для будь-якої організації. Фільтрація електронної пошти не лише запобігає потраплянню спаму до поштових скриньок, але й покращує якість життя ділових електронних листів, гарантуючи їх ефективне функціонування та використання лише за призначенням [4].

Фільтрація фішингу є, по суті, інструментом боротьби зі шкідливим програмним забезпеченням, оскільки багато атак електронною поштою намагаються обманом змусити користувачів натискати на небезпечні вкладення або URL-адреси, запитувати конфіденційну інформацію тощо.

Фішинг спричиняє кілька проблем прямо чи опосередковано для системи електронної пошти [5]. Серед них: мережеве з'єднання, нецільове використання місця для зберігання та обчислювальних ресурсів, втрата продуктивності роботи та роздратування користувачів, юридичні проблеми в результаті порнографічної

реклами та інших небажаних матеріалів, фінансові втрати через фішинг та інші пов'язані атаки, поширення вірусів, черв'яків та троянських коней, атаки типу «відмова в обслуговуванні» та «збір даних з каталогів».

Фішингові атаки потенційно більш шкідливі порівняно зі спам-листами. Оскільки вони розроблені, щоб виглядати законно, але мають на меті завдати шкоди, маніпулювати або обманом змусити людей робити те, чого вони зазвичай не роблять або не повинні робити. Через це ми зосереджуємося на підвищенні ефективності виявлення фішингових листів зокрема.

Швидке зростання фішингових методів фільтрації небажаної пошти спамерами створює труднощі для дослідників. Через це виявлення фішингу та механізм фільтрації спам-повідомлень є критично важливою сферою діяльності дослідників.

Тому впровадження, моделювання та проектування механізмів виявлення спам-повідомлень час від часу потребує змін у цих механізмах відповідно до особливостей фішингу. Інша річ полягає в тому, що зараз спамери можуть легко обійти всі програми фільтрації спаму [7].

Протягом четвертого кварталу 2024 року 22,5 відсотка фішингових атак установи [8], однією з яких є постачальники інтернет-послуг. Інтернет широко використовується в наш час, і збільшення кількості фішингових спам-повідомлень призводить до втрати часу та грошей для користувачів, які щодня отримують близько сотні фішингових листів. Виявлення фішингу є важливою темою для захисту людей від небажаних комерційних листів.

Таким чином, існує потреба в дослідженні кращих способів виявлення фішингових електронних листів та сповіщення користувача про них або запобігання їх потраплянню до користувачів. Головна мета полягає в тому, щоб забезпечити ефективний, покращений спосіб класифікації фішингових електронних листів, фільтрації та запобігання відкриттю та відповіді на фішингові електронні листи користувачем на основі складності сучасних методів виявлення.

Фішингові електронні листи часто пов'язані зі спамом, і більшість цих методів спрямовані на контроль спаму як механізм запобігання таким шахрайським

схемам з крадіжкою особистих даних. Основна відмінність полягає в тому, що спам-повідомленням бракує належного вибору ознак, які належним чином розмежовують спам від фішингових повідомлень.

Жан та ін. [9] запропонували метод виявлення та фільтрації фішингових електронних листів, використовуючи стохастичні слабкі оцінювачі на основі навчання (SLWE) у реальному середовищі.

Підхід SLWE був вивчений та впроваджений на основі наївної баєсівської класифікації для фільтрації фішингових електронних листів, які мають непередбачуваний характер. Вони використовували два різні набори даних: 1200 реальних нешкідливих електронних листів та 600 реальних фішингових листів. Щоб оцінити ефективність своєї пропозиції, вони порівняли отримані результати, отримані за допомогою підходу SLWE, з методом максимальної правдоподібності (MLE).

MLE – це популярна та широко використовувана схема оцінки. Їхні результати показали, що наївний баєсівський підхід на основі SLWE перевершує схему MLE щодо точності. Однак запропонований ними метод має недоліки у вигляді величезної кількості функцій, які можуть впливати на продуктивність системи, та необмеженого навчання, яке може споживати великі обсяги пам'яті [9].

Чандрасекаран покладався на відмінні структурні особливості електронної пошти для виявлення фішингових листів. Ці функції працюють у співпраці з SVM, щоб передбачати фішингові листи та запобігати їх початковому надходженню до користувача [10].

Луег представив короткий огляд, щоб дослідити прогалини в тому, чи можна застосувати технологію фільтрації інформації та пошуку інформації для постулювання виявлення спаму в електронній пошті логічним, теоретично обґрунтованим чином, щоб сприяти впровадженню методу фільтрації спаму, який міг би працювати ефективно.

Однак, в огляді не було представлено детальної інформації про алгоритми машинного навчання, інструменти моделювання, загальнодоступні набори даних та архітектуру середовища боротьби зі спамом в електронній пошті.

Марсоно, М.Н. та ін. [14] представили розробку обладнання для наївного баєсівського висновку для контролю спаму з використанням двокласової класифікації електронної пошти. Це може обробляти понад 117 мільйонів ознак щосекунди, враховуючи потік ймовірностей як джерела інформації. Ця робота може бути застосована для дослідження проактивного спаму, що враховує плани приймаючих поштових серверів та регулювання спаму на мережевих шлюзах.

Ю. Танг, С. Крассер та ін. [15] розробили фреймворк, який використовує SVM для класифікації, такий фреймворк видаляє інформацію про поведінку відправника електронної пошти з урахуванням глобальної дисперсії відправлення, досліджує їх та призначає оцінку довіри кожній IP-адресі, яка надсилає електронне повідомлення. Експериментальні результати показують, що класифікатор SVM є життєздатним, точним та значно швидшим, ніж класифікатор випадкових лісів (RF).

Раті та ін. [16] запропонували метод визначення найкращого класифікатора для класифікації електронної пошти за допомогою методів інтелектуального аналізу даних. Вони порівняли продуктивність численних класифікаторів, використовуючи методи інтелектуального аналізу даних «з алгоритмом вибору ознак» та «без алгоритму вибору ознак». Також вчені розглянули заданий алгоритм вибору ознак після вибору найкращого методу вибору ознак.

Вони використовують різноманітні алгоритми для експериментів зі своїми даними, включаючи метод Нейва Баєса, мережу Баєса, метод опорних векторів, дерево функцій, J48, випадковий ліс та випадкове дерево. У всьому наборі даних є 4601 вхід та 58 атрибутів. Максимальна точність для методу випадкового дерева становила 99,72 відсотка, тоді як найнижча точність для алгоритму Нейва Баєса становила 78,94 відсотка.

ДеБарр та ін. [17] використовують алгоритми випадкового лісу для класифікації спам-пошти, а потім застосовують активне навчання для вдосконалення моделі класифікації. Вони взяли дані з електронних листів RFC 822 (Інтернет), розділили кожне на два розділи, а потім перетворили кожне повідомлення на ознаки TF/IDF.

Вибрали початкову колекцію електронних листів для позначення як навчальних прикладів за допомогою алгоритму розділення навколо медоїдів (PAM) та методу кластеризації. Вони досліджують за допомогою випадкового лісу, наївного баєсівського методу, SVM та KNN після розгляду повідомлень-прототипів кластера для навчання. З точністю 95,2% алгоритм випадкового лісу є найкращим класифікатором. З точністю 95,2% є найкращим класифікатором.

Сахамі та ін. [18] запропонували використовувати ознаки для фільтрації небажаної пошти та створили баєсівський класифікатор. Явними виділеннями були такі фрази, як «Безкоштовні гроші» та «!!!!» над орнаментованими знаками наголосу. Точність фільтрів була покращена шляхом розміщення цих додаткових виділень поруч із матеріалом торгової марки електронного повідомлення.

Коли йдеться про ділову електронну пошту, є кілька речей, які слід пам’ятати: Дешевість — незалежно від відстані чи кількості одержувачів, надсилання електронного листа коштує однаково. Швидкість — електронний лист має досягти місця призначення протягом кількох хвилин, якщо не годин.

Співпраця передбачає одночасне спілкування з групою людей. Багато компаній та організацій спілкуються та керують своїм листуванням за допомогою поштових програм, таких як Microsoft Outlook. Архітектура системи електронної пошти проілюстрована на рисунку 1.

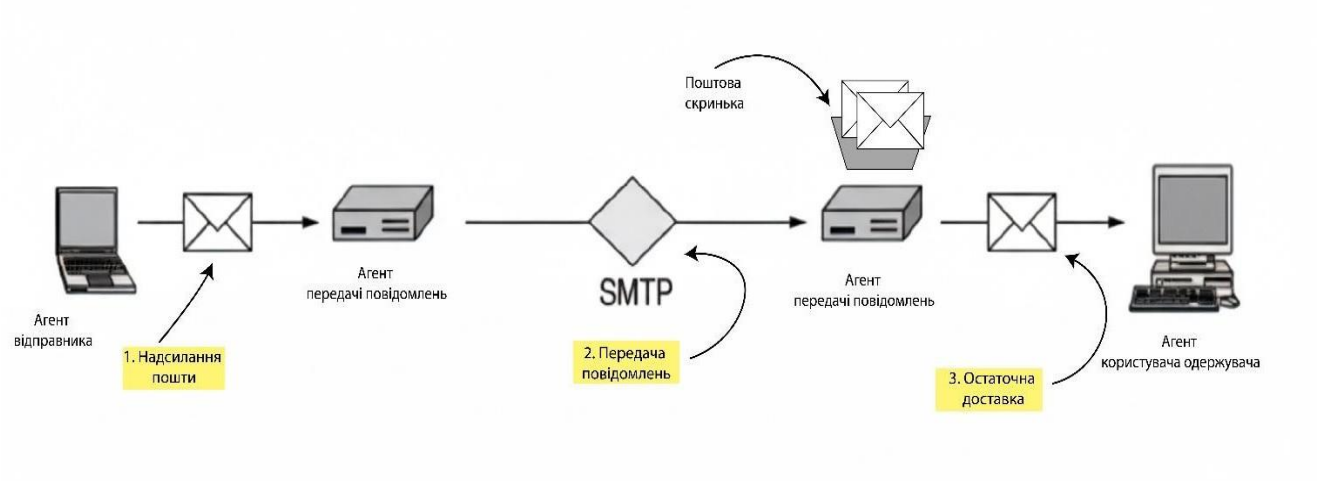


Рис. 1 Архітектура системи електронної пошти

Архітектура системи електронної пошти містить дві підсистеми: агенти користувача використовуються для читання, надсилання, створення, відповідей на повідомлення, відображення вхідних повідомлень та впорядкування повідомлень шляхом їх зберігання, пошуку та видалення. Прикладами найпоширеніших агентів користувача є Google Gmail, Microsoft Outlook, Mozilla та Apple Mail. (ii) Агенти передачі повідомлень використовуються для надсилання повідомлень від джерела до місця призначення за допомогою простого протоколу передачі пошти (SMTP). Вони також відомі як поштові сервери [20].

Електронна пошта — це засіб комунікації, спосіб надсилання та отримання повідомлень через Інтернет. Електронна пошта використовує кілька протоколів у рамках набору TCP/IP. Електронна пошта вже досить давно є надзвичайно важливим засобом комунікації, що дозволяє майже миттєво зв’язатися з будь-якою точкою світу за допомогою підключення до Інтернету [19].

1.2 Проблеми обробки даних і взаємодії з API при виявленні фішингових листів

Ландшафт загроз електронної пошти постійно розвивається. Щороку кіберзлочинці розробляють нові способи обману та атаки своїх жертв через електронну пошту. Контекст, сценарії та типи листів відрізняються, але основні загрози залишаються незмінними. Три основні загрози електронної пошти – це шкідливі вкладення, шкідливі URL-адреси та соціальна інженерія. Ці загрози призводять до втрати даних, крадіжки інформації, порушення роботи бізнесу та грошових втрат. Більшість фішингових листів містять одну або декілька з цих трьох загроз.

Шкідливі вкладення.

Відомо, що шкідливі вкладення електронної пошти містять шкідливе програмне забезпечення (Malware), яке може встановлювати віруси, троянські програми, шпигунські програми, боти, налаштовувати атаки програм-вимагачів, заражати файли Office за допомогою макросів або запускати розширені постійні

загрози (APT).

Шкідливе програмне забезпечення призначене для запуску під час відкриття вкладення електронної пошти. Воно може маскуватися під документи, PDF-файли, голосові повідомлення, електронні факси, зображення та інші типи файлів, які здаються надійними або цікавими.

У 2023 році Symantec повідомила про рівень шкідливих електронних листів 1 на 412, де 48% усіх шкідливих вкладень були файлами Office, такими як файли Word та Excel.

Шкідливі URL-адреси.

Шкідливий уніфікований локатор ресурсів (URL) – це посилання, на яке можна натискати, вбудоване в тіло або вкладення електронного листа [24]. Воно створюється з єдиною метою – компрометувати одержувача електронного листа. Шкідливі URL-адреси часто маскуються під зображення, кнопки або текст, які не відповідають цільовому використанню. Symantec повідомила у 2023 році, що, згідно зі зібраними даними за 2024 рік, шкідлива URL-адреса була знайдена в кожній 170-й URL-адресі електронного листа [25].

Натискання на шкідливу URL-адресу може призвести до завантаження та виконання шкідливих скриптів або встановлення шкідливого програмного забезпечення. Це також може бути веб-адреса, яка перенаправляє ціль на підроблений веб-сайт. Це робиться для того, щоб спонукати її невмисно та паролі, або викрити їх у небезпечному місці, де можна встановити шкідливе програмне забезпечення на їхній комп’ютер.

Соціальна інженерія, що використовується в електронних листах, передбачає форму психологічної маніпуляції, обманюючи нічого не підозрюючих одержувачів електронної пошти [26].

Така маніпуляція намагається викликати у жертви терміновість, страх або інтерес за допомогою тексту в електронному листі. Соціальна інженерія може спонукати жертву натискати шкідливі посилання, відкривати шкідливі файли або виконувати такі дії, як надання конфіденційної інформації чи переказ грошей незаконному джерелу. Запобігти таким загрозам може бути важко, оскільки вона

використовує людські помилки [26].

Фільтрація електронної пошти — це обробка електронних листів для їх перевпорядкування відповідно до певних стандартів [4]. Поштові фільтри зазвичай використовуються для керування вхідною поштою, фільтрації спаму, виявлення та видалення листів, що містять будь-які шкідливі коди, такі як віруси, трояни або шкідливі програми.

Фішингові листи зазвичай містять повідомлення соціальної інженерії з певними фразами таких як клік за URL-адресою. Тому вміст цих електронних листів є корисними функціями для виявлення фішингу. Розроблено дуже мало фільтрів фішингової електронної пошти, на відміну від багатьох існуючих фільтрів розроблені для спаму [27].

Архітектура фільтрації спаму електронною поштою.

Метою фільтрації спаму є зменшення до мінімуму кількості спонтанних електронних листів.

На роботу електронної пошти впливають деякі основні протоколи, включаючи SMTP. Деякі з широко використовуваних поштових агентів користувача (MUA) - це Mutt, Elm, Eudora, Microsoft Outlook, Pine, Mozilla Thunderbird, IBM Notes, KMail та Balsa [28]. Це поштові клієнти, які допомагають користувачеві читати та складати електронні листи.

Спам-фільтри можуть бути розгорнуті в стратегічних місцях як на клієнтах, так і на серверах. Спам-фільтри розгортаються багатьма інтернет-провайдерами (ISP) на кожному рівні мережі, перед поштовим сервером або на ретрансляції пошти, де є брандмауер [28].

Поштовий сервер служить інтегрованим антиспамовим та антивірусним рішенням, що забезпечує комплексний захід безпеки для електронної пошти на периметрі мережі. Фільтри можна впроваджувати в клієнтах, де їх можна встановлювати як доповнення на комп’ютерах, щоб вони служили посередниками між деякими кінцевими пристроями.

Фільтри блокують небажані або підозрілі електронні листи, які становлять загрозу безпеці мережі, від потрапляння до комп’ютерної системи. Також на рівні

електронної пошти користувач може мати налаштований файл спаму, який блокуватиме спам-листи відповідно до певних встановлених умов [29].

Як працюють фільтри спаму Gmail, Outlook та Yahoo використовують кілька методів фільтрації спаму для доставки лише легітимних електронних листів своїм користувачам та фільтрації нелегітимних повідомлень. І навпаки, ці фільтри також іноді помилково блокують справжні повідомлення. Повідомляється, що близько 20 відсотків електронних листів на основі авторизації зазвичай не потрапляють до папки "Вхідні" очікуваного одержувача [30].

Постачальники послуг електронної пошти розробили різні механізми використання у фільтрах спаму електронної пошти, щоб зменшити небезпеку, яку становлять фішинг, шкідливе програмне забезпечення електронної пошти та програми-вимагачі для користувачів електронної пошти.

Ці механізми використовуються для визначення рівня ризику кожного вхідного електронного листа. Серед таких механізмів є задовільні обмеження спаму, рамки політики відправників, білі та чорні списки, а також інструменти перевірки одержувачів [30].

Спам-фільтр Gmail.

Центри обробки даних Google використовують сотні правил для визначення класифікації електронної пошти, чи є лист шилом чи спамом. Кожне з цих правил відображає певні ознаки спаму, і з ним пов'язане певне статистичне значення, залежно від ймовірності того, що ця ознака є спамом. Зважена важливість кожної ознаки потім використовується для побудови рівняння.

Тест проводиться використовуючи оцінку відносно порогу чутливості, визначеного спам-фільтром кожного користувача. І, як наслідок, лист класифікується як легальний або небажаний. Вважається, що Google класифікує електронні листи за допомогою передових методів машинного навчання для виявлення спаму, таких як логістична регресія та нейронні мережі. Оптичне розпізнавання символів (OCR) також використовується Gmail для захисту користувачів від спаму зображень [31].

Gmail також може пов'язувати сотні параметрів для покращення виявлення

спаму завдяки алгоритмам машинного навчання, створеним для агрегації та ранжування величезних колекцій результатів пошуку Google.

Такі фактори, як репутація домену, посилання в заголовках повідомлень та інші, змінювали характер спаму з часом. Повідомлення можуть потрапити до кошика зі спамом внаслідок цих факторів. Фільтрація спаму базується на «фільтрах», які постійно оновлюються, оскільки виявляються нові інструменти, алгоритми та спам, а також коментарі користувачів Gmail щодо потенційних спамерів.

Багато спам-фільтрів використовують текстові фільтри для усунення загроз спамерів на основі відправників та їхньої історії.

Фільтр спаму в пошті Yahoo.

Yahoo Mail – перший у світі безкоштовний постачальник веб-пошти з понад 320 мільйонами користувачів [32]. Постачальник електронної пошти має власні алгоритми спаму, які він використовує для виявлення спам-повідомлень.

Основні методи, що використовуються Yahoo для виявлення спам-повідомлень, включають фільтрацію URL-адрес, вмісту електронної пошти та скарг на спам від користувачів [33].

На відміну від Gmail, Yahoo фільтрує електронні листи за доменами, а не за IP-адресами. Yahoo Mail використовує комбінацію методів для фільтрації спам-повідомлень. Він також надає механізми, які запобігають помилковому прийняттю дійсного користувача за спамера. Прикладами є здатність користувачів виправляти помилки SMTP, звертаючись до своїх журналів SMTP. Іншим є сервіс зворотного зв'язку щодо скарг, який допомагає користувачеві підтримувати позитивну репутацію в Yahoo. Також передбачено білий список Yahoo (внутрішній білий список та сертифікація зворотного шляху) [34].

На відміну від чорного списку, білий список блокує, дозволяючи користувачеві вказати список відправників, від яких потрібно отримувати пошту. Адреси таких відправників розміщуються у списку довірених користувачів.

Фільтри спаму в пошті Yahoo дозволяють користувачеві використовувати комбінацію білого списку та інших функцій боротьби зі спамом, щоб зменшити

кількість дійсних повідомлень, які помилково класифікуються як спам. Використання лише білого списку, з іншого боку, зробить фільтр надзвичайно жорстким, що означає, що будь-який неавторизований користувач буде автоматично заблокований. Автоматичне додавання до білого списку використовується кількома антиспам-системами.

У цій ситуації адреса електронної пошти анонімного відправника перевіряється в базі даних; якщо немає історії спаму, повідомлення доставляється до поштової скриньки одержувача, а відправник додається до білого списку.

Фільтр спаму електронної пошти Outlook.

Слідуючи за Gmail та Yahoo Mail, ми розглянули Microsoft Outlook та те, як він обробляє фільтрацію спаму в цій частині [35]. Hotmail та Windows Live Mail були перейменовані на Outlook.com компанією Microsoft у 2013 році.

Outlook.com базується на мові дизайну Metro від Microsoft та дуже нагадує інтерфейс Microsoft Outlook. Outlook.com від Microsoft – це набір програм, однією з яких є веб-пошта Outlook. Користувачі можуть надсилати та отримувати електронні листи через свій веб-браузер за допомогою веб-пошти Outlook. Це дозволяє користувачам підключати хмарні служби зберігання до свого облікового запису, щоб, коли вони хочуть надіслати електронний лист із вкладеними файлами, вони могли вибирати файли не лише зі свого комп’ютера та облікового запису OneDrive, але й з облікових записів Google Диска, Box та Dropbox.

Крім того, веб-пошта Outlook також дозволяє користувачам шифрувати свої електронні повідомлення та забороняти одержувачу пересилати електронний лист.

Щоразу, коли повідомлення шифрується в Outlook.com, лише особа з паролем зможе розшифрувати повідомлення та прочитати його. Це функція безпеки, яка гарантує, що повідомлення прочитає лише визначений одержувач.

Основна відмінність між класичною програмою MS Outlook та веб-службою пошти Outlook.com полягає в тому, що класична програма MS Outlook дозволяє надсилати та отримувати електронні листи через поштовий сервер, тоді як Outlook.com є поштовим сервером.

З іншого боку, веб-служба пошти Outlook.com розроблена для підприємств

та професіоналів, які покладаються на електронну пошту. Крім того, настільна програма MS Outlook – це комерційне програмне забезпечення, яке постачається разом з пакетом Microsoft Office. Це комп'ютерна програма, яка надає такі послуги, як керування електронною поштою, адресна книга, блокнот, веб-браузер та календар, що дозволяє користувачам планувати свої зустрічі та організовувати майбутні зустрічі. Outlook.com має майже 400 мільйонів користувачів [35].

Згідно зі статистикою, їхній сайт отримує приблизно вісім мільярдів електронних листів щодня, причому 30–35 відсотків цих листів надсилаються до поштових скриньок споживачів. Outlook.com має власні унікальні методи фільтрації спаму електронною поштою [35].

Розглянемо процес фільтрації спаму в електронній пошті.

Електронне повідомлення складається з двох основних компонентів: заголовка та тіла. Заголовок – це область, яка містить загальну інформацію про вміст електронного листа. Він включає тему, відправника та одержувача.

Тіло – це серце електронного листа. Воно може містити інформацію, яка не має попередньо визначених даних. Приклади включають веб-сторінку, аудіо, відео, аналогові дані, зображення, файли та розмітку HTML.

Заголовок електронного листа складається з таких полів, як адреса відправника, адреса одержувача або позначка часу, які вказують, коли повідомлення було надіслано проміжними серверами до агентів транспортування повідомлень (MTA), які функціонують як офіс для організації пошти.

На рисунку 2 зображено архітектуру поштового сервера та те, як здійснюється фільтрація спаму.

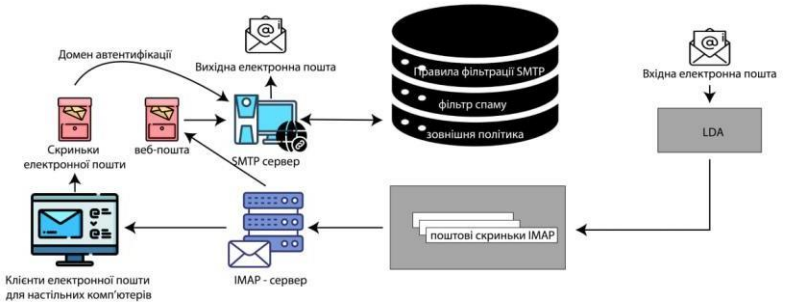


Рис. 2. Архітектура фільтрації спаму на поштовому сервері

Рядок заголовка зазвичай починається з поля «Від» і зазнає певних змін щоразу, коли він переміщується з одного сервера на інший через проміжний сервер. Заголовки дозволяють користувачеві переглядати маршрут, який проходить електронний лист, та час, який кожен сервер витрачає на обробку пошти. Наявна інформація має пройти певну обробку, перш ніж класифікатор зможе використовувати її для фільтрації [36].

1.3 Технічні підходи до оптимізації виявлення фішингових листів

Фішинг – одна з найпопулярніших форм злому, спроба отримати інформацію про обліковий запис та дані облікових даних користувача, видаючи себе за директиву, що надходить від законного джерела та авторитету, такого як надійна компанія чи організація. Фішингові електронні листи також є одним із найпростіших та найчастіше використовуваних методів. Фішинг є серйозною загрозою для всіх користувачів Інтернету, і його важко відстежити або захистити від нього, оскільки він не проявляє себе як явно шкідливий за своєю природою [6].

Деякі люди, відомі як кіберзлочинці, знайшли способи використовувати недоліки та помилки, виявлені в електронній пошті, використовуючи основні протоколи електронної пошти, функції електронної пошти та слабкі місця у взаємодії людини з машиною.

Це відомо як фішинг електронною поштою, вектор атаки кіберзлочинців, кількість інцидентів якого різко зросла за останні роки [37].

Фішинг електронною поштою можна описати як тип атаки соціальної інженерії, маніпуляції жертвою (одержувачем електронної пошти) з метою примусити її робити те, що хоче зловмисник (відправник електронної пошти) [38]. Це відбувається, коли зловмисник, маскуючись під довірену особу, обманює ціль, змушуючи її виконувати дії на основі вмісту небажаного електронного листа [39].

Також може бути натискання на шкідливі посилання або вкладення, знайдені в небажаному електронному листі. Це може призвести до встановлення шкідливого програмного забезпечення або крадіжки конфіденційної, особистої або фінансової

інформації та даних.

Фішинг електронною поштою найбільш відомий спробою викрасти особисту та фінансову інформацію, але він також використовується для компрометації комп'ютерів та IT-мереж на особистому, бізнес- та національному рівнях. Він служить шлюзом та ранньою фазою кіберзлочинних атак, що призводить до складніших та небезпечніших ситуацій [40].

Фішинг – це дія, спрямована на електронне отримання конфіденційної інформації від користувачів (зазвичай з метою шахрайства) шляхом створення копії веб-сайту легітимної організації [40].

Фішинг зазвичай здійснюється шляхом надсилання користувачам шахрайських та добре складених електронних листів.

Ці електронні листи зазвичай містять посилання на клоновані веб-сайти, і натискання на ці посилання може перенаправити користувачів на фішинговий веб-сайт або веб-сайт, що розміщує шкідливе програмне забезпечення. Веб-сайти, що розміщують шкідливе програмне забезпечення, зазвичай заражені шкідливими кодами, які можуть отримати доступ до особистої інформації користувачів, а також завдати шкоди комп’ютерам користувачів.

Через величезну кількість електронних листів, які отримують різні користувачі сьогодні, відокремлення легітимних електронних листів від фішингових є складним завданням, тому потребу в швидшому, надійному та ефективному методі фільтрації не можна переоцінити.

У наукових статтях було запропоновано кілька підходів, включаючи мережевий підхід, чорний список, білий список та контентний підхід. Мережеві підходи є дорогими для впровадження, складними в обслуговуванні та трудомісткими [41].

Підходи чорного списку (тобто список зареєстрованих фішингових веб-сайтів) та білого списку (тобто список цільових компаній) дають високі показники FP та FN; їхня ефективність обмежена інформацією, що зберігається в них. Це обмеження робить чорний список та білий список нездатними автоматично виявляти нові фішингові атаки в міру їх виникнення [42].

Робоча група з боротьби з фішингом (APWG) зазначила, що середній час безвідмовної роботи фішингового веб-сайту становить 44,39 години (тобто менше 2 днів) [43].

Контентний підхід спрямований на фіксацію контенту та структурних властивостей даних. За даними White et al, підхід чорного списку є широко використовуваним підходом до виявлення фішингу, який сьогодні приймають багато хто.

Тим не менш, Bergholz et al. зазначив, що метод на основі контенту є найточнішим і найбезпечнішим з усіх вищезгаданих методів виявлення фішингу. Це пояснюється тим, що метод на основі контенту здатний виявляти нові шахрайські моделі у великих наборах даних у міру їх розвитку [45].

Фішинг – це проблема класифікації, і Мартін та ін. [46] окреслили п'ять етапів фішингових атак. А саме:

1. Планування: на цьому етапі визначається, хто має бути цільовою організацією та як отримати адресу електронної пошти клієнтів організації.

2. Налаштування: розробляється метод надсилання повідомлень (зазвичай масова розсилка) та отримання розкритої інформації про користувача.

3. Атака: на цьому етапі шахрайське та оманливе повідомлення надсилається на адреси користувачів.

4. Збір: фіксується інформація про користувачів-жертв.

5. Атака: відбувається фактичне шахрайство з використанням отриманої інформації, розкритої користувачами на етапі збору.

Виявлення фішингу підходило різними способами. Адіда та ін. [47] припустили, що фішинг можна вирішувати та усувати на рівні електронної пошти, оскільки багато шахраїв використовують електронну пошту як інструмент для скоєння шахрайства.

Дхаміджа та Тайгар [48] також припустили, що електронну пошту можна усувати на рівні веб-сайту. Вони запропонували інтегрувати панель інструментів безпеки у веб-браузери.

Інший підхід, запропонований Dynamic Security Skins [49], передбачає

використання візуального хешу. У цьому підході візуальний хеш генерується випадковим чином і використовується для налаштування веб-версії браузера та версії Windows. Візуальний хеш відповідає за ідентифікацію веб-сайтів, які були успішно автентифіковані браузером.

Бантін також запропонував метод під назвою «Криптографічна перевірка ідентичності [50]. Автор зазначив, що цей метод може працювати лише за умови зміни всієї веб-інфраструктури (як серверів, так і клієнта). Крім того, підвищення обізнаності користувачів може збільшити пом'якшення наслідків зловмисної атаки; користувачі повинні бути добре навчені різним способам виявлення фішингових веб-сайтів.

Хонджі [51] підсумовує підходи, які можна застосувати для боротьби з фішинговими атаками за чотирма категоріями, а саме: наступальний захист, корекція, запобігання та виявлення відповідно.

A) Атакуючі підходи захисту.

Метою підходів, що належать до цієї категорії, є нейтралізація ефекту фішингової атаки.

Цей метод, здебільшого, застосовується до користувачів, які вже стали жертвою атаки (тобто користувачів, які вже заповнили та надіслали свою особисту інформацію в HTML-форми фішингового веб-сайту).

За цього підходу, щоразу, коли користувача вводять в оману на фішинговий веб-сайт, програмне забезпечення, встановлене в браузері користувача, також надсилає кілька підроблених зразків інформації на фішинговий веб-сайт, тому зловмисникам буде важко знайти фактичну інформацію, надану користувачем.

B) Підходи до виправлення.

Підходи цієї категорії спрямовані або на видалення фішингових файлів з веб-сайту, або на зроблення фішингового веб-сайту недоступним. Обидва ці кроки можна досягти, повідомивши провайдера інтернет-послуг, який розмістив веб-сайт, щоб він вжив відповідних або необхідних заходів.

C) Підходи до запобігання.

Ці підходи спрямовані як на запобігання тому, щоб користувачі стали жертвами,

так і на те, щоб фішери не могли обманювати користувачів у майбутньому. Останнього можна досягти, залучивши правоохоронні органи, які можуть проводити розслідування та карати зловмисників, змушуючи їх дорого платити за свої злочини. Це служить стримуючим фактором і, в свою чергу, мінімізує подальші атаки.

D) Методології виявлення.

Основною метою методологій цієї категорії є розпізнавання фішингових атак та класифікація їх як легітимних або нелегітимних.

Зазвичай це досягається шляхом сканування кожного електронного листа на наявність сотень підозрілих фішингових ознак та їх автоматичної фільтрації.

Аналіз фішингових ознак дозволяє системі виявлення реагувати на нові фішингові атаки в міру їх появи.

Метод виявлення за чорним списком, підхід виявлення за білим списком, підхід виявлення на основі мережі та підхід виявлення на основі контенту – це чотири типи підходів до виявлення. Розглянемо чотири типи підходів:

Підхід до виявлення у чорному списку.

Термін «чорний список» стосується списку фішингових веб-сайтів, про які повідомлялося. Щоб визначити фішингові адреси у чорному списку, деякі інтернет-провайдери (ISP), онлайн-браузери та постачальники послуг електронної пошти (такі як Gmail, Yahoo, Microsoft та інші) використовують стратегію чорного списку.

Компанії використовують інформацію з чорного списку для захисту своїх систем і, як наслідок, захисту своїх клієнтів від фішингових атак. Якщо електронний лист надіслано з IP-адреси, яка вже була внесена до чорного списку, постачальник електронної пошти може або відмовитися доставляти електронний лист, або надіслати його до папки спаму одержувача.

Чорний список зазвичай містить доменні імена та IP-адреси раніше виявлених фішингових веб-сайтів.

Деякі чорні списки також містять ключові слова, IP-адреси відкритих проксі-серверів та ретрансляторів, IP-адреси інтернет-провайдерів, які розміщують фішингові веб-сайти, та порушників RFC (IP-адреси, що порушують стандарти

Інтернету та мережевої інженерії).

Алмомані та ін. [52] повідомляють, що існує понад 20 чорних списків спаму, які зазвичай використовуються, і ці чорні списки зазвичай оновлюються через регулярні проміжки часу; наприклад, чорний список браузера Firefox (що зберігається в профілі користувача) зазвичай оновлюється кожні 30 хвилин [53].

Підхід до виявлення за допомогою білого списку.

Білий список – це список компаній, на які було націлено атаку (наприклад, eBay, Paypal, Visa тощо). Підходи до білого та чорного списків досить схожі тим, що обидва захищають користувачів від шахрайських атак. Інформація, що міститься як у білому, так і в чорному списках, є основною відмінністю між ними.

Білий список – це набір адрес електронної пошти, IP-адрес та доменних імен, вільних від спаму. Різні постачальники, як правило, використовують білий список, щоб впливати на свої рішення щодо фільтрації.

Наприклад, мережевий адміністратор організації може створити білий список MAC-адрес (Media Access Regulate) та використовувати його для контролю доступу до мережі. Крім того, деякі спам-фільтри підтримують білий список адрес електронної пошти, IP-адрес та доменних імен, які вони використовують для визначення автентичності електронного листа.

Мережевий підхід до виявлення.

Цей підхід використовується різними мережевими адміністраторами для захисту своєї мережі від вторгнення.

Зазвичай, коли користувач надсилає повідомлення через мережу, воно форматується в менший блок, який називається пакетом і містить повідомлення, надіслане користувачем, та IP-адресу мережі-відправника.

Однак IP-адресу можна підробити таким чином, щоб вона була прихована. Зазвичай, мережевий підхід спрямований на блокування будь-якого мережевого пакета, який вважається незаконним (тобто пакетів, що містять замасковані IP-адреси).

Підхід до виявлення на основі контенту.

Підхід на основі контенту – ще один метод, який можна використовувати для

виявлення шахрайських атак. Цей підхід передбачає аналіз вмісту та структурних властивостей даних. Наприклад, Microsoft Internet Explorer (версія 7) має вбудований класифікатор, який аналізує вміст веб-сторінок та фільтрує їх на основі певних критеріїв [53].

Берггольц та ін. зазначили, що підхід на основі контенту є найефективнішим та найбезпечнішим з усіх підходів до фільтрації, хоча також зазначили, що підхід чорного списку є найбільш широко використовуваним підходом [53].

Злочинці мають безліч методів та типів фішингових електронних листів, щоб обдурити користувачів електронної пошти. Різниця між методами фішингу електронною поштою полягає в контексті та сценаріях, у яких використовуються загрози.

Хакери розсилають шахрайські електронні листи буквально мільйонам людей, сподіваючись, що деякі з них натиснуть на додані посилання, документи чи зображення, з метою спонукати одержувачів добровільно надавати цінну конфіденційну інформацію, таку як номери соціального страхування, паролі, банківські номери, PIN-коди, номери кредитних карток тощо. Цього можна досягти кількома різними методами. Фішингові електронні листи можна розділити на 10 категорій [53], а саме:

Урядовий маневр.

Цей тип електронного листа виглядає так, ніби він надійшов від федерального органу, такого як Microsoft, і намагається залякати людей, щоб вони надали свою інформацію. Звичайні повідомлення включають: «У вашій страховці відмовлено через неповну інформацію. Натисніть тут, щоб надати інформацію».

Або «Оскільки ви незаконно завантажили файли, ваш доступ до Інтернету буде скасовано, доки ви не введете запитану інформацію у форму нижче».

Тактика друга.

Якщо невідома особа стверджує, що знає вас в електронному листі, ви, ймовірно, не страждаєте від амнезії.

Швидше за все, це спроба змусити вас переказати йому/їй гроші. Варіацією на цю тему є те, що один із ваших відомих друзів знаходиться в чужій країні та

потребує вашої допомоги.

Перш ніж ви надсилаєте гроші своєму «другу», зателефонуйте йому, щоб перевірити. Список контактів електронної пошти вашого справжнього друга, ймовірно, був викрадений.

Проблема з виставленням рахунків.

Фішингова тактика є складною, оскільки вона виглядає цілком законною. У цьому електронному листі повідомляється, що товар, який ви замовили онлайн, не буде доставлено вам, оскільки термін дії вашої кредитної картки закінчився (або ваша платіжна адреса неправильна тощо). Якщо ви натиснете на надане посилання, вас буде перенаправлено на підроблений веб-сайт, який, серед іншого, запитує оновлену інформацію про оплату/доставку.

Термін дії у цьому типі електронного листа неправдиво пояснюється, що термін дії вашого облікового запису в [назва компанії] незабаром закінчиться, і ви повинні увійти якомога швидше, щоб уникнути втрати всіх своїх даних. Зручно, що в електронному листі є посилання, яке знову ж таки перенаправляє вас на підроблену сторінку входу.

Вірус або скомпрометований обліковий запис.

У цих електронних листах повідомляється, що ваш комп’ютер заражено або що один із ваших облікових записів був зламаний. Щоб уникнути втрати грошей, даних або зараження комп’ютера, в електронному листі вам пропонується перейти за посиланням для завантаження вкладення.

Переможець конкурсу.

Не надто радійте, коли отримуєте електронні листи, в яких стверджується, що ви щось виграли або отримали спадщину від родича, про якого ви ніколи не чули. У 99,9% випадків це абсолютно фальшиві листи.

Щоб отримати свій приз, в електронному листі потрібно натиснути на посилання та ввести свою інформацію для його доставки.

Дружній банк.

Ваш банк може пропонувати сповіщення про стан рахунку, коли з ваших рахунків знімаються певні суми.

Цей трюк обманює вас фальшивим сповіщенням про те, що з вашого рахунку було знято суму, яка перевищує ваш ліміт сповіщень.

Якщо у вас є якісь запитання щодо цього зняття (що, ймовірно, ви б і мали), вам буде надано зручне посилання, яке веде до веб-форми, де запитується номер вашого банківського рахунку «для цілей перевірки».

Замість того, щоб натискати на посилання, зателефонуйте до свого банку. Можливо, вони захочуть вжити заходів щодо шкідливого електронного листа.

Жертва.

Коли вас у чомусь помилково звинувачують, це неприємно. Цей тип фішингового електронного листа діє як розгніваний клієнт, який нібито надіслав вам гроші в обмін на відправлений товар. Електронний лист закінчується погрозою, що вони повідомлять органи влади, якщо не отримають від вас відповіді.

Податкове повідомлення: Практично кожен має щорічну податку. Ось чому ця спроба фішингу така популярна. У повідомленні зазначається, що ви або маєте право на отримання податкового відшкодування, або вас обрали для перевірки. Потім у ньому пропонується подати запит на податкове відшкодування або податкову форму.

Перевірка.

Це одна з найскромніших спроб фішингових електронних листів. У ній стверджується, що [назва компанії] проводить стандартну процедуру безпеки та просить вас підтвердити свій обліковий запис, надавши інформацію. Ця афера особливо ефективна, якщо ви є клієнтом цієї компанії.

2 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

2.1 Класифікація алгоритмів машинного навчання для електронної пошти

Підхід машинного навчання був широко вивчений і існує багато алгоритмів, які можна використовувати для фільтрації електронної пошти.

Вони включають випадковий ліс, логістичну регресію, методи опорних векторів, наївний байєсівський алгоритм, нейронні мережі та багатошаровий перцептрон, адаптивні та бустінгові.

Випадковий ліс.

Класифікатор випадкових лісів (RF) – це класифікатор, який використовує дерева рішень. Він генерує велику кількість дерев рішень; кожне дерево використовує випадкову кількість вибірок та ознак з нашого набору даних. Якщо вводяться класифіковані дані, класифікатор повертає мітку, яка була визначена найбільшою кількістю дерев рішень [54].

Логістична регресія.

Логістична регресія (LogReg) – це добре зарекомендувала себе статистична модель для класифікації даних. Вона бінарно класифікує дані, підганяючи точки даних до логістичної функції.

Класифікатор дуже потужний для простих, лінійно роздільних даних, але його продуктивність починає знижуватися для даних зі складними зв'язками між змінними [54].

Нейронні мережі та багатошаровий перцептрон.

Нейронні мережі – це тип класифікатора, який намагається імітувати біологічний мозок. Мережа складається з пов'язаних шарів так званих вузлів, які нагадують нейрони, як ми їх знаємо в біології. Хоча нейрони не можуть багато зробити самі по собі, введення належної кількості пов'язаних нейронів дозволяє оцінювати складні функції. Деякі з найпростіших, хоча й дуже корисних структур, можуть емулювати логічні вентилі [54].

Трохи складніші мережі здатні класифікувати лінійно роздільні дані, але розумне маніпулювання вхідними ознаками (тобто за допомогою функцій ядра) дозволяє нам обійти навіть це обмеження [54].

Нейронні мережі є високо налаштовуваними шляхом встановлення гіперпараметрів, таких як кількість прихованих шарів, кількість нейронів на шар та алгоритм оптимізації. Налаштування гіперпараметрів нейронних мереж – це нетривіальне завдання, яке зазвичай вимагає багато досвіду та знань, хоча вони часто можуть ініціалізувати модель, роблячи деякі обґрунтовані припущення, та вдосконалюватися звідти [55].

Хоча існує багато різних типів нейронних мереж, їхні дослідження зосереджені на багатошаровому перцептроні (MLP), який є типом нейронної мережі прямого зв'язку. У нейронних мережах прямого зв'язку вузли не утворюють цикл, тобто нейрони виводять дані лише до наступного шару нейронів.

На Рисунку 3. Представлено Процес фільтрації фішингу на основі алгоритму SVC.

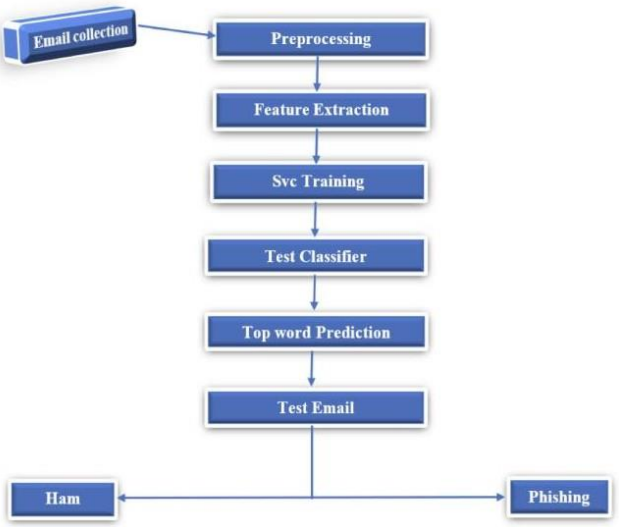


Рис. 3. Процес фільтрації фішингу на основі алгоритму SVC

1. Крок попередньої обробки було використано для видалення з електронного листа шумів, які є нерелевантними та не повинні бути доступними. Крок попередньої обробки включає:

2. Видалення чисел

b. Видалення спеціальних символів

c. Видалення URL-адрес

d. Видалення HTML

e. Визначення кореневих слів.

2. Вилучення ознак було використано для відділення основних та важливих ознак від тіла електронного листа. Ця функція перетворює електронний лист у двовимірний векторний простір з певною кількістю ознак.

3. На кроці навчання SVM для потреб навчання використовувалися електронні листи зі спамом.

Багатошаровий персептрон — одна з найбазовіших версій нейронної мережі, що складається лише з вхідного шару, налаштовуваної кількості прихованих шарів та вихідного шару [55].

Методи опорних векторів.

Методи опорних векторів – це моделі навчання з учителем та пов'язаними з ними алгоритмами навчання, які аналізують дані, що використовуються для класифікації та регресійного аналізу. Вони мають високу точність, хороші теоретичні гарантії щодо перенавчання, а з відповідним ядром можуть добре працювати, навіть якщо дані не є лінійно роздільними в базовому просторі ознак. Особливо популярні в задачах класифікації тексту, де нормою є дуже багатовимірні простори.

Набір даних для навчання включає вміст спаму, і класифікатор був підготовлений з його використанням.

Після навчання класифікатор був підготовлений до класифікації спам-листів.

4. Класифікатор було протестовано на четвертому кроці, який є кроком тестування класифікатора з різною навчальною інформацією для перевірки точності класифікатора.

5. На п'ятому кроці, який є кроком тестування електронної пошти, після завершення етапу навчання, приклад електронної пошти було надано як вхідні дані для класифікатора для характеристики електронної пошти.

Адаптивне підвищення.

AdaBoost, або адаптивне підвищення, – це метод машинного навчання, який поєднує набір менш ефективних класифікаторів для створення сильнішого.

Класифікатор вибирає «команду» інших, простіших класифікаторів, таких як SVM (2.1) та Random Forests (2.2), і надає кожному з них вагу.

Окремі класифікатори самостійно вирішують проблему та голосують за рішення на основі своїх прогнозів.

Потім алгоритм AdaBoost оцінює їхню продуктивність і перепризначає вагу кожному класифікатору на основі використаних критеріїв.

Цей цикл повторюється, доки критерії зупинки не будуть задоволені.

Найефективніша комбінація класифікаторів разом з їхніми вагами буде нашим остаточним класифікатором [55].

Наївний алгоритм Баєса.

Наївний алгоритм Баєса – це простий ймовірнісний класифікатор, який обчислює набір ймовірностей шляхом підрахунку частоти та комбінації значень у заданому наборі даних.

Наприклад, якщо фішингові електронні листи отримані через наявність ключових слів фішингових електронних листів, то певне ключове слово може бути використане для точнішої оцінки ймовірності того, що певний електронний лист справді є фішинговим листом, порівняно з оцінкою ймовірності фішингових електронних листів, надісланих без врахування цього конкретного ключового слова.

У дослідженні наївний баєсівський класифікатор використовує ознаки мішка слів для ідентифікації фішингових електронних листів, а текст представляється як мішок його слів.

Мішок слів завжди використовується в методах класифікації документів, де частота зустрічальності кожного слова використовується як ознака для навчання класифікатора.

Наївний баєсівський метод використовував теорему Баєса для визначення ймовірності небажаної електронної пошти. Деякі слова мають певні ймовірності зустрічатися в небажаній пошті або хакерській електронній пошті.

Наприклад, припустимо, що ми точно знаємо, що слово «Безкоштовно» ніколи не може зустрічатися в електронному листі, який не є спамом.

Тоді, коли ми побачимо повідомлення, що містить це слово, ми зможемо точно сказати, що це спам-лист.

Баєсівські спам-фільтри виявили дуже високу ймовірність спаму для таких слів, як «Безкоштовно» та«Віагра», але дуже низьку ймовірність спаму для слів, що зустрічаються в не-спам-листах, таких як імена друзів та членів родини.

Отже, для розрахунку ймовірності того, що електронний лист є спамом чи не є спамом, наївний баєсівський метод використовував теорему Баєса, як показано у формулі нижче [56].

$$P(spam|word) = \frac{\mathrm{P(spam).\ P(word|spam)}}{\mathrm{P(spam).P(word|spam)+P(non-spam).P(word|non-spam)}}$$

Р(спам|слово) =

Р(спам). Р(слово|спам)

Р(спам).Р(слово|спам)+Р(не-спам).Р(слово|не-спам) (2.1)

Де:

Р(спам_слово) – ймовірність того, що електронний лист містить певне слово, враховуючи, що електронний лист є спамом.

Р(спам) – ймовірність того, що будь-яке повідомлення є спамом.

Р(словос_спам) – ймовірність того, що це конкретне слово з'являється в спам-повідомленні.

Р(не-спам) – ймовірність того, що будь-яке конкретне слово не є спамом.

Р(словос_не-спам) – ймовірність того, що це конкретне слово з'являється в не-спам-повідомленні.

На рисунку 4 представлено Процес фільтрації фішингу на основі наївного баєсівського алгоритму.

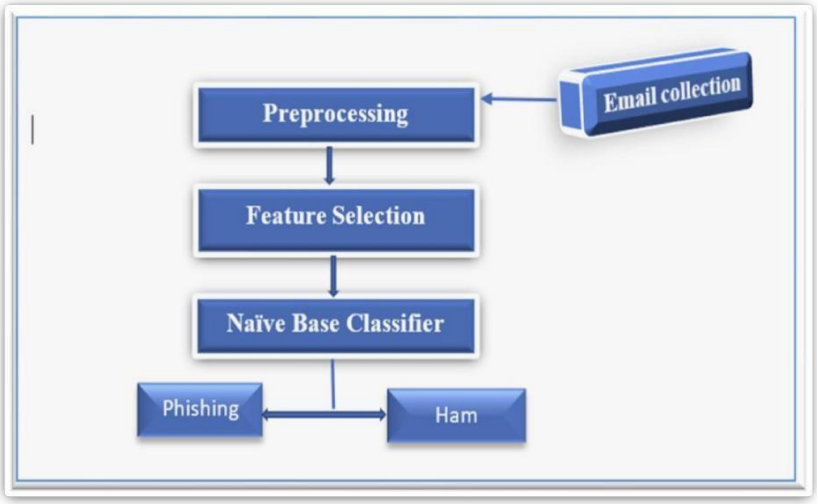


Рис. 4. Процес фільтрації фішингу на основі наївного баєсівського алгоритму

У нашій роботі ми обрали гібридний алгоритм машинного навчання, а саме наївний алгоритм Баєса та SVM, для розробки ефективного методу виявлення фішингу, оскільки:

Згідно з найпопулярнішим класифікатором машинного навчання, наївний Баєс має дуже задовільну продуктивність серед інших методів для досягнення вищої точності [57].

Ймовірнісний класифікатор, заснований на теоремі Баєса з сильними (наївними) припущеннями незалежності між ознаками. Він вимагає лише невеликої кількості навчальних даних для оцінки параметрів класифікації. Класифікатор добре працює у випадку категоріальних вхідних змінних порівняно з числовими змінними.

Наївна модель Баєса проста у побудові та особливо корисна для дуже великих наборів даних.

Поряд з простотою, SVM показав, що перевершує навіть найсучасніші алгоритми класифікації.

Застосування наївних баєсівських алгоритмів.

Прогнозування в реальному часі: Наївний баєсівський алгоритм – це захопливий класифікатор, що навчається, і він, безумовно, швидкий. Таким чином, його можна використовувати для прогнозування в реальному часі.

Багатокласове прогнозування: Цей алгоритм також добре відомий своєю

функцією багатокласового прогнозування. Тут ми можемо передбачити ймовірність кількох класів цільової змінної.

Класифікація тексту/Фільтрація спаму/Аналіз настроїв: Наївні баєсівські класифікатори, які здебільшого використовуються в класифікації тексту (завдяки кращому результату в багатокласових задачах та правилу незалежності), мають вищий рівень успішності порівняно з іншими алгоритмами. Як результат, він широко використовується в фільтрації спаму (виявлення спам-електронної пошти) та аналізі настроїв (в аналізі соціальних мереж для виявлення позитивних та негативних настроїв клієнтів).

Система рекомендацій: Наївний баєсівський класифікатор та спільна фільтрація разом створюють систему рекомендацій, яка використовує методи машинного навчання та аналізу даних для фільтрації невидимої інформації та прогнозування того, чи сподобається користувачеві певний ресурс чи ні.

2.2 Огляд поточних методів фільтрації

Згідно з дослідженнями Ханіфа Бхуіяна [59], у яких використовувалися різні методи фільтрації спаму електронною поштою, такі як методи, засновані на знаннях, методи кластеризації, методи, засновані на навчанні, евристичні процеси тощо.

У роботі наведено огляд різних існуючих систем фільтрації спаму електронною поштою, щодо методів машинного навчання, таких як наївний баєсівський метод, метод SVM, метод K-найближчих сусідів, адитивна баєсівська регресія, дерево KNN. Серед усіх існуючих методів фільтрації спаму електронною поштою, деякі є ефективними, а деякі намагаються впровадити інший процес для підвищення їхньої точності.

У таблиці 1 наведено огляд поточних методів фільтрації небажаної пошти на основі автора, алгоритму, який вони використовували для класифікації, набору даних та показників точності.

Таблиця 1

Огляд поточних методів фільтрації

№	Автор(и)	Алгоритми	Корпус або набори даних	Точність / Результативність
1	Мохаммед та ін.	Наївний Байєс, SVM, KNN, Дерево рішень, Правила	Email-1431	Досягнуто 85,96% точності
2	Субраманіам та ін.	Наївний Байєсівський класифікатор	Колекція спам-листів з облікового запису Google Gmail	Досягнуто 96,00% точності
3	Шарма та ін.	Адаптації різних алгоритмів машинного навчання	SPAMBASE	Досягнуто 94,28% точності
4	Бандей та ін.	Наївний Байєс, K-найближчих сусідів, SVM, Адитивне регресійне дерево Байєса	Реальний набір даних	Досягнуто 96,69% точності
5	Авад та ін.	Наївний Байєс, SVM, k-найближчих сусідів, Штучні нейронні мережі, Грубі множини	Spam Assassin	Досягнуто 99,46% точності
6	Чабра та ін.	Нелінійний класифікатор SVM	Набір даних Enron	Задовільні показники повноти (Recall) та точності (Precision) при співвідношенні спаму до реальних листів 1:3
7	Третьяков та ін.	Байєсівська класифікація, k-NN, ШНМ, SVM	Корпус PU1	Досягнуто 94,4% точності
8	Шахі та ін.	Наївний Байєс, SVM	Непальські SMS	Досягнуто 92,74% точності
9	Каул та ін.	SVM	Вибірка електронних листів	Досягнуто точності в межах 90% ~ 95%
10	Суганья та ін.	Метод на основі правил	Пости користувачів соціальних мереж (OSN)	Відмінна точність для наданих наборів даних
11	Ратхі та ін.	Наївний Байєс, мережа Байєса, SVM та випадковий ліс (Random Forest)	Власна колекція	Досягнуто 99,72% точності
12	Мохаммед та ін.	Наївний Байєс, k-найближчих сусідів, SVM, Штучна нейронна мережа	Nielson Email-1431	Задовільна точність для запропонованого методу
13	Сінгх та ін.	Різні алгоритми машинного навчання	Власна колекція	Повідомлено про покращення точності

				(precision) щонайменше на 2%
14	Абдулхамід та ін.	Наївний Байєс, SVM	Репозиторій машинного навчання UCI та власна колекція	Досягнуто 94,2% точності
15	Сах та ін.	Кастомізований SVM	Публічний корпус Apache	Повідомлено про хорошу загальну точність
16	Верма та ін.	Модифікований Наївний Байєс із вибірковими ознаками	Spam Base, Spam Data	Повідомлено про 98% точності
17	Русланд та ін.	Платформа Microsoft Azure (дерево рішень та SVM)	Власна колекція	Spam Base: 88% точності (Precision); Spam Data: 83% точності
18	Іксель та ін.	Наївний Байєс з інженерією ознак	Корпус SMS-спаму v.0.1	Точність SVM: 97,6%; Точність дерева рішень: 82,6%
19	Чоудхарі та ін.	Алгоритм випадкового лісу (Random Forest)	Власна колекція	96,5% справжніх позитивних результатів (True Positive Rate)
20	ДеБарр та ін.	Алгоритм випадкового лісу	Власна колекція	Досягнуто 95,2% точності
21	Кумар та ін. (2012)	Дерево рішень	Spam Base UCI	99%
22	Войташек та ін.	Простий SVM з персоналізованим словником	Електронна пошта	95,26%
23	Чжао та Чжан	На основі теорії грубих множин	Власна колекція	97,37%

Також детально розглядаються різні існуючі системи фільтрації спаму за допомогою методів машинного навчання, досліджуючи кілька методів, завершуючи огляд кількох методів фільтрації спаму та підсумовуючи точність різних запропонованих підходів щодо кількох параметрів.

2.3 Проблеми існуючих методів фільтрації елктронної пошти при виявленні фішингових листів

Використання єдиного системного методу замість гібридних систем є однією

з проблем.

Гібридні методи виглядають найефективнішим способом створення успішного антиспам-фільтра на сьогодні.

Швидка адаптація спамерів до спаму/фішингу для отримання особистої інформації про користувача для шахрайських пропозицій та негнучкість спам-фільтрів для адаптації до змін.

Підходи машинного навчання є добре відомими підходами, що забезпечують кращі методи, здатні контролювати небажану пошту, але через динамічний характер Інтернету у світі немає100% безпечних систем, які могли б впоратися з цією проблемою.

Більшість існуючих методів фільтрації електронної пошти не призначені спеціально для фішингової електронної пошти, натомість вони намагаються розрізняти спам-листи та фальшиві електронні листи, останні також відомі під назвою фальшиві електронні листи [59].

Інший дослідник виявив, що модель «пакета слів» є відносно ефективною функцією для фільтрації спаму та фішингових листів, а заголовки електронних листів є функціями, які є такими ж важливими, як і тіло повідомлення, для виявлення спаму.

У деяких дослідженнях використання теми, заголовка та тіла повідомлення розглядалося як найважливіша функція для класифікації повідомлень як спаму або фальшивих. Однак варто зазначити, що підозріла тема, заголовок та тіло листа самі по собі можуть призвести до помилки в класифікації спаму.

Користувачам також може знадобитися вибирати функції вручну.

Деякі статті зосереджувалися на методах фільтрації спаму електронною поштою без використання функцій, оскільки вони довели вищу точність, ніж метод на основі функцій.

Однак слід зазначити, що методи без використання функцій мають високі обчислювальні витрати, оскільки зазвичай займають набагато більше часу на завдання класифікації електронних листів. Вони також страждають від складності реалізації.

Деякі дослідники використовували поведінкові моделі спамерів як важливий аспект виявлення спаму, тоді як алгоритми машинного навчання використовувалися для вилучення важливих ознак з тіла повідомлення. Для кращої точності може знадобитися комплексна інженерія ознак.

Ми використовували гібридні методи для алгоритму класифікації (тобто наївний алгоритм Баєса та SVM).

Обидва мають покращену продуктивність серед інших методів машинного навчання. Через швидке розвиток методів атак спамерів, дослідники час від часу впроваджують, моделюють та розробляють механізми виявлення фішингової електронної пошти відповідно. Інша річ полягає в тому, що зараз спамери можуть легко обійти всі ці програми фільтрації спаму.

Це не тому, що фільтри недостатньо потужні, а через швидку адаптацію нових методів спамерами та негнучкость спам-фільтрів для адаптації змін.

Для вилучення ознак поведінкових моделей, важливих для алгоритму машинного навчання, може знадобитися комплексна інженерія ознак для кращої точності.

Опис програмної реалізації.

Методологія, що використовується в роботі, спочатку представляє собою комплексний огляд літератури щодо ознак, які використовувалися для виявлення фішингових електронних листів, а також методів інтелектуального аналізу даних, що використовуються.

Крім того, проведено аналіз ознак фішингу, знайдених в існуючих дослідженнях у літературі. Набір даних може бути взято з даних електронної пошти Ethio Telecom. Набір даних складається зі спам-листів, фішингових листів та електронних листів радіоаматорів. Після попередньої обробки набору даних (на основі методів категоризації тексту NLP) за допомогою комбінації NLP та наївного баєсівського алгоритму класифікації використовується для навчання моделі класифікації, таким чином, фільтрація перетворюється на проблему класифікації.

Етапи, використані для нашого дослідження, проілюстровані наступним чином: попередня обробка, токенізація, вилучення ознак, вибір ознак, наївний

базовий класифікатор та класифікатор опорних векторів, тестовий класифікатор, нарешті, ідентифікують спам/фішинг та радіоаматорів.

Збір електронної пошти.

Для створення колекції даних можна використовувати дані електронної пошти Ethio Telecom та зразок фішингових даних онлайн. Загалом було зібрано 5229 електронних листів, з яких 4115 легітимних листів та 1114 фішингових листів складають набір даних. Дані збиралися між 2023 і 2024 роками.

Архітектура запропонованого методу.

У цьому розділі ми детально описуємо класифікатор Naïve Base та SVM, що використовуються в нашій статті, а також запропонований набір ознак, які ми виділили. Архітектура запропонованого виявлення фішингових електронних листів наведено на рисунку 5.

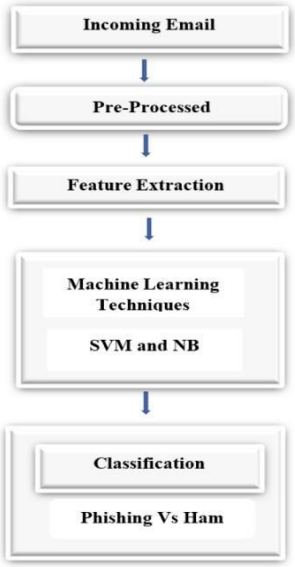


Рис. 5. Архітектура запропонованого алгоритму

Ми виділили набір ознак з чотирьох частин, включаючи заголовок електронної пошти, URL-адресу електронної пошти, тіло електронної пошти та функції сценарію електронної пошти. Потім ми вибираємо класифікатор Naïve Base та SVM для виявлення фішингових електронних листів.

Попередня обробка.

Попередня обробка даних – це процес підготовки необроблених даних та їх

придатності для моделі машинного навчання. Вона є вирішальним кроком для покращення якості даних, що сприяє вилученню змістовних знань з даних. У машинному навчанні попередня обробка даних стосується навичок організації необроблених даних, щоб зробити їх придатними для побудови та навчання моделей машинного навчання. Іншими словами, попередня обробка даних у машинному навчанні – це техніка інтелектуального аналізу даних, яка перетворює необроблені дані у зрозумілий та читабельний формат.

Попередня обробка даних – це крок, на якому дані трансформуються або кодуються, щоб привести їх до такого стану, щоб машина могла легко їх проаналізувати. Попередня обробка даних підвищує точність та ефективність моделі машинного навчання. Іншими словами, характеристики даних тепер можуть бути легко інтерпретовані алгоритмом.

У нашій роботі ми проілюстрували 5 кроків попередньої обробки.

Видалення пробілів.

Чистий текст часто означає токени або список слів, з якими можуть працювати наші моделі машинного навчання.

Це означає перетворення необробленого тексту на список слів та його повторне збереження. Дуже простий спосіб зробити це – розділити документ за пробілами, включаючи «», символи нового рядка, табуляцію та деякі інші.

Ми можемо досягти цього в Python за допомогою функції split() на завантаженому рядку.

Видалення пунктуації.

У процесі видалення пунктуації ми спочатку визначаємо рядок пунктуації. Потім нам потрібно перебрати наданий рядок за допомогою циклу for, де ми перевіряємо, чи є символ знаком пунктуації чи ні, використовуючи тест на належність. У нас є порожній рядок, до якого ми об'єднуємо символ, якщо він не є знаком пунктуації. Нарешті, ми відображаємо очищений рядок.

Токенізація.

Токенізація означає розділення більших текстових даних, есе або корпусів на менші сегменти. Ці менші сегменти можуть бути у формі менших документів або

рядків текстових даних, іншими словами, це засіб перетворення речення на послідовність слів, щоб обробку слово за словом можна було легко виконувати.

Враховуючи послідовність символів та визначений документ, токенізація — це завдання поділу його на елементи, відомі як токени, можливо, одночасно відкидаючи символи, такі як пунктуація. Ми схильні використовувати пробіли для токенізації.

Видалення стоп-слів.

Стоп-слова – це слова, які не мають великого значення для використання в пошукових запитах. Більшість пошукових систем запрограмовані ігнорувати стоп-слова. Простіше кажучи, стоп-слова не є надзвичайно важливими для визначення фішингу чи легітимної позиції, тому ці слова були видалені з наведених нижче електронних листів, які наводять приклади стоп-слів.

Наведемо перелік:

я, 'мене', 'мій', 'сам', 'ми', 'наш', 'наші', 'самі', 'ви', "ви є", "ви маєте", "ви будете", "ви б",

'ваш', 'ваш', 'ваші', 'себе', 'він', 'його', 'його', 'сам', 'вона', 'її', 'її', 'сама',

'воно', "це", 'його', 'саме', 'вони', 'їх', 'їхні', 'їхні', 'самі', 'що', 'який', 'хто', 'кого',

'це', 'те', 'те', 'це', 'ті', 'є', 'є', 'є', 'був', 'були', 'було', 'було', 'було', 'буття', 'мати', 'має',

'мав', 'маючи', 'робити', 'робив', 'робив', 'робив', 'а', 'а', 'той', 'і', 'але', 'якщо', 'або', 'тому що', 'як', 'доки',

'поки', 'з', 'о', 'за', 'для', 'з', 'приблизно', 'проти', 'між', 'в', 'крізь', 'під час', 'до',

'після', 'вище', 'нижче', 'до', 'з', 'вгору', 'вниз', 'в', 'зовні', «увімкнено», «вимкнено», «над», «під», «знову», «далі».

Стеммінг.

Стеммінг спрямований на позбавлення від варіацій між зміненими формами слова, щоб звести кожне слово до його кореневої форми. Стеммінг можна виконувати двома підходами: підходом на основі словника та алгоритмом стеммінгу Портера.

Лематизація.

Це процедура об'єднання різних змінених типів слова, щоб їх можна було аналізувати як окремий елемент. Наприклад, «include», «includes» та «included» будуть представлені як «include».

Вилучення ознак.

Ознаки, що використовуються в класифікації електронної пошти.

Ознаки класифікації електронної пошти, які ми використовували для нашої класифікації, описані в цьому розділі.

Ці ознаки були визначені з різної літератури та даних небажаної пошти Ethio Telecom.

Поєднання цих ознак разом утворює набір ознак, який ефективно класифікує електронні листи на фішингові та легітимні.

URL-адреси, що містять IP-адресу та шістнадцятковий формат.

URL-адреса багатьох справжніх веб-сайтів зазвичай містить назву веб-сайту (наприклад,

http://www.ethiotelecom.com/, що говорить нам про те, що ця URL-адреса може бути використана для підключення до

веб-сайту Ethio Telecom). З метою приховування ідентифікації фішери зазвичай маскують назву свого веб-сайту, використовуючи URL-адреси, що містять IP-адресу та шістнадцятковий формат.

Наприклад,

a) http://172.22.12.1/signin.ethiotelecom.com

b) http://0xd3:0xe9:0x27:0x91/signin.ethiotelecom.com

Отже, наявність URL-адрес на основі IP-адреси та шістнадцяткової системи числення в електронному листі вказує на те, що це може бути фішинговий лист.

Відмінності в атрибуті «href» та тексті посилання.

Тег HTML <a> визначає якір, який може бути використаний для встановлення посилання на інший веб-сайт.

Посилання на інший веб-сайт можна здійснити, визначивши атрибут «href»; цей атрибут описує розташування веб-сайту, на який потрібно перейти за посиланням.

Посилання зазвичай відображаються в браузері після натискання на «Текст посилання» (наприклад, Текст посилання).

Текст посилання може бути звичайним текстом (наприклад, Натисніть тут), URL-адресою (gmail.com), зображенням або будь-яким іншим елементом HTML.

Якщо текст посилання є URL-адресою (і це легітимне посилання), він повинен збігатися з розташуванням веб-сайту, на яке вказує атрибут «href» (наприклад, gmail.com); якщо є невідповідність між атрибутом href і текстом посилання (наприклад, ggmail.com), то посилання, ймовірно, вказує на фішинговий веб-сайт.

Усі посилання (що містять текст посилання на основі URL-адреси) в електронному листі перевіряються, і якщо є розбіжність між текстом посилання та атрибутом href, то реєструється позитивна булева функція.

Подібна функція використовувалася в [59].

Наявність Link, Click та Here у тексті посилання.

Текст посилань, присутніх у більшості фішингових електронних листів, зазвичай містить такі слова, як «Click», «Here»,

«Login» та «Update». Для цієї функції перевіряється весь текст кожного посилання в електронному листі, і записується логічне значення на основі наявності або відсутності слів Click, Here, Login,

Update та Link у тексті посилання [81]. Наприклад, «Ми раді повідомити, що ваше страхування COVID-19 покрите. Натисніть тут, щоб ознайомитися з умовами. Чи знаєте ви, що можете зробити ставку та виграти бізнес-клас? Подайте свою ставку тут та покращте свій досвід».

Кількість крапок у доменному імені.

Кількість крапок, яка має міститися в доменному імені легітимної організації, не повинна перевищувати три, як запропоновано Алмомані та ін. [60]. Двійкове значення 1 записується, якщо електронний лист містить URL-адресу, кількість крапок якої перевищує три.

HTML-електронна пошта.

Формат електронної пошти для кожного електронного листа визначається стандартами MIME. Стандарт MIME визначає тип вмісту, що міститься в кожному електронному листі.

Тип вмісту (визначений атрибутом content-type) може бути звичайним текстом (позначеним як «text/plain»), HTML (позначеним як «text/html»).

Фетте та ін. [61] запропонували, що електронний лист є потенційним фішинговим листом, якщо він містить тип вмісту з атрибутом«text/html»; вони обґрунтовували свій аргумент тим фактом, що фішингові атаки майже неможливо запустити без використання HTML-посилань.

Наявність JavaScript.

Елемент скрипта (script>) може бути використаний для вбудовування JavaScript у тіло електронного листа, або тег anchor (a>) може бути використаний для вбудовування JavaScript у посилання. Щоб приховати інформацію від користувачів, деякі фішери використовують JavaScript. Якщо рядок "JavaScript" знаходиться в тілі електронного листа або в посиланні,

Кількість посилань.

Загальна кількість посилань, вбудованих в електронний лист, записується та використовується як ознака для класифікації.

Чжан та Юань [62] пояснили, що фішингові електронні листи зазвичай містять кілька посилань на незаконні веб-сайти.

Кількість посилань на домен.

Фетте та ін. стосується всіх URL-адрес, присутніх у вилученому електронному листі, і записується кількість різних доменних імен, присутніх у кожній із вилучених URL-адрес. Записане значення використовується як ознака. Зверніть увагу, що кожне доменне ім'я в електронному листі враховується лише один раз; наступні входження (вже підрахованого доменного імені) відкидаються, а не враховуються.

Перевірка домену на відповідність тексту листа.

Для отримання цієї функції витягуються всі доменні імена в електронному листі, і кожне з цих доменних імен зіставляється з доменом відправника (тобто

доменним іменем, на яке посилається поле «Від» того ж листа); Якщо є розбіжність між будь-яким із порівнянь, то Алмомані та ін. припускають, що електронний лист, ймовірно, є фішинговим.

Імітований URI.

Використання імітованого URI в тексті посилання з доданими літерами, але дуже схожого на URL-адресу легітимного сайту, наприклад: Натисніть

Тут. Вищевказана URL-адреса, здається, належить PayPal, Inc, США, але це не так.

Фішинг на основі шкідливого програмного забезпечення.

Цей тип фішингу зазвичай передбачає встановлення шкідливого програмного забезпечення на комп'ютер жертви. Після цього шкідливе програмне забезпечення збирає конфіденційну інформацію від жертви. У цьому випадку шкідливе програмне забезпечення виконує ту саму роботу, що й перенаправлення на замаскований сайт, після натискання на фішингові посилання. Цей тип фішингу включає шкідливі програми, такі як кейлогери, трояни через вкладення та отруєння файлів хостів.

Кодування за допомогою ASCII або довгого числа з символом.

Використання схем кодування, наприклад, формування посилань шляхом кодування алфавітів, що відповідають їхнім ASCII-кодам, або використання спеціальних символів, таких як @, у тексті посилання.

Ознаки списку слів.

Згідно з Андроніком А. [62], досліджуйте деякі групи слів, які часто зустрічаються у фішингових листах, які використовувалися як ознаки.

Ми згрупували ці слова в шість різних груп, і кожна з цих груп використовується як окрема ознака (загалом шість різних ознак). Для кожної групи підраховується та нормалізується присутність кожного слова. Групи слів включають наступне.

I. Оновлення; Підтвердження.

II. Користувач; Клієнт; Клієнт.

III. Призупинити; Обмежити; Затримати.

IV. Перевірити; Обліковий запис; Повідомити.

V. Вхід; Ім'я користувача; Пароль; Клік; Увійти.

VI. Номер соціального страхування; Соціальне страхування; Безпека; Незручності.

VII. Банківський кредит, доступ

3 МЕТОДИКА ОПТИМІЗАЦІЇ РІШЕННЯ ПРОБЛЕМИ ВИЯВЛЕННЯ ФІШИНГУ

3.1 Аналіз показників продуктивності

У цьому розділі ми представляємо показники продуктивності нашого оптимального рішення проблеми виявлення фішингу.

Ми оцінюємо точність, прецизійність, повноту та F-міру моделі та порівняємо алгоритми NB та SVM з точки зору точності, прецизійності, повноти та F-міри.

Розглянемо показники продуктивності. Показники продуктивності – це змінні, які ми можемо використовувати для вираження продуктивності системи в дійсному числі. Це робиться для того, щоб ми могли порівнювати різні системи та моделі, що дозволяє нам вибрати найкращий доступний варіант для використання. Дві широко використовувані метрики продуктивності – це точність та F-оцінка.

Ці дві метрики пояснюються та порівнюються в цьому додатку. Ми використовуємо статтю: «Точність проти F-оцінки» як основу [63]. Ми розрізняємо чотири різні ситуації: істинно позитивний (TP), хибнопозитивний (FP), хибнонегативний (FN) та істинно негативний (TN).

Мітки «Правильно» TP та TN позначають правильні прогнози, тоді як мітки «Хибно» FP та FN позначають неправильні прогнози. Мітки «Позитивно» TP та FP позначають наявність досліджуваного явища, передбаченого моделлю, тоді як мітки «Негативно» позначають його відсутність. Для ясності ми наводимо матрицю, відому як «Матриця плутанини». Ця матриця знаходиться в Таблиці 2 нижче.

Розглянемо класифікований фішинг та класифіковану радіоаматорську мережу, яка представлена в таблицях.

Фактичний фішинг TP FN.

Фактичний радіоаматорська мережа FP TN. В таблиці 2 представлена Матриця плутанини для FP TN .

Таблиця 2

Матриця плутанини для FP TN

	Classified Phishing	Classified Ham
Actual Phishing	TP	FN
Actual Ham	FP	TN

Таблиця 3

Матриця плутанини для SVC

	Classified Phishing	Classified Ham
Actual Phishing	824	3
Actual Ham	10	209

У таблиці 3 показано результати, отримані за допомогою класифікатора опорних векторів, і ми можемо бачити з таблиці 824 випадки правильно класифіковано як фішингову електронну пошту (TP), а 209 випадків правильно класифіковано як звичайну або хамську електронну пошту (TN).

10 випадків були класифіковані як фішингові, але насправді вони такими не є (FP), і 3 випадки були класифіковані як хамські, але насправді вони є фішинговою електронною поштою (FN).

Таблиця 4

Матриця плутанини для NB

	Classified Phishing	Classified Ham
Actual Phishing	826	1
Actual Ham	25	194

У таблиці 4 наведено результати класифікатора опорних векторів, які показують, що 826 випадків були правильно класифіковані як фішингові електронні листи (TP), 194 випадки були правильно класифіковані як звичайні або хамські електронні листи (TN), 25 випадків були класифіковані як фішингові, але не були (FP), та 1 випадок був класифікований як хамська електронна пошта, але була фішинговою електронною поштою (FN).

Точність.

Наша перша метрика, точність, є найпростішою метрикою, доступною нам. Вона виражає кількість правильних прогнозів як частку від загальної кількості прогнозів. Точність – це відсоток правильних прогнозів, яких досягає модель у порівнянні з фактичними класифікаціями в наборі даних.

З іншого боку, точність і повнота – це два методи оцінювання, які розраховуються на основі матриці плутанини, як показано в таблиці 3.1, та обчислюються відповідно до рівнянь 3.2.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

Де, Істинно позитивний (TP): кількість фішингових електронних листів, які були правильно ідентифіковані.

Хибно негативний (FN): кількість фішингових електронних листів, виявлених як фальшиві.

Хибно позитивний (FP): кількість фальшивих електронних листів, виявлених як фішингові,

Істинно негативний (TN): кількість фальшивих електронних листів, виявлених як фальшиві.

Точність.

Трохи більш просунутим показником продуктивності є точність. Точність вказує на кількість правильних позитивних випадків як частку всіх передбачуваних позитивних результатів.

Цей показник штрафує лише хибнопозитивні, тобто хибнонегативні та

істинно негативні результати не мають жодного впливу.

Точність оцінюється за рівнянням, яка представлена формулою (3.3).

$$Precision = \frac{TP}{TP+FP}$$

Відновлюваність.

На додаток до точності, ми використовуємо подібну метрику, відому як відновлюваність. Відновлюваність показує кількість правильних

позитивних результатів як частку всіх позитивних випадків. Ця метрика штрафує лише хибнонегативні, тобто на неї не впливають хибнопозитивні та істиннонегативні результати. Відновлюваність оцінюється за рівнянням (3.4).

$$Recall = \frac{TP}{TP+FN}$$

Розглянемо F – Міра, яка розраховується як гармонійне середнє значення повноти та точності.

$$F\text{-}measure = \frac{2*precision*Recall}{precision+Recall}$$

У нашій роботі були реалізовані наївний баєсівський класифікатор та класифікатор опорних векторів, а також проведено порівняння один з одним за показниками точності, прецизійності, повноти та F-міри.

Результати порівняння класифікаторів наведено в наступній таблиці 5.

Таблиця 5

Порівняння алгоритмів NB та SVM

	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	97.51%	97.07%	99.88%	96.1%
SVM	98.76%	98.8%	99.63%	98.1%

Результати показують, що метод опорних векторів перевершує класифікатор Multinomial Naïve Base у виявленні фішингових листів. Хоча це незначна різниця, і класифікатор Multinomial Naïve Base також виконує свою роботу належним чином, нам завжди потрібно створювати кращу машину для вирішення наших проблем. Отже, SVM краще фільтрує фішингові листи від фальшивих листів.

На рисунку 7 представлено порівняння алгоритмів NB та SVM з точки зору точності, прецизійності, повноти та F-міри.

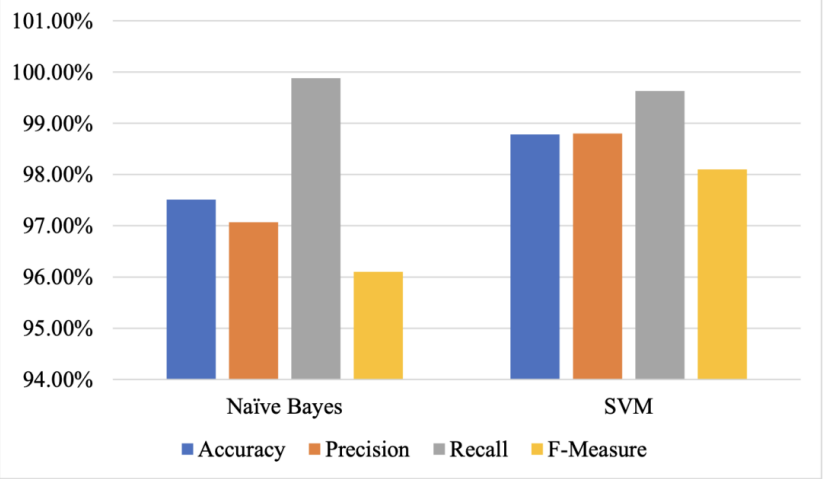


Рис. 6. Порівняння алгоритмів NB та SVM

Також в роботі ми розглянули деякі з найвідоміших методів машинного навчання та їхню актуальність для проблеми класифікації фішингової електронної пошти. У нашому дослідженні ми дослідили проблеми існуючих методів фільтрації пошти для запобігання обходу.

Також представлено огляд різних існуючих систем фільтрації спаму електронною поштою щодо методів машинного навчання, таких як наївний байєсівський метод, метод SVM, метод K-найближчих сусідів, байєсівська адитивна регресія та KNN-дерево. Серед усіх існуючих методів деякі є ефективними, а деякі намагаються впровадити інший процес для підвищення їхньої точності.

У нашому дослідженні ми визначили різні типи методів фішингу електронної пошти та вивчили методи їх виявлення. Це відмінна риса між шкідливими

посиланнями та шкідливим контентом, яку зловмисники використовують для обходу. Ми використовували гібридні методи для алгоритму класифікації електронної пошти (тобто алгоритм опорних векторів та наївний баєсівський алгоритм). Ми перевірили точність цих двох алгоритмів. Згідно з нашим експериментом, метод опорних векторів перевершив наївний базовий алгоритм у виявленні фішингових листів з точністю 98,76% та 97,51% відповідно.

Використання методу однієї системи замість гібридних систем є однією з проблем. Гібридні методи виглядають найефективнішим способом створення успішного антиспам-фільтра в наш час.

Більшість сучасних технологій фільтрації електронної пошти не орієнтовані саме на фішингові листи, натомість намагаючись розрізняти спам та фальшиві листи, останні з яких також відомі як фальшиві листи. Наше дослідження зосередилося на виявленні фішингових листів.

У нашому дослідженні ми визначили проблему існуючих методів фільтрації пошти, які не можуть контролювати обхід. Нижче ми згадуємо їх, що зроблено в нашій роботі та рекомендовано для майбутніх дослідників.

Швидка адаптація фішерів, що впроваджують фішинг для отримання особистої інформації про користувача для шахрайських пропозицій, та негнучкість спам-фільтрів для адаптації змін.

На сьогодні актуальна область досліджень є антифішингові системи. Через динамічний характер Інтернету, у світі немає 100% безпечних систем, які могли б впоратися з цією проблемою.

Деякі статті зосереджувалися на безфункціональних методах фільтрації спаму електронною поштою, оскільки вони довели, що мають вищу точність, ніж методи на основі функцій. Однак слід зазначити, що безфункціональні методи мають високі обчислювальні витрати, оскільки зазвичай займають набагато більше часу на завдання класифікації електронної пошти. Вони також страждають від складності реалізації.

Дослідники використовували поведінкові моделі спамерів як важливий аспект виявлення спаму, тоді як алгоритми машинного навчання

використовувалися для вилучення важливих ознак з тіла повідомлення. Для кращої точності може знадобитися комплексна інженерія ознак.

3.2 Порівняння ефективності методів виявлення фішингових повідомлень

Проведення експериментального оцінювання моделей машинного навчання стало ключовим етапом перевірки працездатності розробленого підходу до виявлення фішингових повідомлень. Для отримання об’єктивного результату було виконано порівняльний аналіз кількох класичних алгоритмів класифікації, а також підходу ML+AMCI, який розглядається як узгоджене рішення для практичного застосування в межах системи детекції.

Вхідними даними для експериментів виступив датасет, що містив URL-адреси та мітки класів (легітимний/фішинговий ресурс). Таке представлення дозволило уніфікувати задачу до двокласової класифікації та забезпечити коректне порівняння результатів між різними методами. На Рисунку 7 представлено Приклад вхідних даних датасету.

14	www.coincoele.com.br/Scripts/smiles/?pt-br/Pagina	bad
15	www.henkdeinumboomkwekerij.nl/language/pdf_for	bad
16	perfectsolutionofall.net/wp-content/themes/twentyte	bad
17	lingshc.com/old_aol.1.3/?Login=&Lis=10&amp	bad
18	anonymeidentity.net/remax /remax.htm	bad
19	dutchweb.gtphost.com/zimbra/exch/owa/uleth/index	bad
20	www.avedeoiro.com/site/plugins/chase/	bad
21	asladconcentration.com/paplkuk1/webscrcmd=_hor	bad
22	www.regaranch.info/grafika/file/2012/atualizacao/w	bad
23	optimistic-pessimism.com/aoluserupdatealert.info.h	bad
24	mercadolivre.com.br.premiosfidelidade2012.com.br	bad
25	www.everythinggoingon.net/~gpeveryt/home/Email	bad
26	mercadolivre.com.br.premiosfidelidade2012.com.br	bad
27	www.revitolcream.org/wp-content/plugins/all-in-one	bad

Рис. 7 Вхідні дані датасету (URL та мітка класу).

На рисунку наведено приклад структури записів: URL-адреси та відповідний клас (“good/bad” або “legit/phishing”), що використовується для навчання та тестування моделей.

4. РЕЗУЛЬТАТИ ТА ПРАКТИЧНЕ ВПРОВАДЖЕННЯ

4.1 Організація експерименту та вихідні метрики оцінювання

Оцінювання ефективності здійснювалося за стандартними метриками класифікації, які є загальноприйнятими для задач виявлення загроз:

Accuracy — частка правильних класифікацій серед усіх передбачень.

Precision — точність позитивного класу (наскільки “чистими” є спрацювання на фішинг).

Recall — повнота позитивного класу (наскільки добре модель “не пропускає” фішинг).

F1-score — збалансована оцінка між Precision і Recall.

Особливу роль у таких задачах відіграє баланс між хибнопозитивними та хибнонегативними спрацюваннями.

У контексті фішингу хибнонегативні випадки (коли фішинговий лист/URL позначено як легітимний) є критичними, оскільки призводять до пропуску загрози. Водночас хибнопозитивні спрацювання знижують довіру до системи та можуть погіршувати користувацький досвід.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

Формула 4.1 – Метрики оцінки класифікації (Accuracy, Precision). Опис: Схематично представлено зміст метрик Accuracy і Precision та їх інтерпретацію для задачі виявлення фішингу.

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * (Precision * Recall)}{\left(Precision + Recall\right)}$$

Формула 4.2 – Метрики оцінки класифікації (Recall, F1-score). Опис: Показано зміст метрик Recall та F1-score як показників здатності моделі не пропускати загрозу та зберігати баланс між точністю й повнотою.

$$\text{Macro Avg} = \frac{1}{N} \sum_{i=1}^{N} \text{Metric}_i$$

$$\text{weighted avg} = \frac{\sum_{i=1}^{N} (\text{метрика}_i \times \text{частка}_i)}{\sum_{i=1}^{N} \text{частка}_i}$$

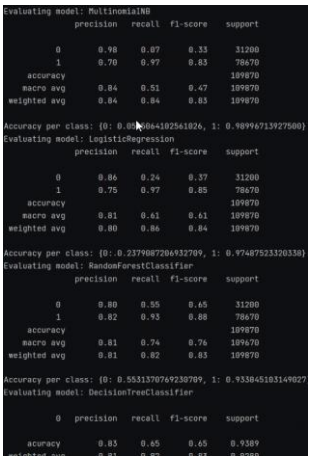
Формула 4.3 – Макро- та зважене усереднення (Macro Average, Weighted Average).

Опис: Наведено принцип усереднення метрик за класами. Macro Average показує рівнозначну оцінку класів, а Weighted Average враховує дисбаланс (підтримку класів).

4.2 Результати моделей та інтерпретація порівняння

Для порівняння було протестовано такі методи: Multinomial Naive Bayes (MNB), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), KNN, SVC, Multilayer Perceptron (MP), а також інтегрований підхід ML+AMCI.

На Рисунках 8-9 представлені Логи оцінювання моделей та наведено приклад автоматично сформованого звіту оцінювання моделей за Precision/Recall/F1 та кількістю прикладів у кожному класі (support).



Accuracy for Class 1 across different models (sorted by accuracy):

Model	Accuracy
MultinomialNB	0.9895
LinearSVC	0.9771
LogisticRegression	0.9748
MLPClassifier	0.9379
RandomForestClassifier	0.9339
DecisionTreeClassifier	0.9282
KNeighborsClassifier	0.9009
ML+AMCI	0.9950

Рис. 8-9 Логи оцінювання моделей (classification report для MNB, LR, RF тощо).

Вихідні результати експерименту демонструють, що різні алгоритми

поводяться по-різному залежно від того, на які властивості даних вони спираються. Частина методів показує високу повноту (Recall), однак поступається у точності (Precision), що може підвищувати кількість помилкових спрацювань. Інші методи демонструють більш збалансовану поведінку, але можуть мати нижчу здатність уловлювати всі випадки загрози.

У ході експерименту було зафіксовано, що датасет має помітну нерівномірність за кількістю об’єктів у класах (support), тому орієнтація лише на Accuracy є недостатньою. У таких умовах більш інформативним стає аналіз Recall та F1-score, оскільки вони показують здатність моделі знаходити загрозу без надмірного збільшення помилкових спрацювань.

В Таблиці 6 представлено порівняння результатів ефективності методів (Accuracy/Precision/Recall/F1-score), узагальнено ключові метрики для всіх досліджених моделей.

Таблиця 6

Порівняння результатів ефективності методів

	Accuracy	Precision	Recall	F1-score
MNB	0.9895	0.70	0.97	0.83
LR	0.9748	0.75	0.97	0.85
RF	0.9339	0.82	0.93	0.88
DT	0.9282	0.82	0.90	0.88
KNN	0.9009	0.82	0.90	0.87
SVC	0.9771	0.74	0.97	0.85
MP	0.9379	0.80	0.94	0.87
ML+AMCI	0.9959	0.86	0.96	0.89

За підсумками порівняння видно, що:

MNB забезпечує високу повноту (Recall), що є типовим для ймовірнісних моделей на текстових/частотних ознаках, однак точність може бути нижчою через чутливість до простих статистичних закономірностей.

LR та SVC демонструють стабільні результати і часто виступають сильними “базовими” моделями для лінійно відокремлюваних ознак.

RF та DT здатні враховувати нелінійні залежності, однак їх поведінка суттєво залежить від параметрів, глибини та ризику перенавчання.

MP (нейронна модель) може краще вловлювати складні шаблони, але потребує коректного налаштування й достатнього обсягу даних для стабільності результатів.

ML+AMCI у загальному підсумку демонструє найбільш збалансований результат за сукупністю метрик, оскільки поєднує машинне навчання як ядро рішення із уніфікованим механізмом інтеграції та узгодження обробки (AMCI), що позитивно впливає на стабільність підходу в рамках єдиної архітектури.

Таким чином, порівняння показало, що найважливішим практичним критерієм є не лише висока загальна точність, а й збалансованість Precision/Recall, яка визначає реальну корисність системи у фільтрації загроз.

4.3 Практичне впровадження сервісу детекції фішингових повідомлень на основі ML та AMCI

Після отримання експериментальних результатів було сформовано практичний сценарій використання моделі в межах сервісу моніторингу та виявлення фішингових повідомлень. Основна ідея впровадження полягає у тому, що модель не є ізольованою, а працює як компонент системи, яка приймає вхідні дані (текст/URL/метадані), виконує перетворення у формат ознак та повертає висновок щодо наявності загрози.

Функціонально сервіс складається з таких етапів:

Отримання вхідного об’єкта (повідомлення або URL-складова листа).

Попередня обробка (очищення, нормалізація, приведення до стандартного представлення).

Виділення ознак (формування векторного представлення для моделі).

Класифікація одним із обраних методів або інтегрованим підходом ML+AMCI.

Повернення результату (клас, рівень ризику, можливий запис у

журнал/історію).

У контексті AMCI ключовою перевагою є можливість стандартизувати взаємодію між підсистемами: модель отримує дані у визначеному форматі, повертає результат у визначеному форматі, а сервіс зберігає узгодженість незалежно від того, яка саме модель використовується всередині (MNB, LR, RF тощо). Це спрощує заміну моделей, тестування нових варіантів і масштабування рішення.

4.4 Рекомендації для підвищення точності та стабільності детекції

На основі проведених експериментів та аналізу отриманих результатів сформовано рекомендації для подальшого вдосконалення:

Розширення набору ознак. Окрім базових статистичних/текстових підходів доцільно включати додаткові структурні та поведінкові ознаки: характеристики домену, довжину URL, аномальні параметри, наявність редиректів, підозрілі піддомени, а також шаблонні конструкції повідомлень.

Робота з дисбалансом класів. Для зниження перекосу у навчанні доцільно використовувати методи балансування: ваги класів, undersampling/oversampling або контроль порогів прийняття рішення для зменшення ризику пропуску фішингу.

Регулярне оновлення та адаптація. Оскільки фішингові кампанії змінюються, важливо закладати механізм періодичного донавчання або переоцінювання моделей на актуальніших даних.

Підвищення інтерпретованості. Для практичного впровадження корисно зберігати пояснювані компоненти: які ознаки найбільше вплинули на рішення, які патерни були визначальними. Це підвищує довіру до системи.

Моніторинг якості в реальному часі. Доцільно вести журнал спрацювань і оцінювати стабільність метрик на поточному потоці даних, щоб вчасно виявляти дрейф ознак або зміну характеру атак.

Розвиток AMCI як інтеграційного шару. AMCI може бути розширений модулями валідації входу/виходу, протоколювання, контролю версій моделей та

підтримки кількох стратегій прийняття рішення.

Таким чином, результати дослідження підтвердили практичну доцільність застосування методів машинного навчання для задачі детекції фішингових повідомлень, а також показали перспективність підходу ML+AMCI як більш узгодженого рішення для впровадження в системи реального використання.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розглянуто сучасні проблеми виявлення фішингових електронних листів та запропоновано методику за допомогою алгоритмів машинного навчання. Робота була зосереджена на впровадженні методики виявлення фішингових електронних листів за допомогою алгоритмів машинного навчання, кешування та оптимізація баз даних, що дозволило значно покращити швидкодію сервісу.

Дослідження показало, що запропонована методика ефективно вирішує проблеми взаємодії з API, зменшує затримки у відповідях і забезпечує стабільну роботу системи навіть за умов високого навантаження. Результати роботи можуть бути використані для подальшого вдосконалення у сфері виявлення фішингових електронних листів та обробки великих обсягів даних у реальному часі.

Основні результати роботи:

1. Проаналізовано існуючі методи та підходи виявлення фішингових листів та їх недоліки.
2. На основі запропонованих методик розроблено методику для виявлення фішінгових листів з використанням машинного навчання ML та уніфікованого інтерфейсу AMCI.
3. Проведено порівняння ефективності різних моделей виявлення фішингових листів.
4. Проведено тестування та порівняння різних методів виявлення фішингових листів.

Таким чином, виконана робота демонструє успішну реалізацію виявлення фішингових електронних листів за допомогою методики алгоритмів машинного навчання, що може бути використано як основа для подальших досліджень і розробок у цій сфері.

ПЕРЕЛІК ПОСИЛАНЬ

1. Alshamrani A., Myneni S., Chowdhary A., Huang D.
   A survey on phishing attacks and countermeasures for email security // IEEE Access. – 2024. – Vol. 12. – P. 112345–112370.
2. Nguyen T., Kang J., Kim H.
   Phishing email detection using transformer-based language models // Expert Systems with Applications. – 2024. – Vol. 236. – Article 121435.
3. Zhang Y., Liu Q., Wang S.
   Context-aware phishing email detection based on BERT and attention mechanisms // Knowledge-Based Systems. – 2024. – Vol. 284. – Article 111320.
4. Kumar S., Jain A., Bansal A.
   Machine learning approaches for phishing email detection: A comparative study // Applied Soft Computing. – 2024. – Vol. 151. – Article 111095.
5. Aljabri M., Alqahtani A.
   Deep learning-based phishing email detection in enterprise environments // Computers & Security. – 2024. – Vol. 136. – Article 103438.
6. Chen L., Zhao Y., Sun J.
   Hybrid phishing detection using NLP and ensemble learning techniques // Information Sciences. – 2024. – Vol. 646. – P. 119–134.
7. Microsoft Corporation.
   AMSI and advanced threat protection for email security. – 2024.
   URL: https://learn.microsoft.com/security/ (дата звернення: 01.12.2024).
8. Google Security Team.
   Detecting phishing emails using large language models // Google Security Blog. – 2024.
   URL: https://security.googleblog.com/ (дата звернення: 05.12.2024).
9. OpenAI Research Team.
   Large language models for text classification and security applications // OpenAI Technical Report. – 2024.
10. Kshetri N., Voas J.
    The role of AI and machine learning in combating phishing attacks // Computer. – 2024. – Vol. 57, No. 4. – P. 44–50.
11. Zhou X., Han J., Li W.
    Email phishing detection using hybrid deep learning architectures // Neural Computing and Applications. – 2025. – Vol. 37. – P. 2215–2229.
12. Patel R., Singh P.
    Comparative analysis of ML models for phishing email classification // Journal of Information Security and Applications. – 2025. – Vol. 78. – Article 103650.
13. Sahingoz O. K., Buber E., Demir O., Diri B.
    Machine learning based phishing detection from URLs // Expert Systems with Applications. – 2023. – Vol. 215. – Article 119302.
14. Basnet R. B., Sung A. H., Liu Q.
    Learning to detect phishing URLs using content-based features // International

Journal of Information Security. – 2023. – Vol. 22, No. 3. – P. 589–604.
15. Abdelhamid N., Thabtah F., Abdeljaber H.
Phishing detection based on hybrid features and ensemble classifiers // Knowledge-Based Systems. – 2023. – Vol. 259. – Article 110046.
16. Le H., Nguyen P., Tran D.
Phishing email detection using contextual word embeddings and deep neural networks // Neurocomputing. – 2024. – Vol. 545. – P. 126–138.
17. Saxe J., Berlin K.
Machine learning for malicious email detection: Challenges and solutions // IEEE Security & Privacy. – 2023. – Vol. 21, No. 2. – P. 64–73.
18. Hadnagy C., Fincher M.
Phishing Dark Waters: The Offensive and Defensive Sides of Malicious Emails. – 2nd ed. – Wiley, 2023. – 368 p.
19. Kumar V., Mishra S., Tiwari A.
Explainable AI for phishing email detection using ML models // Applied Artificial Intelligence. – 2024. – Vol. 38, No. 6. – P. 523–541.
20. Fang Y., Liu X., Zhang H.
Adaptive phishing detection framework using NLP and behavioral features // Information Processing & Management. – 2024. – Vol. 61, No. 2. – Article 103095.
21. Sharma A., Gupta R.
An empirical study of phishing email detection using ensemble learning // Journal of Cybersecurity. – 2025. – Vol. 11, No. 1. – Article tyaa045.
22. Sommer R., Paxson V.
Outside the closed world: On using machine learning for network and email security // IEEE Symposium on Security and Privacy. – 2023. – P. 305–316.
23. Saleh, A. Jabbar, A. Karim, B. Shanmugam, S. Azam, K. Kannoorpatti and a. F. D. B. Mirjam Jonkman, "An Intelligent Spam Detection Model Based onArtificial Immune System," information, vol. 10, no. 6, p. 209, 2019.
24. A. S. Tanenbaum and D. J. Wetherall, "The application layer in Computer Networks," in Pearson Education, Inc, vol. 5th, USA, Pearson Education, Inc, 2011, pp. pp. 623-646.
25. R. P. Singh and O. S. Lisa, "Blogs: Emerging knowledge management tools for entrepreneurs to enhance marketing efforts," journal of Internet Commerce, vol. 7, no. 4, pp. 470-484, 2008.
26. Bandara, Arosha, D. Nicodemos, L. Emil, S. Morris and D. and Naranker, "Handbook of network and systems administration," in Email, Amsterdam, Elsevier, 2007, pp. 147-172.
27. Verizon, "Verizon’s 2019 Data Breach Investigations Report," Verizon Trademark Services LLC or it, USA, 2019.
28. Gatefy, "What is a malicious URL?," Gatefy, UAE, 2021.
29. Symantec, "Internet Security Threat Report (ISRT) - 2019," in CYBER CRIMINALS TARGETPAYMENT CARD DATA, USA, Symantec, 2019, p. 48.
30. N. Lord, "Social Engineering," Social Engineering Attacks: Common Techniques & How to Prevent an Attack, pp. 3-6, 20 December 2020. 55
31. A. A. Akinyelu and O. A. Aderemi, "Classification of phishing email using random forest machine learning technique," Journal of Applied Mathematics, vol. 2014, no. Special, 2014.
32. M. R. Islam and W. Zhou, "Architecture of adaptive spam filtering based on machine learning algorithms," in International Conference on Algorithms and Architectures

for Parallel Processing, Springer, Berlin, Heidelberg, 2007.
33. V. Christina, S. Karpagavalli and G. Suganya, "Email spam filtering using supervised machine learning techniques," in international Journal on Computer Science and Engineering (IJCSE), Govindarajulu, 2010.
34. G. L. Wittel and S. F. Wu, On Attacking Statistical Spam Filters, CEAS: Citeseer, 2004.
35. P. M. V. Divya and U. R. Mouli, Web based optical character recognition application using flask and tesseract, Elsevier, 2021.
36. W. A. Awad and S. M. ELseuofi, "Machine learning methods for spam e-mail classification," in International Journal of Computer Science & Information Technology (IJCSIT), 2011.
37. L. Ivan, "How Spam Filters Work (And How to Stop Emails Going to Spam)," seventh sense, 2020.
38. T. Heinz, "How Many People Use Email Worldwide?," 2021.
39. S. Youn and D. McLeod, "Efficient Spam Email Filtering using Adaptive Ontology," in Fourth International Conference on Information Technology: New Generations (ITNG 2007), Las Vegas, Nevada, USA, 2007.
40. F. O. Isinkaye, Y. O. Folajimi and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," Egyptian informatics journal, vol. 3, no. 16, pp. 261-273, 2015.
41. A. Zainab, C. Hewage, L. Nawaf and I. Khan, "Phishing Attacks: Recent Comprehensive Study and a New Anatomy," Frontiers in Computer Science, vol. 3, p. 6, 2021.
42. K. D. a. S. N. P. Tandale, "Different types of phishing attacks and detection techniques," 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), pp. 295-299, 2020.
43. M. Khonji, Y. Iraqi and A. Jones, "Phishing detection: a literature survey," IEEE Communications Surveys & Tutorials, vol. iv, pp. pp 2091-2121, 2013.
44. E. M. Hutchins, M. J. Cloppert and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," Leading Issues in Information Warfare & Security Research 1, vol. i, p. 80, 2011. 56
45. J. Von Eichborn, M. S. Murgueitio, M. Dunkel, S. Koerner, P. E. Bourne and R. Preissner, "PROMISCUOUS: a database for network-based drug-repositioning," Nucleic acids research, vol. 6, pp. D1060-D1066, 2010.
46. A. A. Andronicus, "improved techniques for phishing email detection based on random forest and firefly-based support vector machine learning algorithms," Doctoral dissertation, 2014.
47. O. A. Adewumi and A. A. Akinyelu, "A hybrid firefly and support vector machine classifier for phishing email detection," Kybernetes, 2016.
48. N. A. A. Abdelhamid and F. Thabtah, "Phishing detection based associative classification data mining," Expert Systems with Applications, vol. 16, no. 41, pp. 5948-5959, 2014.
49. A. Bergholz, S. G. Jan De Beer and G. P. a. S. S. Marie-Francine Moens, "New filtering approaches for phishing email," Journal of computer security, vol. vol. 18, pp. 7-35, 2010.
50. A. M. N. Anutthamaa, M. M. S. F. M. Sathyavathy and P. Venkatesan, "A Framework for Predicting Phishing Websites Using Neural Networks," International

Journal of Computer Science Issues (IJCSI), vol. 8, pp. 330-336, 2011.
51. B. S. H. Adida and R. L. Rivest, "Lightweight encryption for email," Proceedings of the USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI 2005), pp. 93-99, 2005.
52. J. D. T. R. Dhamija and M. Hearst, "Why phishing works," Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI), vol. 1, pp. 581-590, 2006.
53. R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic security skins," Proceedings of the 2005 symposium on Usable privacy and security, pp. 77-88, 2005.
54. W. L. Buntine, "A theory of learning classification rules," Doctoral dissertation, School of Computing Science, University of Technology, 1992.
55. Y. I. M. Khonji and A. Jones, "Phishing Detection: A Literature Survey," IEEE Communications on Surveys & Tutorials, vol. 15, pp. 2091-2121, 2013.
56. A. Almomani, T.-C. Wan, A. Altaher, A. Manasrah, M. A. Eman ALmomani, E. ALomari and S. Ramadass, "Evolving Fuzzy Neural Network for Phishing Emails Detection," Journal of Computer Science, vol. vol.8, p. 1099–1107, 2012.
57. N. P. F. Schneider, M. C. R. Moll and B. Rakowski, "Phishing Protection Design Documentation," Phishing Protection Design Documentation", p. http://wiki.mozilla.org/PhishingProtection:_Design_Documentation, 2014 September 2007.
58. A. Bergholz, G. P. Jeong Ho Chang, F. Reichartz and S. Strobel, "Improved Phishing 57 Detection using Model-Based Features," Proceedings of the Conference on Email and Anti-Spam (CEAS), 2008.
59. J. S. White, J. N. Matthews and J. L. Stacy, "A method for the automated detection phishing websites through both site characteristics and image analysis," SPIE Defense, Security, and Sensing, pp. 84080B-84080B-11, 2012.
60. D. E. GCIH and P. C. QSA, "Cybersecurity Top 10 Types of Phishing Emails," 2011.
61. V. M. M. D. Shahrivari and M. Izadi, "Phishing Detection Using Machine Learning Techniques," arXiv preprint arXiv, p. 11116, 2020.
62. TenserFlow, "TenserFlow," TenserFlow, 21 03 2020. [Online]. Available: https://www.tensorflow.org/resources/learn-ml. [Accessed 06 june 2021].
63. D. W. Forbes, "Forbes," 2020. [Online]. Available: https://www.forbes.com/sites/daveywinder/2020/02/28/google-confirms-new-ai-tool-scans-300-billion-gmail-attachments-every-week/#4a4d0813edd1. [Accessed 27 03 2021].

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ПРЕЗЕНТАЦІЯ

ДОДАТОК А

ДЕРЖАВНИЙ УНІВЕРСИТЕТІНФОРМАЦІЙНО – КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

МАГІСТЕРСЬКА РОБОТА
«Розробка методики виявлення фішингових електронних листів за допомогою алгоритмів машинного навчання»

Виконав: Студент групи ПДМ – 62 Гайшук Артур Олегович

Керівник: доцент кафедри, доктор філософії (PhD) Худік Богдан Олександрович

Київ - 2025

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: Підвищення ефективності виявлення фішингових електронних листів за допомогою алгоритмів машинного навчання та уніфікованого інтерфейсу AMSI.

Об'єкт дослідження: Процес вдосконалення методики виявлення фішингових електронних листів.

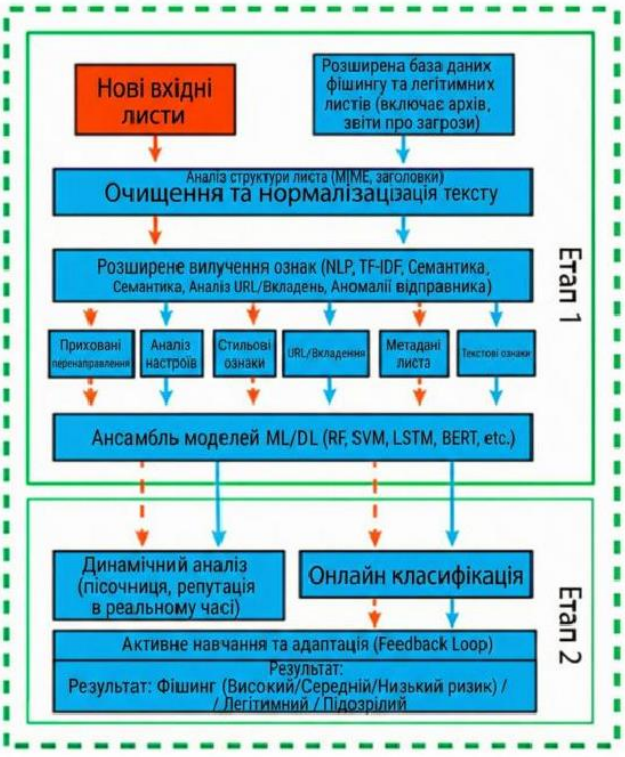
Предмет дослідження: Методи та алгоритми оптимізації обробки даних та взаємодії API для покращення здатності виявляти фішингові електронні листи.

АКТУАЛЬНІСТЬ РОБОТИ

Підхід/методика	Що використовує	Переваги	Недоліки
Класичні ML-моделі (Logistic Regression, Random Forest)	Статистичні ознаки листа: довжина, кількість посилань, частотність ключових слів, прості технічні поля	Прості в реалізації, швидко навчаються, відносно легко пояснювати результати.	Гірше виявляють складні та нові фішингові шаблони, сильно залежать від якості ручних ознак.
Глибокі нейронні мережі (LSTM / CNN для тексту)	Сире текстове представлення листа як послідовність слів або символів.	Краще захоплюють контекст і структуру тексту, ніж класичні моделі.	Потребують багато даних і обчислювальних ресурсів, результати важко інтерпретувати.
Transformer-моделі (BERT та подібні)	Контекстні векторні подання слів і фраз у листі.	Сучасний рівень точності для NLP-задач, добре виявляють прихований контекст і тонкі маніпуляції.	Висока обчислювальна вартість; працюють переважно з вмістом листа, без доступу до системних сигналів.
Запропонований додаток (ML + AMCI)	Текст листа і технічні ознаки (заголовки, URL) та сигнали з AMCI (скрипти, макроси, аномалії).	Поєднує контентний та системний аналіз; краще виявляє нові й складні атаки; модель можна гнучко донавчати.	Потребує інтеграції з AMCI та якісно розміченого датасету для навчання моделі.

ПОРІВНЯННЯ СТАНДАРТЕЗОВАНОГО МЕТОДУ ТА РОЗРОБЛЕНОГО

Стандартизований метод



Розроблений метод

Метрики оцінки класифікації

1. Accuracy (Точність загальна)

Частка всіх правильних прогнозів серед усіх передбачень. Показує, наскільки модель “в цілому” не помиляється.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

2. Precision (Точність позитивних)

Частка коректних позитивних передбачень серед усіх позитивних. Важливо для зменшення хибнопозитивних спрацьовувань.

$$Precision = \frac{TP}{TP + FP}$$

TP (True Positive) – кількість істинно позитивних результатів
TN (True Negative) – кількість істинно негативних результатів
FP (False Positive) – кількість хибнопозитивних результатів
FN (False Negative) – кількість хибнонегативних результатів

Метрики оцінки класифікації

3. Recall (Повнота)

Частка знайдених реальних позитивних випадків. Коли критично важливо не пропустити загрозу.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4. F1-Score

Гармонійне середнє між Precision і Recall. Баланс між “не помилятися” і “не пропускати”.

$$F1 = \frac{2 * (Precision * Recall)}{\left(Precision + Recall\right)}$$

TP (True Positive) – кількість істинно позитивних результатів
FN (False Negative) – кількість хибнонегативних результатів

Метрики оцінки класифікації

5. Macro Average

Середнє значення метрики по всіх класах окремо (без ваг). Показує ефективність моделі без урахування дисбалансу.

$$\text{Macro Avg} = \frac{1}{N} \sum_{i=1}^{N} \text{Metric}_i$$

6. Weighted Average

Середнє значення метрики з урахуванням розміру кожного класу. Відображає реальний вплив кожного класу на результат.

$$\text{weighted avg} = \frac{\sum_{i=1}^{N} (\text{метрика}_i \times \text{частка}_i)}{\sum_{i=1}^{N} \text{частка}_i}$$

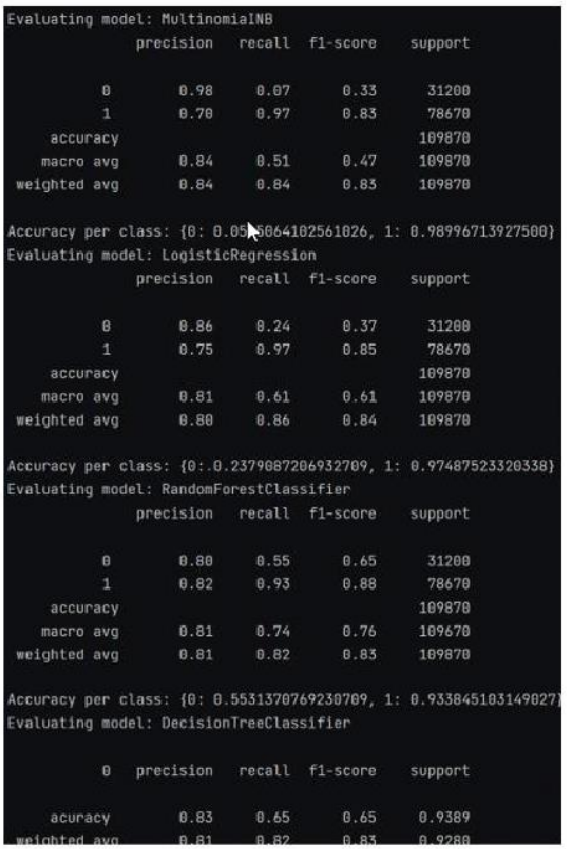
N – кількість класів (Metric i).
Metric i – значення вибраної метрики (Precision, Recall або F1) для класу i.
Частка i – кількість елементів у красі (i) поділена на загальну кількість елементів

ВХІДНІ ДАНІ

Використано датасет платформи Kaggle, який містив URL-адреси та відповідні мітки класів, що визначали належність кожного запису до категорії легітимних або фішингових сайтів.

14	www.coincoele.com.br/Scripts/smiles/?pt-br/Pagina	bad
15	www.henkdeinumboomkwekerij.nl/language/pdf_for	bad
16	perfectsolutionofall.net/wp-content/themes/twentyte	bad
17	lingshc.com/old_aol.1.3/?Login=&Lis=10&amp	bad
18	anonymeidentity.net/remax./remax.htm	bad
19	dutchweb.gtphost.com/zimbra/exch/owa/uleth/index	bad
20	www.avedeoiro.com/site/plugins/chase/	bad
21	asladconcentration.com/paplkuk1/webscrcmd=_hor	bad
22	www.regaranch.info/grafika/file/2012/atualizacao/w	bad
23	optimistic-pessimism.com/aoluserupdatealert.info.h	bad
24	mercadolivre.com.br.premiosfidelidade2012.com.br	bad
25	www.everythinggoingon.net/~gpeveryt/home/Email/	bad
26	mercadolivre.com.br.premiosfidelidade2012.com.br	bad
27	www.revitolcream.org/wp-content/plugins/all-in-one	bad

РЕЗУЛЬТАТ ОЦІНКИ ЕФЕКТИВНОСТІ МЕТОДІВ



```
Evaluating model: MultinomialNB
              precision    recall  f1-score   support

           0       0.98      0.07      0.13     31200
           1       0.70      0.97      0.83     78670
    accuracy                           0.72    109870
   macro avg       0.84      0.51      0.47    109870
weighted avg       0.84      0.84      0.83    109870

Accuracy per class: {0: 0.05[illegible]6064102561026, 1: 0.98996713927500}
Evaluating model: LogisticRegression
              precision    recall  f1-score   support

           0       0.86      0.24      0.37     31200
           1       0.75      0.97      0.85     78670
    accuracy                           0.76    109870
   macro avg       0.81      0.61      0.61    109870
weighted avg       0.80      0.86      0.84    109870

Accuracy per class: {0:.0.2379087206932709, 1: 0.97487523320338}
Evaluating model: RandomForestClassifier
              precision    recall  f1-score   support

           0       0.80      0.55      0.65     31200
           1       0.82      0.93      0.88     78670
    accuracy                           0.82    109870
   macro avg       0.81      0.74      0.76    109870
weighted avg       0.81      0.82      0.83    109870

Accuracy per class: {0: 0.5531370769230709, 1: 0.9338451031490271
Evaluating model: DecisionTreeClassifier

           0   precision    recall  f1-score   support

    accuracy       0.83      0.65      0.65     0.9389
weighted avg       0.81      0.82      0.83     0.9280
```

РЕЗУЛЬТАТ ОЦІНКИ ЕФЕКТИВНОСТІ МЕТОДІВ

```
Accuracy for Class 1 across different models (sorted by accuracy):

Model                     Accuracy
MultinomialNB             0.9895
LinearSVC                 0.9771
LogisticRegression        0.9748
MLPClassifier             0.9379
RandomForestClassifier    0.9339
DecisionTreeClassifier    0.9282
KNeighborsClassifier      0.9009
ML+AMCI                   0.9950
```

ПОРІВНЯННЯ РЕЗУЛЬТАТІВ ЕФЕКТИВНОСТІ МЕТОДІВ

	Accuracy	Precision	Recall	F1-score
MNB	0.9895	0.70	0.97	0.83
LR	0.9748	0.75	0.97	0.85
RF	0.9339	0.82	0.93	0.88
DT	0.9282	0.82	0.90	0.88
KNN	0.9009	0.82	0.90	0.87
SVC	0.9771	0.74	0.97	0.85
MP	0.9379	0.80	0.94	0.87
ML+AMCI	0.9959	0.86	0.96	0.89

MNB (Multinomial Naive Bayes) – мультиноміальний Наївний Байєсівський класифікатор
LR (Logistic Regression) – логістична регресія
RF (Random Forest) – випадковий ліс
DT (Decision Trees) – дерево рішень
KNN (k-nearest neighbor method) – метод k-найближчих сусідів
SVC – класичний класифікатор на основі методу опорних векторів
MP (Multilayer Perceptron) – багатошаровий персептрон
ML+AMCI – Машинне навчання з використанням AMCI

ВИСНОВКИ

1. Проаналізовано існуючі методи та підходи виявленя фішінгових листів та їх недоліки.

3. На основі запропонованих методик розроблено методику для виявлення фішінгових листів з використанням машинного навчання ML та уніфікованого інтерфейсу AMCI.

2. Проведено порівння ефективності різних моделей виявлення фішінгових листів.

4. Провдено тестування/порівняння різних методів виявлення фішіногивих листів.

ПУБЛІКАЦІЯ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Худік Б.О., Гайшук А.О.Розробка методики виявлення фішингових електронних листів « Виклики та рішення в програмній інхенерії! », 26 листопада 2025 р., Київ, Дерхавний університет інформаційно - комунікаційних технологій. Збірник тез. К.:ДУІКТ, 2025. С.215 -216.
2. Худік Б.О., Гайшук А.О. Розробка методики виявлення фішингових електронних листів // «Сучасні досягнення компанії Hewlett Packard Enterprise в галузі ІТ та нові мохливості її вивчення і застосування». 11 грудня 2025 р., Київ, Дерхавний університет інформаційно - комунікаційних технологій. VII Міхнародна науково-технічна конференція. Збірник тез. К.: ДУІКТ, 2025. С. 175 –176.

ДОДАТОК В

Цей скрипт додає спеціальну мітку (Label) до листів, які система розпізнає як потенційно небезпечні.

Основний код (Code.gs)

```
/**
 * МАГІСТЕРСЬКА РОБОТА: Система виявлення фішингу
 * Модулі: Preprocessing, Feature Extraction, Classification
 */

// Словник стоп-слів (спрощений)
const STOP_WORDS = ["і", "в", "на", "що", "це", "з", "the", "and", "is", "of", "to"];

/**
 * 1. МОДУЛЬ ПОПЕРЕДНЬОЇ ОБРОБКИ (Preprocessing)
 */
function preprocessText(text) {
  if (!text) return [];

  // Приведення до нижнього регістру та видалення спецсимволів
  let cleaned = text.toLowerCase().replace(/[^\w\sа-яіїєґ]/gi, ' ');

  // Токенізація (розбиття на слова)
  let tokens = cleaned.split(/\s+/);

  // Видалення стоп-слів та коротких символів
  return tokens.filter(word => word.length > 2 && !STOP_WORDS.includes(word));
}

/**
 * 2. МОДУЛЬ ВИЛУЧЕННЯ ОЗНАК (Feature Extraction)
```

```
 */
function extractFeatures(message) {
  const body = message.getPlainBody();
  const htmlBody = message.getBody();
  const subject = message.getSubject();

  let features = {
    urgencyWords: 0,
    hasLinks: 0,
    exclamationMarks: (subject.match(/!/g) || []).length,
    isHtml: htmlBody.includes('<a') ? 1 : 0,
    senderDismatch: 0
  };

  // Список маркерів терміновості
  const triggers = ["urgent", "account", "suspended", "verify", "password", "терміново",
"рахунок", "підтвердіть"];
  const tokens = preprocessText(body + " " + subject);

  tokens.forEach(token => {
    if (triggers.includes(token)) features.urgencyWords++;
  });

  // Перевірка кількості посилань
  const linkMatches = htmlBody.match(/href=/g);
  features.hasLinks = linkMatches ? linkMatches.length : 0;

  return features;
}

/**
 * 3. МОДУЛЬ КЛАСИФІКАЦІЇ (Classification)
```

```
*/
 function classifyPhishing(features) {
   // Проста лінійна модель (аналог Logistic Regression/SVM)
   // Ваги (Weights) можна підібрати експериментально
   let score = 0;

   score += features.urgencyWords * 1.5;
   score += features.hasLinks * 0.8;
   score += features.exclamationMarks * 0.5;
   score += features.isHtml * 1.2;

   // Поріг (Threshold)
   const threshold = 3.5;
   return score > threshold;
 }

 /**
  * ГОЛОВНА ФУНКЦІЯ (Entry Point)
  */
 function runPhishingDetection() {
   const threads = GmailApp.getInboxThreads(0, 10); // Перевіряємо останні 10 ланцюжків
   let label = GmailApp.getUserLabelByName("Phishing_Alert");

   if (!label) {
     label = GmailApp.createLabel("Phishing_Alert");
   }

   threads.forEach(thread => {
     const messages = thread.getMessages();
     const lastMessage = messages[messages.length - 1];
```

```
    const features = extractFeatures(lastMessage);
    const isPhishing = classifyPhishing(features);

    if (isPhishing) {
      thread.addLabel(label);
      console.log(Попередження: Лист "${thread.getFirstMessageSubject()}"
класифіковано як фішинг.);
    }
  });
}
```

Попередня обробка (Preprocessing)

У функції preprocessText
реалізовано:

* Tokenization: через split(/\s+/).
* Cleaning: видалення пунктуації через регулярні вирази.
* Stop Words: фільтрація через масив STOP_WORDS.
  Вилучення ознак (Feature Extraction)

У функції extractFeatures скрипт аналізує:

* Текстові ознаки: пошук слів-тригерів ("Urgent", "Account").
* Структурні: наявність HTML-посилань (<a> теги).
* Стилістичні: кількість знаків оклику в темі листа.