Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Вишковського Дениса Петровича

на здобуття ступеня вищої освіти магістра

Метод захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи

Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека та захист інформації
Освітня програма Кібербезпека та захист інформації

Шифр КРМКБЗІ.2301133.23.01.05 ПЗ

Виконав студент 2 курсу група КБЗІм-23-1 Денис ВИШКОВСЬКИЙ
Керівник канд. техн. наук, доцент Віктор ЧЕШУН
Нормоконтролер старший викладач Сергій МОСТОВИЙ

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

18 12 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Магістр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека та захист інформації

Освітня програма Кібербезпека та захист інформації

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ _____

2 09 2024 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Вишковському Денису Петровичу

1 Тема роботи Метод захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи

Керівник роботи канд.техн.наук, доцент Віктор ЧЕШУН

Затверджено наказом ректора університету від 26 08 2024 № 60

2 Строк подання студентом кваліфікаційної роботи на кафедру 2.12.2024р.

3 Вихідні дані до роботи Здійснити огляд методів і способів захисту інформаційних ресурсів від фішингових атак та несанкціонованого доступу, дослідити особливості фішингових атак та несанкціонованого доступу у сфері банківської справи, виявити перспективні напрямки вдосконалення технологій і розробити метод захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Теоретична база понять загроз фішингу та існуючі методи захисту. Мета та ціль дослідження. Системний аналіз обраного методу , порівняння та недоілки. Тестування алгоритму на виявлення шкідливих посилань та вкладень. Реалізація запропонованого алгоритму. Висновки.

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

–

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7 Дата видачі завдання 2 09 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Ґрунтовне ознайомлення та дослідження предметної галузі	15.09.2024	Виконано
Визначення змісту, структури кваліфікаційної роботи	22.09.2024	Виконано
Підготовка першого розділу кваліфікаційної роботи	29.09.2024	Виконано
Підготовка другого розділу кваліфікаційної роботи	10.10.2024	Виконано
Підготовка третього розділу кваліфікаційної роботи	20.10.2024	Виконано
Підготовка статті/тези за темою кваліфікаційної роботи	4.11.2024	Виконано
Підготовка четвертого розділу кваліфікаційної роботи	17.11.2024	Виконано
Підготовка та оформлення ілюстративного матеріалу	24.11.2024	Виконано
Оформлення кваліфікаційної роботи	24.11.2024	Виконано
Попередній захист кваліфікаційної роботи	27.11.2024	Виконано
Захист кваліфікаційної роботи на засіданні ЕК	19.12.2024	Виконано

Студент Денис ВИШКОВСЬКИЙ

Керівник кваліфікаційної роботи Віктор ЧЕШУН

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи

Автор роботи: Вишковський Денис Петрович

Керівник роботи: к.т.н., доц. Чешун Віктор Миколайович

Загальний обсяг роботи: 87 сторінок, 32 рисунки, 15 таблиць, 5 додатків, 69 посилань.

Ключові слова: види фішингу, фішингові листи в банківській сфері, алгоритм фільтрування листів , захист в банківській сфері.

Технологія створення клієнтського алгоритму виявлення фішингу дозволяє ефективно аналізувати листи, що надходять співробітникам банку, виявляти у них шкідливі посилання та вкладення і автоматично позначати лист як шкідливий, потім переносити його у спам без зайвих рухів клієнта.

У магістерській роботі вирішується одна з ключових задач у сфері кібербезпеки в банківської галузі – мінімізація ризику успішних фішингових атак через автоматичне виявлення та блокування підозрілих електронних листів. Таким чином, робота вирішує актуальну проблему захисту банківських клієнтів, використовуючи інноваційний підхід до аналізу тексту з акцентом на реальну практичну застосовність.

Запропоновано клієнтський алгоритм виявлення фішингу електронної пошти, що має такі переваги над іншими алгоритмами як, локальне виконання – усі обчислення виконуються на пристрої клієнта, що забезпечує приватність і знижує залежність від серверної інфраструктури. Naive Bayes є обчислювально легким алгоритмом, дозволяє швидко обробляти вхідні листи навіть на пристроях з обмеженими ресурсами. Модель можна оновлювати, додаючи нові шаблони фішингових атак. Алгоритм легко інтегрується в банківські клієнтські програми.

2.12.2024

ANNOTATION

Theme of qualification work: A method of protection against phishing attacks and unauthorized access in the field of banking

Author of the work: Vyshkovskyi Denys Petrovich

Mentor: Ph.D. Cheshun Victor Mykolayovych

Total volume of work: 87 pages, 32 figures, 15 tables, 5 appendices, 69 links.

Keywords: types of phishing, phishing emails in the banking sector, letter filtering algorithm, protection in the banking sector.

The technology for creating a client-side phishing detection algorithm allows you to effectively analyze letters received by bank employees, detect malicious links and attachments in them, and automatically mark the letter as malicious, then transfer it to spam without unnecessary actions of the client.

The master's work solves one of the key tasks in the field of cyber security in the banking industry – minimizing the risk of successful phishing attacks through automatic detection and blocking of suspicious e-mails. Thus, the work solves the actual problem of protecting bank clients, using an innovative approach to text analysis with an emphasis on real practical applicability.

A client-side e-mail phishing detection algorithm is proposed, which has such advantages over other algorithms as local execution, all calculations are performed on the client's device, which ensures privacy and reduces dependence on the server infrastructure. Speed of operation, Naive Bayes is a computationally light algorithm that allows fast processing of incoming emails even on devices with limited resources. Adaptability, the model can be updated by adding new phishing attack patterns. Integration, the algorithm is easily integrated into banking client programs.

2.12.2024

ЗМІСТ

ВСТУП

Захист банківських систем від фішингових атак та несанкціонованого доступу набуває все більшої актуальності в умовах цифрової трансформації. Зростання популярності онлайн-банкінгу, мобільних додатків і віддаленого обслуговування клієнтів робить банківський сектор особливо вразливим до кіберзагроз. Це вимагає постійного вдосконалення механізмів безпеки, спрямованих на захист конфіденційної інформації клієнтів і зміцнення їхньої довіри до банківських послуг.

Фішингові атаки, спрямовані на викрадення персональних даних і облікових записів клієнтів, є однією з найбільш поширених загроз для банків. За статистикою, понад 70% кіберзлочинів пов’язані саме з фішингом. Банківський сектор є привабливою мішенню через високу цінність даних, які зберігають фінансові установи.

Водночас у цифровому банкінгу впроваджуються новітні підходи до безпеки даних, такі як використання штучного інтелекту для моніторингу та аналізу аномалій у поведінці користувачів. Це дозволяє банківським установам завчасно виявляти потенційно небезпечні дії та блокувати шахрайські операції до завдання шкоди. Таким чином, захист від фішингових атак і несанкціонованого доступу стає невід’ємною частиною стратегії сталого розвитку банківських послуг в умовах глобалізації кіберзагроз. Основним завданням цієї роботи є аналіз сучасних методів захисту банківських систем від кіберзагроз і розробка рекомендацій щодо вдосконалення таких методів на основі останніх наукових досягнень..

Основне завдання полягає у проведенні огляду сучасних методів захисту інформації в банківському секторі, зокрема таких як багатофакторна аутентифікація (MFA), DNS- та URL-фільтрація, поведінковий аналіз та інші технології, що використовуються для запобігання фішингу в банківський сферх обслуговування та захисту даних.

Також однією з головних задач є вивчення та аналіз недоліків та обмежень існуючих підходів у сфері кібербезпеки банківських систем на основі наукових

джерел та реальних прикладів. Ознайомлення з останніми досягненнями в області штучного інтелекту, машинного навчання, щоб посилити захист від фішингу. Розробка рекомендації для вдосконалення систем захисту, враховуючи сучасні технологічні тенденції та специфіку банківської діяльності. Оцінка ефективності запропонованих рекомендацій на основі порівняльного аналізу, включаючи моделювання або статистичні методи, щоб забезпечити доказову базу для їх реалізації. Цей підхід допоможе визначити найефективніші стратегії для забезпечення надійного захисту інформаційних систем банків від сучасних кіберзагроз та фішингових атак, враховуючи постійний розвиток кіберзлочинності та технологій захисту.

Об’єктом дослідження є система інформаційної безпеки банківської установи, що забезпечує конфіденційність, цілісність і доступність інформації. Вона охоплює технічні, організаційні та процедурні заходи, які спрямовані на захист від кіберзагроз, включаючи фішингові атаки, витоки даних і несанкціонований доступ. Предметом дослідження виступають технології та методи захисту банківських систем. Аналіз цих технологій дозволяє оцінити їхню ефективність у протистоянні сучасним кіберзагрозам..

Цей вибір об'єкта і предмета дозволяє оцінити, наскільки ефективні новітні підходи до захисту від фішингових атак у банківському секторі, та запропонувати вдосконалення, виходячи з аналізу останніх наукових досліджень і практичного досвіду в цій сфері .У сучасному банківському секторі захист від фішингових атак та несанкціонованого доступу став надзвичайно важливим завданням, особливо на фоні цифрової трансформації, яка змінила способи надання банківських послуг. З впровадженням нових технологій, таких як онлайн-банкінг та мобільні платформи, з'явились нові виклики в сфері безпеки. Фішингові атаки стають все більш складними, а зловмисники використовують соціальну інженерію для маніпуляції з користувачами, що може призводити до серйозних фінансових втрат для банків та їхніх клієнтів.

Актуальність роботи зумовлена тим, що, попри наявність різноманітних методів захисту, таких як антифішингові фільтри, двофакторна аутентифікація та

системи виявлення аномалій, зловмисники постійно вдосконалюють свої методи, а це обумовлює необхідність розробки нових, більш ефективних і адаптивних підходів до протидії фішингу, які відповідатимуть сучасним викликам**.**

Ця магістерська робота присвячена створенню методу захисту від фішингових атак та несанкціонованого доступу в банківській сфері, заснованого на алгоритмі Naive Bayes. Запропонований метод спрямований на виявлення фішингових листів шляхом багаторівневого аналізу текстів повідомлень, URL-адрес і вкладень, що дозволяє ідентифікувати потенційні загрози з високою точністю. У роботі детально розглянуто типи фішингових атак, їх характерні риси та механізми реалізації у банківській сфері, проаналізовано наявні методи захисту від фішингу та їх недоліки, створено новий клієнтський алгоритм для автоматичного аналізу електронних листів на основі статистичних моделей, що враховують ймовірнісний підхід до класифікації загроз.

Метою кваліфікаційної роботи є розробка методу захисту від фішингових атак та несанкціонованого доступу, який забезпечить надійне виявлення шкідливих елементів у банківських операціях, зокрема в електронній пошті та інших цифрових каналах комунікації, з підвищенням ефективності у порівнянні з існуючими рішеннями.

Об'єктом дослідження є фішингові атаки, що спрямовані на банківський сектор, і механізми забезпечення інформаційної безпеки в цьому контексті.

Предметом дослідження є алгоритми та технології, які дозволяють виявляти й запобігати фішинговим атакам та несанкціонованому доступу у банківській сфері**.**

Щоб реалізувати програму досліджень необхідно:

а) визначити найбільшу вразливість у банківській сфері за допомогою системного аналізу та визначити напрямки вдосконалення технології, що можуть бути використані в підвищені ефективності;

б) затвердити основні положення роботи клієнтського алгоритму виявлення фішингу електронної пошти в банківській сфері;

в) зробити алгоритмічну та математичну модель клієнтського алгоритму виявлення фішингу електронної пошти в банківській сфері;

г) здійснити алгоритмічну реалізацію технології;

д) довести покращення теоретичних і алгоритмічних рішень технології.

Наукова новизна отриманих результатів: полягає у створенні й реалізації методу виявлення фішингових атак та несанкціонованого доступу в банківській сфері на основі алгоритму Naive Bayes, що демонструє поліпшення точності, продуктивності та адаптивності до сучасних загроз. Основні аспекти новизни включають:

Практична значимість отриманих результатів полягає у підвищенні кібербезпеки банківських установ, запропонований алгоритм може стати основою для розробки інтегрованих рішень захисту від фішингових атак. Універсальність алгоритму, використання Naive Bayes дозволяє адаптувати алгоритм для інших сфер, включаючи загальну електронну комунікацію чи інші галузі з високим ризиком фішингових атак.

Публікації. За темою магістерської роботи зроблено доповідь і опубліковано тези на міжнародній науково-практичній конференції.

1 ТЕОРЕТИЧНА БАЗА ПОНЯТЬ ЗАГРОЗ ФІШИНГУ ТА ІСНУЮЧІ МЕТОДИ ЗАХИСТУ

1.1 Огляд технологій фішингових атак, дослідження специфіки атак у банківській сфері

Фішинг - це тип кіберзагрози, коли зловмисники намагаються обманом змусити користувачів розкрити конфіденційні дані, такі як логіни, паролі, номери банківських карток та іншу особисту інформацію. Основний механізм фішингу полягає в тому, що шахраї намагаються завоювати довіру своїх жертв, видаючи себе за надійне джерело, наприклад, банк, соціальну мережу або популярний сервіс.

Фішингові вебсайти, також маскує інтернет сторінку , яка виглядає так само, як справжній сайт схожий на легальний. Нічого не підозрюючи, користувачі перенаправляються на цей вебсайт після натискання на посилання або рекламу, вбудовану в електронний лист (клік джекінг).Якщо користувач продовжує взаємодіяти з фальшивим вебсайтом, конфіденційна інформація розкривається і збирається фішером [1].

Електронний фішинг – цей тип фішингу, який часто називають «фішингом обману», є одним з найпоширеніших видів атак. Спочатку зловмисники надсилають користувачам електронного листа від імені повноважної особи і використовують тактику соціальної інженерії, щоб підвищити їхню активність. Потім вони просять користувачів завантажити ресурси або перейти за посиланнями.

Ці посилання ведуть на шкідливі вебсайти, які викрадають облікові дані або встановлюють шкідливий код (шкідливе програмне забезпечення).

У випадку з файлами, завантажений файл (зазвичай PDF) містить шкідливий вміст і шкідливе програмне забезпечення встановлюється, як тільки користувач відкриває документ.

Розглянемо ознаки фішингового електронного листа:

– справжня інформація – у цьому випадку можна знайти контактну

інформацію та іншу надійну інформацію про фальшиву організацію. Помітно, що адреса електронної пошти відправника вказано в неправильному домені або написана з помилкою;

– шкідливий і безпечний код намагається обдурити Exchange Online Protection (EOP). Наприклад, неправильно написані посилання або завантаження;

– скорочені посилання – рекомендується не натискати скорочені посилання, оскільки вони використовуються для обману безпечних шлюзів електронної пошти;

– підроблені логотипи брендів можуть містити підроблені та шкідливі атрибути HTML, тому слід шукати справжні логотипи у своїх повідомленнях;

– електронні листи, які містять лише зображення та дуже мало тексту – їх слід ігнорувати, оскільки за зображеннями може ховатися шкідливий код.

Також розглянемо принцип дії шахраїв:

– фішер створює шахрайський електронний лист із посиланням або вкладенням (фаза планування);

– фішер здійснює атаку, надсилаючи фішинговий електронний лист потенційній жертві за допомогою відповідного середовища (фаза проведення атаки);

– посилання (якщо натиснуто) спрямовує користувача на шахрайський вебсайт або на завантаження шкідливого програмного забезпечення у разі натискання вкладення (фаза взаємодії);

– шкідливі вебсайти спонукають користувачів надавати конфіденційну інформацію та облікові дані, які потім збираються зловмисниками і з метою незаконного збагачення використовуються для вчинення шахрайства (фаза реалізації цінності).

У більшості випадків фішери не використовують облікові дані безпосередньо, щоб не розкрити анонімність вони зазвичай перепродують отримані облікові дані та інформацію на вторинному ринку.

Перейдемо до HTTP-фішингу. Протокол передачі гіпертексту (HTTPS) вважається безпечним з'єднанням, оскільки він використовує шифрування для

підвищення рівня безпеки. Багато законних організацій використовують HTTPS замість протоколу передачі гіпертексту (HTTP) .Однак зловмисники тепер можуть використовувати HTTPS для посилань, Хоча ця техніка є частиною фішингових атак, які вони включають у фішингові електронні листи, сучасніший підхід.

Характерні ознаки фішингу з використанням захищеного протоколу передачі гіпертексту (HTTPS):

– гіпертекст – це інтерактивне посилання, вбудоване в текст, щоб приховати фактичну URL-адресу

– скорочення посилання – користувачі повинні переконатися, що це посилання має правильний формат довгого хвоста, а також містить усі частини URL адреси.

Спливаючий фішинг – зловмисники можуть розмістити шкідливий код у невеликому вікні повідомлення, яке з'являється під час відвідування вебсайту. Новіші версії спливаючого фішингу використовують функцію сповіщення браузера. Наприклад, коли користувач відвідує вебсайт, браузер відображає адресу сайту і зазначає повідомлення із назвою сайту «хоче показати вам повідомлення». Якщо на цьому етапі користувач натисне кнопку «дозволити», у спливаючому вікні буде встановлено шкідливий код.

Ознаки спливаючого фішингу:

– повноекранний режим – автоматична зміна розміру екрана може бути індикатором, оскільки зловмисні спливаючі вікна зазвичай можуть спричинити перехід браузера в повноекранний режим;

– невідповідності – слід перевірити на наявність орфографічних помилок і нестандартних схем кольорів.

Фішинг на основі шкідливого програмного забезпечення, як випливає з назви, це тип фішингової атаки ,що здійснюється шляхом запуску шкідливого програмного забезпечення на комп'ютері користувача. Шкідливе програмне забезпечення завантажується на комп'ютер жертви за допомогою одного з методів соціальної інженерії або шляхом технічної експлуатації вразливостей (наприклад, вразливостей в браузері). Шкідливе програмне забезпечення Panda було одним з

найуспішніших шкідливих програм, виявлених Fox-ITв 2016 році. Це шкідливе програмне забезпечення націлене на операційні системи (ОС) Windows. Він поширюється через фішингові кампанії та використовує веб об'єкти, скріншоти активності користувачів реєстрацію клавіатури, вставку буферу обміну (перехоплення паролів та в ставку їх у поля форм) та загальні ресурси віртуальних мережевих обчислень (VNC); У 2018 році шкідливе програмне забезпечення Panda розширило свої цілі на криптовалютні біржі та сайти соціальних мереж. Існує багато форм фішингових атак на основі зловмисного програмного забезпечення, деякі з них обговорюються нижче.

Клавіатурні шпигуни та реєстратори екрану Клавіатурні шпигуни це тип шкідливого програмного забезпечення, що використовується фішерами і встановлюються на комп'ютер користувача через троянський додаток до електронного листа або шляхом прямого завантаження. Програмне забезпечення відстежує дані та записує натискання клавіш користувача, які потім надсилаються фішеру. Фішери використовують клавіатурні шпигуни, щоб отримати конфіденційну інформацію про жертву, таку як імена, адреси, паролі та інші важливі дані. Кейлоггери також можуть використовуватися для цілей, непов'язаних з фішингом. Кейлоггери можуть виявляти зміни URL і записувати інформацію як допоміжні об'єкти браузера, які дозволяють зловмисникам контролювати всі функції, відстежувати введення клавіатури і миші як драйвери пристроїв, відстежувати введення користувача і відображати його як реєстратор екрану, а також можуть бути реалізовані багатьма іншими способами.

Програми-вимагачі – це тип зловмисного програмного забезпечення, яке шифрує дані користувача після запуску виконуваної програми на пристрої. У цьому типі атаки ключ дешифрування зберігається, доки користувач не заплатить викуп. Щорічне використання програмного забезпечення вимагача несе відповідальність за десятки мільйонів доларів США. Що ще гірше, це важко виявити при розробці нових варіантів, що полегшує ухилення від багатьох антивірусних систем і систем виявлення вторгнень (Latto, 2020). Програми-вимагачі зазвичай доставляються на пристрій жертви через фішингові електронні листи. Згідно зі звітом [2], 93% усіх

фішингових електронних листів містили програми-вимагачі для шифрування. Фішинг, як атака соціальної інженерії, переконує жертву виконувати дії, не знаючи про шкідливу програму.

Фішинг ін'єкції вмісту (із впровадженням вмісту) означає вставлення неправдивого вмісту на законний сайт. Цей зловмисний вміст може неправильно спрямувати користувача на підроблені вебсайти, змусивши користувачів розкрити свою конфіденційну інформацію хакеру, або це може призвести до завантаження шкідливого програмного забезпечення на пристрій користувача [3].

Шкідливий вміст може бути впроваджено на законний сайт трьома основними способами:

– хакер використовує вразливість системи безпеки та компрометує вебсервер;

– хакер використовує вразливість міжсайтового сценарію (XSS), яка є недоліком програмування, що дозволяє зловмисникам вставляти клієнтські сценарії на вебсторінки, які переглядатимуть відвідувачі цільового сайту;

– хакер використовує вразливість до мови структурованих запитів (SQL), що дозволяє хакерам викрасти інформацію з бази даних вебсайту, виконуючи команди бази даних на віддаленому сервері.

1.2 Методи та технології захисту від фішингу й несанціонованого доступу

Ключовий елемент системи контролю доступу, який дозволяє підтвердити, що людина є саме тим, за кого себе видає, це аутентифікація. Наприклад, під час перетину кордону демонстрація паспорта митному агенту є формою аутентифікації. У сфері кібербезпеки найпоширенішим прикладом є вхід до онлайн-сервісів. Користувач вводить логін і пароль для входу в Gmail або мобільний банківський додаток і у цей момент система підтверджує їх відповідність, тим самим ідентифікуючи користувача.

Фактори аутентифікації – це різні способи підтвердження особи.

Двофакторна аутентифікація (2FA) – це метод підтвердження особи, який використовує два різних фактори для встановлення ідентичності користувача. Простіше кажучи, перед отриманням доступу користувач повинен пройти перевірку двома способами. 2FA є однією з реалізацій багатофакторної аутентифікації (рисунок 2.1).

Рисунок 1.1 – Принцип аунтифікації

У рамках двофакторної аутентифікації можуть застосовуватися кілька видів факторів перевірки, які зазвичай використовуються для посилення безпеки:

– знання – інформація, яка відома виключно користувачу, наприклад, пароль або відповідь на секретне запитання;

– володіння – фактор базується на наявності у користувача певного фізичного об’єкта, такого як апаратний ключ для генерування кодів доступу чи мобільний пристрій, через який надсилаються коди;

– біометричні дані – унікальні фізіологічні характеристики користувача, що застосовуються для аутентифікації (відбитки пальців, розпізнавання обличчя чи сканування сітківки ока);

– місцезнаходження – фактор використовує географічні дані, наприклад GPS, для обмеження доступу лише для користувачів, які знаходяться у визначеному регіоні.

Двофакторна аутентифікація може здійснюватись різними способами. Найпоширенішим варіантом є комбінація імені користувача, пароля та додаткового підтвердження через SMS. Наприклад, під час створення облікового запису користувач вказує унікальне ім’я, пароль і номер мобільного телефону. Під час входу в обліковий запис користувач вводить свої облікові дані (ім’я користувача та пароль), що становить перший фактор аутентифікації – знання (користувач підтверджує, що знає свої дані для доступу). Після цього система надсилає автоматичне SMS із кодом, який слід ввести. Якщо код правильний, це підтверджує другий фактор аутентифікації – володіння (користувач доводить, що має доступ до мобільного пристрою). Виконавши обидва етапи перевірки, користувач проходить 2FA та отримує доступ до свого облікового запису.

Парольна безпека стає дедалі вразливішою через зростання кіберзагроз, таких як фішингові атаки, атаки методом грубої сили, перехоплення даних і повторне використання паролів. Зловмисники часто викрадають облікові дані для входу, обмінюють їх або продають для здійснення атак із вкиданням облікових даних. Саме тому 2FA стає стандартною практикою для захисту інформації. Додаткова перевірка особи особливо актуальна в умовах зростання кількості віддалених співробітників. Оскільки фізична присутність працівників в офісі більше не є гарантією безпеки, 2FA допомагає запобігти компрометації їхніх облікових записів. Експерти з кібербезпеки рекомендують увімкнути 2FA, коли це можливо, а також вимагати її від сервісів, які обробляють конфіденційну інформацію. Хоча 2FA не є повністю захищеним блоком від зломів, її обходження значно складніше й дорожче для зловмисників порівняно з паролями без додаткового захисту.

Методи фільтрації електронної пошти використовуються, відповідно, в фільтрації електронної пошти і визначають, наскільки ефективно маршрутизується пошта.

Організація повинна подумати про те, чого вона хоче від рішення для фільтрації електронної пошти. Комбінація наступних методів може допомогти організаціям досягти максимальної ефективності:

– фільтри електронної пошти на основі репутації, використовуються для боротьби зі спамом та забезпечення доставки легітимних листів шляхом перевірки відправників на основі баз даних репутації. Репутаційні списки блокування (RBL) містять домени, URL-адреси та IP-адреси, які були проаналізовані й визнані потенційно небезпечними. Це один із ключових інструментів, який організації застосовують для фільтрації спаму та підвищення безпеки;

– список дозволених відправників дає можливість організаціям визначати довірених відправників, додаючи їх до спеціального списку, завдяки чому їхні електронні листи завжди приймаються;

– список блокування дозволяє організаціям створити перелік відправників, від яких електронні листи блокуються автоматично, гарантуючи, що небажані повідомлення не потрапляють до користувачів;

– тимчасове блокування (сірий список) тимчасово відхиляє листи від невідомих відправників. Якщо лист є легітимним, сервер-відправник повторить спробу відправки через деякий час, після чого повідомлення буде прийняте. Цей підхід ефективно захищає від більшості спам-розсилок;

– аналіз контенту дозволяє блокувати електронні листи на основі їхнього вмісту. Наприклад, якщо в повідомленні містяться заборонені слова або вкладення, які організація вирішила не пропускати, такі листи блокуються. Баєсовий аналіз також є формою контентної фільтрації, яка покращує визначення спаму, поступово навчаючись на основі аналізу вхідних повідомлень. Зі збільшенням кількості проаналізованих листів ефективність такого аналізу зростає.

Метод фільтрації дозволяє власникам доменів публікувати політики у своїх DNS-записах, які визначають, хто має право надсилати електронні листи від імені їхнього домену. При отриманні електронного листа отримувач перевіряє DNS-запис домену, ідентифікує відправника та вирішує, чи прийняти, відхилити або відправити лист у карантин. Окрім цього, система надає власникам доменів зворотний зв’язок щодо використання їхнього домену в електронних листах. Така інформація дозволяє виявляти та зупиняти несанкціоноване використання домену, а також покращувати показники доставки пошти. Розглянемо, як можна

захиститися від «цілеспрямованого фішингу». У таких атаках зазвичай не використовується зловмисне програмне забезпечення, тому ефективними стають заходи на основі перевірки автентичності. Наприклад, використання технології "Доменна аутентифікація пошти" (Domain-based Authentication), яка перевіряє справжність електронного листа за базою даних відправників, допомагає виявляти фішингові спроби.

Метод шифрування даних використовує протоколи SSL/TLS. Хоча «SSL» і «TLS» часто використовуються як взаємозамінні, вони представляють різні версії протоколів безпеки, які використовуються для шифрування даних між вебсервером і браузером. Еволюція цих протоколів призвела до значного прогресу в безпеці та продуктивності:

– SSL 3.0 (1996) зараз вважається застарілим через вразливості;

– TLS 1.0 (1999) і TLS 1.1 (2006) також застарілі і вважаються слабкими;

– TLS 1.2 (2008) широко використовується та забезпечує надійний захист;

– TLS 1.3 (2018) остання версія, що забезпечує найнадійніше шифрування з покращеною продуктивністю.

TLS забезпечує удосконалені функції безпеки та використовує більш ефективні алгоритми шифрування в порівнянні з SSL. Наприклад, TLS 1.3 знижує затримку під час рукостискання, що сприяє швидшому встановленню з'єднання. Незважаючи на ці переваги, термін «SSL-сертифікат» усе ще широко використовується в індустрії. SSL/TLS-сертифікати виконують три ключові завдання: забезпечують шифрування даних, підтверджують автентичність сервера та створюють довіру між користувачем і вебсайтом.

Шифрування забезпечує конфіденційність і цілісність даних, які передаються між вебсервером і браузером. Це особливо важливо для захисту чутливої інформації, наприклад, облікових даних, платіжних даних або особистих відомостей від можливого перехоплення.

Аутентифікація вебсайту гарантує, що вебсайт дійсно є тим, за кого себе видає, запобігаючи шахрайським спробам видавати себе за нього. Це включає перевірку права власності на домен і, за необхідності, підтвердження інформації

про організацію. Довіра створюється завдяки впевненості відвідувачів у тому, що їхні дані в безпеці. Видимі ознаки, такі як значок замка та HTTPS у рядку адреси, свідчать про захищеність з'єднання та турботу про конфіденційність користувачів.

Принципи реалізації протоколу TLS демонструє рисунок 1.2.

Рисунок 1.2 Принцип роботи протокола TLS

SSL/TLS використовує комбінацію асиметричного та симетричного шифрування для створення безпечного з'єднання. Процес включає кілька етапів:

– рукостискання SSL/TLS – під час підключення браузера до захищеного вебсайту (HTTPS) сервер надає свій SSL/TLS-сертифікат;

– перевірка – браузер перевіряє сертифікат через довірений центр сертифікації (CA), оцінюючи його термін дії, підпис та статус відкликання;

– генерація ключа сеансу – якщо сертифікат підтверджено, браузер створює симетричний ключ для сеансу, шифрує його за допомогою відкритого ключа сервера та надсилає назад на сервер;

– шифрування – сервер розшифровує симетричний ключ за допомогою свого закритого ключа. Після цього сервер і браузер використовують цей спільний ключ для шифрування даних протягом сеансу;

– встановлення безпечного з'єднання – безпечний канал створено, і всі передані дані шифруються за допомогою симетричного ключа.

Існує три основних типи сертифікатів SSL/TLS, кожен з яких пропонує різні рівні перевірки та підходить для конкретних потреб галузі.

Сертифікати DV забезпечують базовий рівень перевірки, підтверджуючи право власності на домен через електронну пошту або DNS-записи. Вони видаються дуже швидко (зазвичай, протягом кількох хвилин або годин) і є одним із найдоступніших за ціною варіантів. Ці сертифікати ідеально підходять для невеликих проєктів, таких як особисті блоги, інформаційні сайти чи малий бізнес, яким потрібне шифрування без поглибленої перевірки.

Сертифікати OV забезпечують середній рівень перевірки, підтверджуючи як право власності на домен, так і основні дані організації. Процес видачі займає від одного до трьох днів і має помірну вартість. Ці сертифікати підходять для вебсайтів, що збирають дані користувачів, оскільки допомагають створити впевненість у надійності та легітимності організації

Сертифікати EV забезпечують найвищий рівень перевірки, вимагаючи ретельного підтвердження юридичного, фізичного та операційного існування організації. Процес видачі триває від одного до п’яти днів і є найдорожчим серед сертифікатів. Вони ідеально підходять для сайтів електронної комерції, фінансових установ і великих брендів, де необхідна максимальна довіра користувачів.

1.3 Огляд статистичних даних про фішинг, випадки шахрайства та збитки в банківському секторі

У 2023 році постачальники платіжних послуг, торговці та споживачі зазнали збитків на суму 833 мільйони гривень через шахрайські операції з платіжними

картками, що на 73% більше порівняно з 2022 роком. Кількість випадків шахрайства зросла на 25%, досягнувши 272 тисячі операцій. Середня сума однієї незаконної транзакції склала 3065 гривень, порівняно з 2200 гривнями у 2022 році.

Зростання активності платіжних карток також відзначалося, кількість активних карток збільшилася на 13%. Однак більшість шахрайських операцій (83%) відбувалися в інтернеті, тоді як лише 17% – через фізичні пристрої, такі як банкомати і торговельні термінали.

Основною причиною збитків стало розголошення особистих даних клієнтами через методи соціальної інженерії, які спричинили 80% від загальної суми втрат у 2023 році, що значно більше порівняно з 53% у 2022 році. Водночас середня сума шахрайських операцій в інтернеті зросла на 31% – до 3150 гривень, порівняно з 2408 гривнями в 2022 році.

Частка збитків, завданих операторами поштового зв’язку та небанківськими фінансовими установами, залишалася мінімальною – лише 0,4% від загальної суми втрат і 0,2% за кількістю випадків шахрайства. Основний фокус шахраїв був спрямований на банки та інтернет-платформи.

Про кібербезпеку 65% неповнолітніх та 48% пенсіонерів дізнаються з соцмереж, 38% та 23% – від друзів та родичів, а от кожен третій українець віком від 65 років – з освітніх проєктів. Стать практично не впливає на вірогідність натрапити на шахрая. І чоловіки (78%), і жінки (88%) отримують шкідливі посилання від шахраїв.

Дані таблиці 1.1 свідчать про необхідність посилення фінансової грамотності серед населення та впровадження технологій, які допоможуть знизити ризики шахрайства.

Чоловікам удвічі частіше за жінок надходять шахрайські SMS – 9% проти 4% від всіх випадків відповідно

Шахрайства за принципом поширеності посортовані в таблиці 1.2.

Таблиця 1.1 – Статистичні дані коректних дій користувачів

Правила кібербезпеки	до 18 років	після 65 років
Не відкривають посилання від незнайомців	48%	28%
Не розголошують платіжні дані банківської картки	30%	29%
Звертають увагу на оцінки і відгуки	38%	19%
Не обговорюють угоду поза платформою (у сторонніх месенджерах)	39%	38%
Перевіряють посилання	14%	7%
Перевіряють номер продавця/покупця в базах відгуків	26%	21%
Уважно оцінюють текст в SMS чи пошті	22%	14%
Звертають увагу на манеру спілкування	45%	38%

Таблиця 1.2 Класифікація шахрайств за принципом поширеності

Типи шахрайств	Частка (%)
З втраченими і викраденими пластиковими картками	72,2
З підробленими картками	20,5
З картками, не отриманими законним держателем	2,8
З використанням рахунку	1,4
Інші форми шахрайств	3,1

Найпоширенішим методом крадіжки даних, а потім і грошей українців, є соціальна інженерія, яка інтегрується у фішинг. В Українській міжбанківській Асоціації членів платіжних систем ЕМА «Мінфіну» повідомили, що оцінюють приріст таких крадіжок із 92% у 2022 році до 99% у 2023-му.

Втрати в інтернеті незрівнянно менші. У ЕМА оцінюють розрахунковий дохід карткових шахраїв у 2022 році на рівні 968,5 млн грн, а разом із інтернет-крадіжками – понад 1 млрд грн.

Індустрія програм-вимагачів залишається однією з найголовніших загроз для організацій у всьому світі. Адже не лише витік даних, їх відновлення, довготривала боротьба з наслідками (виплати, судові позови, репутація тощо) приносять збитки.

Зловмисники обирають компанії з великими обсягами критично важливих даних, адже є велика ймовірність того, що вони заплатять викуп, щоб уникнути витоку або втрати даних, а також для того, щоб не заплямувати свою репутацію. Це особливо важливо для організацій, чиї дані мають високу комерційну цінність, або ж для тих, хто працює в галузях, де безперервність роботи є критично важливою, наприклад, у фінансовому секторі, охороні здоров'я або енергетиці. Виплата викупу дозволяє уникнути великих фінансових втрат, а також мінімізувати ризики для бізнесу, що може поставити під загрозу довіру клієнтів та партнерів. Зловмисники, розуміючи ці фактори, все частіше обирають своїми цілями великі корпорації та інші організації, що мають значні обсяги конфіденційної інформації. Останніми роками спостерігається значне зростання кількості кібератак, зокрема атак за допомогою програм-вимагачів, що стає серйозною проблемою для всіх бізнес-секторів.

Разом із ростом кількості атак збільшується й кількість організованих злочинних груп, які мають змогу здійснювати складніші та більш масштабні операції. Ці групи використовують більш сучасні методи, зокрема фішинг, соціальну інженерію та різноманітні уразливості в програмному забезпеченні, що дозволяє їм ефективно проникати в корпоративні мережі та викрадати важливі дані. Їхня діяльність суттєво впливає на загальну динаміку кіберзлочинності у світі (рисунок 1.3), оскільки ці групи стають все більш професійними, краще організованими і мають доступ до новітніх технологій.

Найчастіше потрібно до одного тижня, щоб відновити дані. Це не залежить від методу їх повернення (викуп чи відновлення з резервної копії, якщо вона є). У середньому 40% організацій відновлюють дані за цей час, 30% організацій – до одного місяця, а 18% організацій потрібно 1-3 місяці. Середня вартість відновлення для великої організації без урахування викупів у 2021 році становила 1,85 млн доларів США, у 2022 році вона була 1,4 млн, а вартість у 2023 році становитиме біля 1,82 млн.

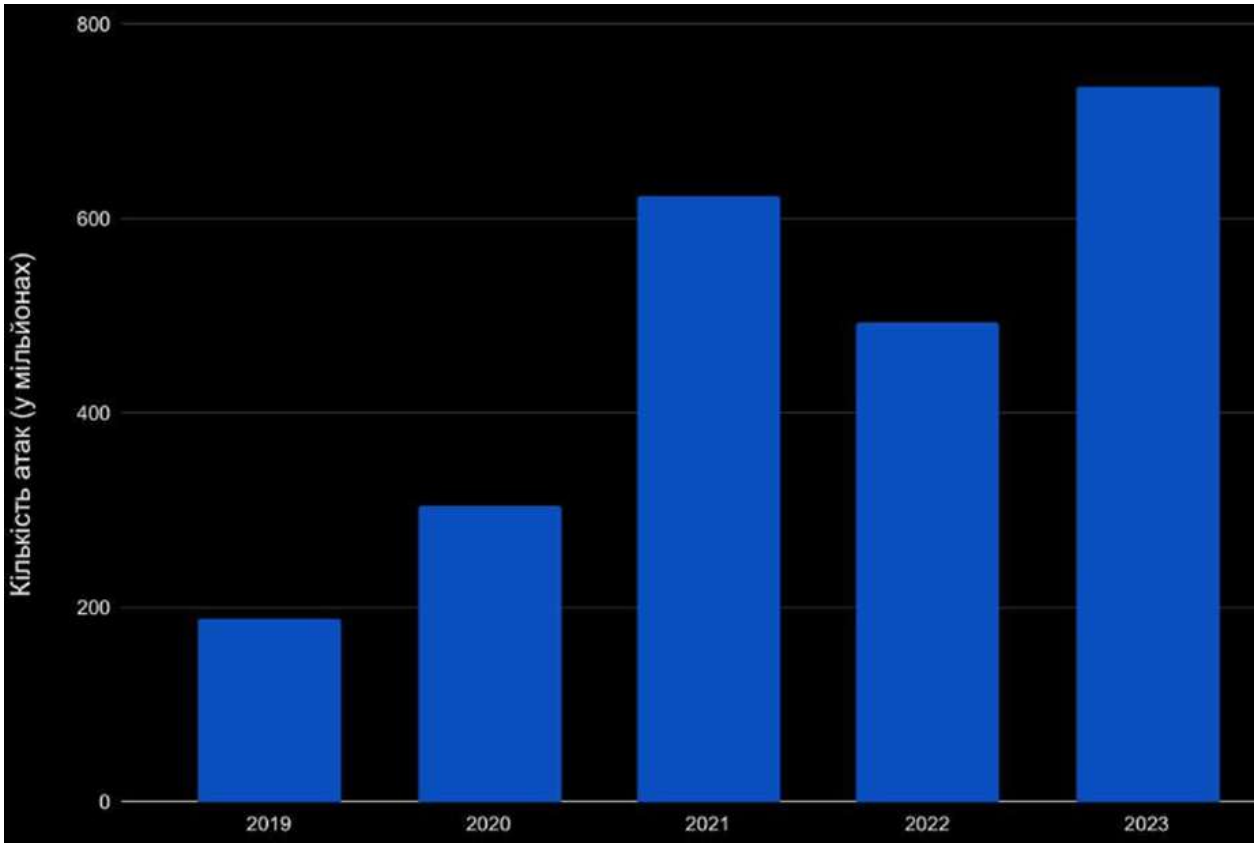

Рисунок 1.3 – Кількість атак програм-вимагачів за 2019-2023 роки в банківскій сфері

Ці суми показують вартість простою, у тому числі, заробітні плати, втрачений дохід та інші збитки.

Середня вартість збитків від одного витоку даних за період з 2020 по 2023 рік демонструє поступове зростання, що свідчить про збільшення масштабів і серйозності наслідків таких інцидентів (рисунок 1.4).

Зростання пов’язане з кількома факторами, зокрема зростанням цін на послуги відновлення даних, юридичними витратами на врегулювання позовів, а також витратами на забезпечення безпеки після витоку.

У 2020 році середня вартість витоку даних оцінювалась приблизно в 3,86 мільйона доларів США. Ця сума включала витрати на розслідування інциденту, відновлення систем та виплати штрафів, а також витрати на захист репутації компанії після публічного оголошення про витік.

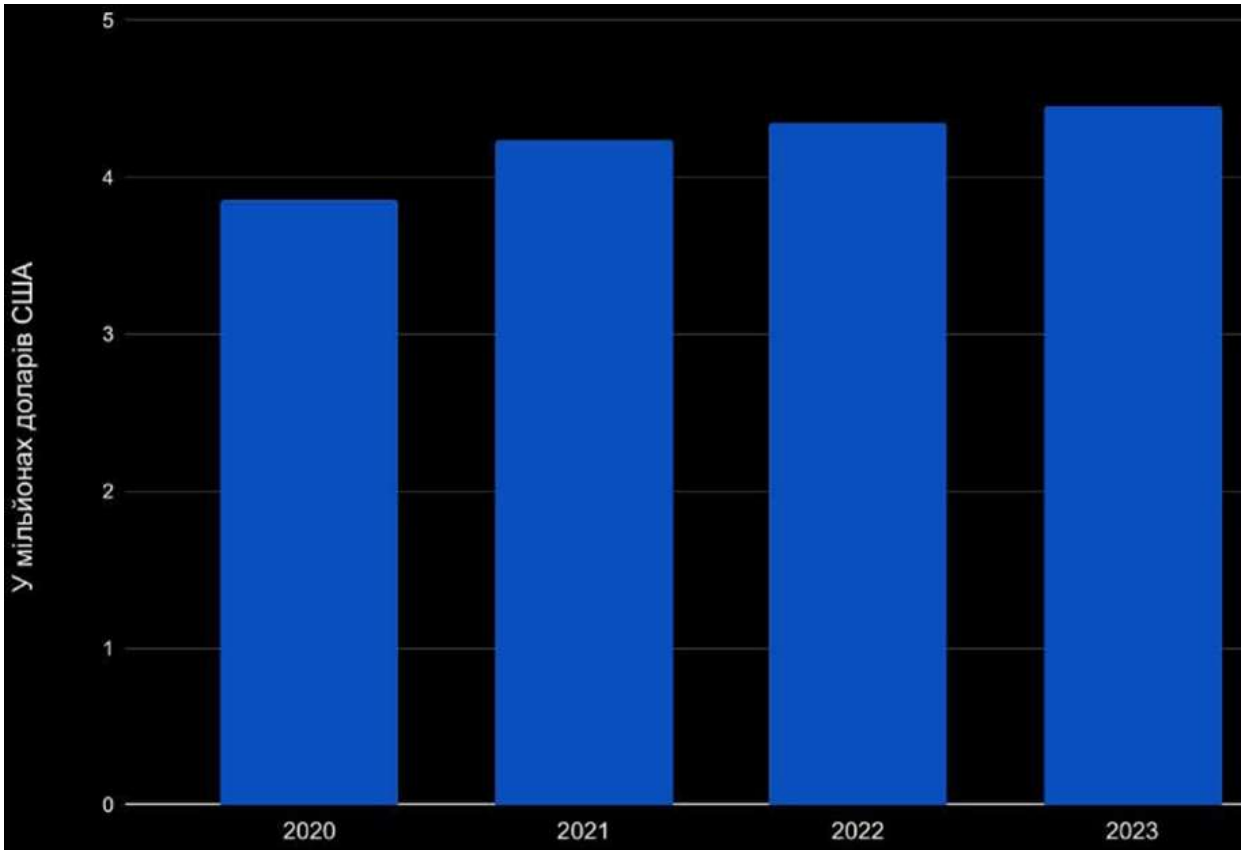

Рисунок 1.4 – Середня вартість збитків від одного витоку даних

Однак до 2023 року ця цифра значно зросла, і середня вартість витоку даних перевищила 4,45 мільйона доларів США. Основним фактором, який сприяв цьому зростанню, є те, що витік даних став серйознішим порушенням для компаній, а його наслідки можуть тривати довгий час.

Крім того, витрати на покращення кібербезпеки після витоку, включаючи оновлення програмного забезпечення, навчання персоналу та впровадження нових систем захисту, значно збільшують суму збитків. Враховуючи підвищену увагу до захисту персональних даних і введення нових регуляцій, таких як Загальний регламент захисту даних (GDPR), організації також стикаються з високими штрафами за невиконання вимог щодо збереження конфіденційності даних. Ще одним важливим аспектом є збільшення витрат на юридичні послуги та компенсації клієнтам, чиї дані були порушені. Витік конфіденційної інформації може призвести до численних позовів від постраждалих осіб або організацій, що ще більше ускладнює фінансові наслідки (рисунок 1.5).

За

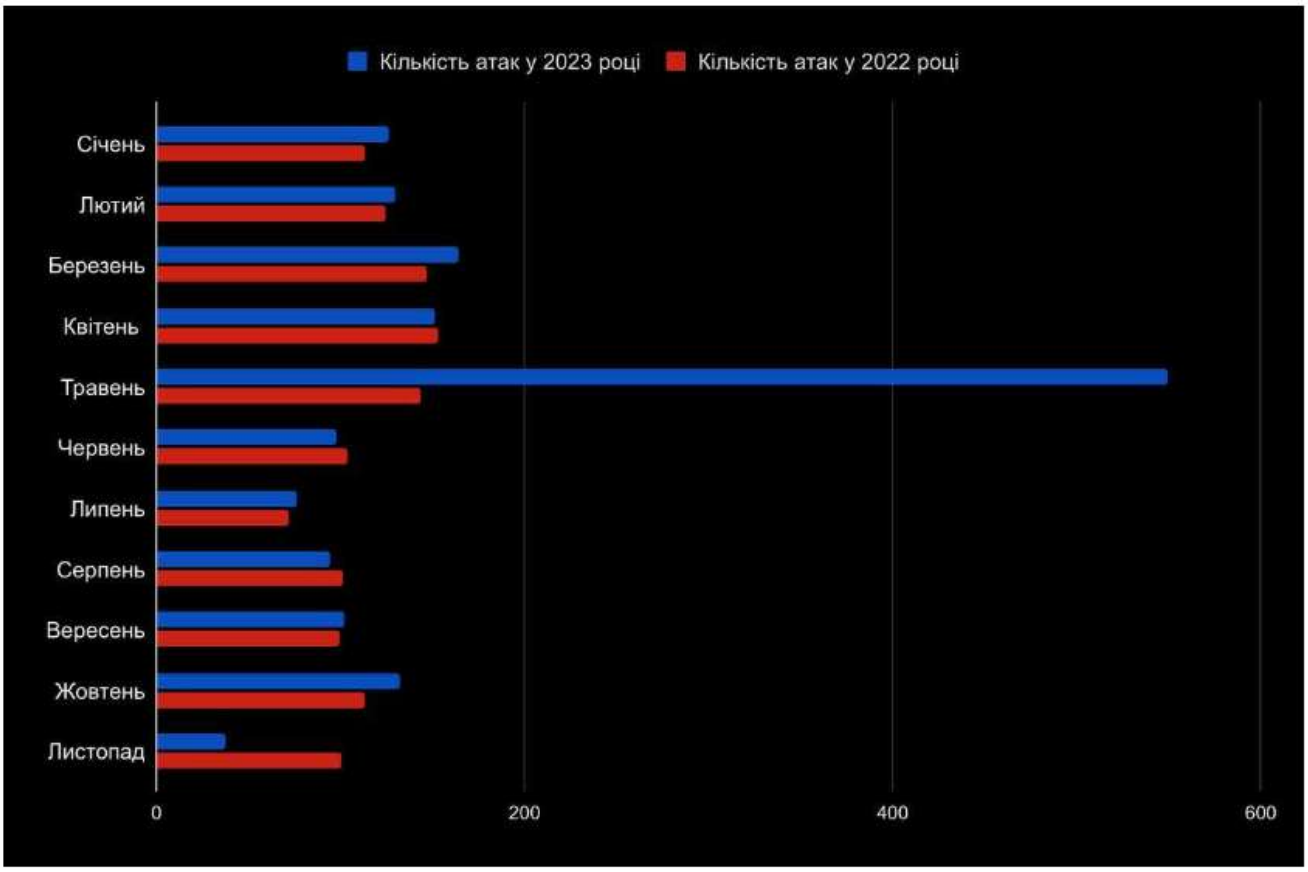

Рисунок 1.5 – Кількість атак із великими витоками даних 2022 та 2023 років

Кількість кібератак станом на кінець листопада 2023 року досягла 1659, що на 10% більше порівняно з 2022 роком. Особливо відзначаються травневі атаки з витоком даних, які спричинили різке збільшення показників, хоча загалом кількість атак зростає поступово. Ці статистичні дані ґрунтуються лише на відомих випадках витоків, оскільки багато компаній не розголошують таку інформацію.

Щодо фішингових атак, то їх кількість зростає з кожним роком. Це зумовлено кількома факторами: новими можливостями в IT-сфері, які перевищують здатність систем безпеки адаптуватися; інтересом кіберзлочинців не тільки до звичайних користувачів, але й до великих корпорацій; а також збільшенням кількості як індивідуальних злочинців, так і організованих груп. У 2023 році кількість великих фішингових атак значно зросла – лише за перше півріччя було зафіксовано більше таких атак, ніж за весь 2020 рік, і майже стільки ж, скільки за 2021 рік (рисунок 1.6).

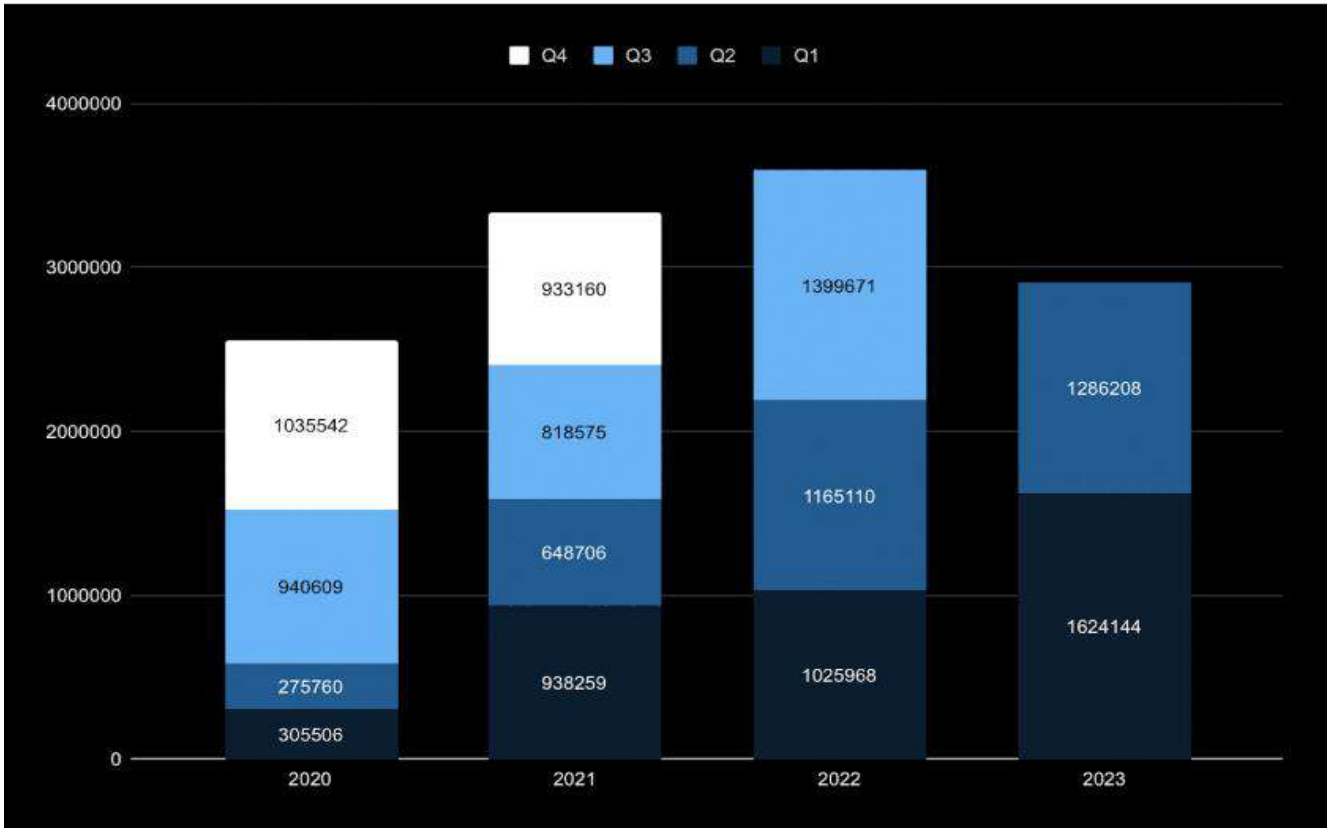

Рисунок 1.6 Кількість фішингових атак за квартали 2020-2023 років

Очікується, що фішингові атаки в наступні роки залишатимуться одними з найбільш поширених видів кіберзлочинів і їх кількість буде зростати, разом із кількістю кіберзлочинців. З 2021 року популярність набирає фішинг, спрямований на крадіжку фінансових коштів. Аналізуючи статистичні дані, можна зробити висновок, що фінансові установи, соціальні мережі, платформи SaaS та електронна пошта залишатимуться основними цілями фішинг атак.

Щоб ефективно протидіяти фішингу, по-перше, потрібно систематично підвищувати обізнаність у сфері кібербезпеки, а по-друге, важливо постійно вдосконалювати навички виявлення соціальної інженерії.

Віртуальний простір залишається місцем постійних викликів і можливостей. Тому кібератаки, здирництво та витоки даних набирають обертів, ставши проблемою як окремих осіб, так і великих організацій. Захист даних і безпека систем повинні бути пріоритетними для кожного з нас.

Кожній компанії потрібно дбати про інформаційну безпеку. Це звучить зрозуміло та очевидно. У той самий час, з урахуванням нашого досвіду, ми можемо стверджувати, що системи та дані багатьох організацій залишаються дуже вразливими.

1.4 Постановка задачі

Основною метою цього дослідження є створення методу і клієнтського алгоритму виявлення фішингу електронної пошти, який виявить і запобіжить співробітникам банку відповідати на фішингові електронні листи, таким чином допомагаючи банкам зменшити кількість фішингових атак електронною поштою.

Щоб реалізувати програму досліджень необхідно:

а) визначити найбільшу вразливість у банківській сфері за допомогою системного аналізу та визначити напрямки вдосконалення технології, що можуть бути використані в підвищені ефективності;

б) затвердити основні положення роботи клієнтського алгоритму виявлення фішингу електронної пошти в банківській сфері;

в) зробити алгоритмічну та математичну модель клієнтського алгоритму виявлення фішингу електронної пошти в банківській сфері;

г) здійснити алгоритмічну реалізацію технології;

д) довести покращення теоретичних і алгоритмічних рішень технології.

2 ТЕОРЕТИЧНЕ ОБГРУНТУВАННЯ ПОЛОЖЕНЬ МЕТОДУ, СИСТЕМНИЙ АНАЛІЗ І ПОРІВНЯННЯ ІЗ ІСНУЮЧИМИ АЛГОРИТМАМИ

2.1 Теоретичне обгрунтування методу захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи

При побудові методу будемо спиратись на забезпеченні таких його властивостей:

- децентралізована перевірка та мінімізація ризиків;
- ефективність обробки;
- захист конфіденційності і клієнто-орієнтований підхід;
- персоналізована аналітика;
- інтеграція з існуючими системами;
- використання новітніх технологій;
- усунута залежність від серверів;
- автоматичне оновлення.

Клієнтські алгоритми працюють безпосередньо на пристрої користувача, що зменшує залежність від серверного рішення. Це додатково підвищує рівень безпеки, оскільки клієнтські алгоритми продовжують виконувати свої функції локально, навіть якщо сервер банку піддається атаці.

Оскільки алгоритм працює безпосередньо на клієнтському пристрої, можливий аналіз електронних листів у режимі реального часу. Це значно зменшує затримки порівняно з централізованими рішеннями, які вимагають передачі даних назад на сервер.

Захист конфіденційності передбачає, що клієнто-орієнтований підхід не вимагає передачі листів та їхнього вмісту на сторонні сервери для аналізу. Це важливо в банківських системах, де конфіденційність даних є критично важливою. Такий підхід мінімізує ризик витоку інформації. Стійкість до мережевих атак , запуск алгоритмів локально на пристрої клієнта забезпечує незалежність від збоїв у мережі та атак на серверну інфраструктуру. Це підвищує надійність системи в цілому.

Персоналізована аналітика передбачає, що клієнтські алгоритми можуть бути налаштовані відповідно до індивідуальних характеристик користувача, таких як звички користування електронною поштою, типи отриманих листів та особливості банківських послуг. Це підвищує точність виявлення фішингу. Можливості клієнтського алгоритму.

Клієнтські алгоритми можуть бути інтегровані в програми, якими користуються клієнти банку (наприклад, мобільні додатки або поштові клієнти). Це забезпечує безперебійну роботу для користувача і не вимагає додаткових дій з його боку.

Для реалізації клієнтських алгоритмів, таких як аналіз лексичних і семантичних властивостей текстів, виявлення аномалій відправників і посилань, перевірка репутації домену в режимі реального часу, можуть використовуватися новітні методи машинного навчання і обробки текстів.

Алгоритм працює на клієнтському пристрої, але може регулярно оновлюватися за допомогою механізму автоматичного оновлення, щоб враховувати нові методи фішингових атак.

Серверні підходи вимагають постійної доступності серверів, що може створити додаткові ризики, якщо сервери недоступні або піддаються атаці. Обмеження в передачі даних передача великих обсягів листів на сервери для аналізу може бути обтяжливою для мережі та спричиняти затримки в обробці, особливо за високих навантажень потенційні вразливості. Централізовані сервери є більш привабливими цілями для зловмисників і підвищують ризик компрометації всієї системи безпеки.

З практичної точки зору, клієнтські алгоритми забезпечують зручність для кінцевих користувачів, оскільки не вимагають складних процедур налаштування. При отриманні фішингового листа алгоритм може автоматично попередити користувача, заблокувати небезпечні посилання або позначити лист як підозрілий. Крім того, впровадження клієнтських алгоритмів дозволяє банкам зміцнювати довіру клієнтів, демонструючи, що вони зацікавлені в безпеці своїх клієнтів.

2.2 Системний аналіз і порівняння можливостей алгоритмів протидії фішингу

Дослідження [12] присвячене колективній базі даних, яка використовує надійну хмарну платформу. Цей набір даних містив 18 366 позначених електронних листів, з яких 3 416 були фішинговими, а 14 950 – звичайними. Точність CNN становить 95,97%, а LSTM – 96,97%. Новизна цього дослідження полягає у використанні глибокого семантичного аналізу для виведення внутрішніх властивостей первинного тексту.

Дослідники в [14] додатково підкреслили цінність раннього виявлення небезпек за допомогою використання передових технологій. Запропонована нами методологія виявлення базується на машинному навчанні, дані були розділені на навчальний і тестовий набір. Функції включають тіло електронного листа, тему, відправника, інформацію про URL-адреси та класифікацію електронних листів як фішингові чи ні. Результати трьох моделей наборів даних показали, що моделі з найбільшими векторами ознак мають найбільший ступінь точності, точність моделі розширеного дерева рішень становила 0,88, 0,92 і 0,94 відповідно. Крім того, у статті обговорюються практичні контрзаходи та пропозиції щодо підвищення обізнаності користувачів, які залишаються важливими для боротьби з фішингом.

У дослідженні [15] автори запропонували метод розрізнення фішингової атаки на електронну пошту та інших атак, які використовують той самий тип алгоритму, наприклад SVM, Decision Tree та LSTM. Заявлена точність дослідження становить 92%-96%, і всі вибрані статистичні методи оцінюються в результатах дослідження.

В [16] запропонували методи вирішення проблеми фішингових атак різного розміру та всіх поширених повідомлень. Для аналізу даних використовувалися такі методи, як DT, RF, MLP, KNN, SEL, SVEL, FMPED і FMMPED. Повідомляється, що точність становить від 88,50% до 96,37%. Вони визнали необхідність вирішення проблем розбіжностей даних, на яких було зосереджено це дослідження, щоб боротися з фішингом.

У [19] автори super covenant вивчали загрозу фішингу та спаму за допомогою глибокого навчання та технології обробки природної мови. Навчаючи LSTM і MLP міченому набору даних повідомлень, вони випустили його. Це дослідження показало точність 92% для LSTM і 94% для MLP, що демонструє потенціал глибокого навчання для підвищення безпеки електронних листів.

Ще одна спроба [20], яка використовує GCN разом із NLP, – це спеціальна спроба виявлення фішингу в тілі електронної пошти. Модель має відносно точний метод виявлення з точністю 96%, використовуючи весь текст електронної пошти, згенерований набір даних, який автоматично змінюється 4%. Новизна запропонованої методології полягає в тому, що GCN використовується для класифікації тексту, який потім стає основою для практичної задачі розпізнавання фішингових листів. Однак основним недоліком дослідження є те, що воно проводилося англійською мовою, яка може бути незастосовною до інших мов чи електронної пошти.

Робота [21] полегшує використання моделі багаторівневого персептрона (MLP) із Spam Base, Spam Assassin і набором даних вебспаму UK.2011, які мають заявлену точність 96,9%, 94,21% і 95,6% відповідно.

Цей проєкт важливий через численні набори даних і функції, які використовуються в розслідуванні, тому оцінка виявлення спаму є більш комплексною. Однак основним недоліком дослідження є те, що воно зосереджено на сфері виявлення спаму, а не на специфіці фішингу, який є проблемною сферою з точки зору боротьби з фішинговими спам-листам.

Інша дослідницька стаття [22], в якій використовувалися мережі CNN і LSTM, порівнювала результати між Адамом і SGD на наборі даних, який включав 18365 електронних листів, з яких 3416 вважалися фішинговими; дослідження досягло точності 97,3% для LSTM і 96,52% для CNN.

Як оптимізатор цієї проблеми було обрано оптимізатор Adam, який виявився ефективнішим, ніж SGD, на основі дослідження, згаданого раніше в статті. Однак порівняння проводилося виключно для класифікації текстових даних для

практичних цілей, які можуть не враховувати інші важливі функції, такі як метадані та поведінка користувачів під час виявлення фішингу.

Для більш обґрунтованого вибору методів класифікації нижче наведено порівняльні дані (таблиця 2.1).

Наївний алгоритм Байєса – це класифікатор умовної ймовірності на основі теореми Байєса, яка описує ймовірність події на основі попередніх знань про умови, пов'язані з подією. Наприклад, якщо під час надходження фішингового листа присутній фішинговий електронний лист, вірогідність того, що певний електронний лист дійсно є фішинговим, можна точніше оцінити за допомогою конкретного ключового слова, ніж без нього. Формула така:

$$\mathrm{P(A\,|B)} = \mathrm{P(B\,|A)P(A)} \div \mathrm{P(B)}\,, \tag{2.1}$$

де A і B є події та $\mathrm{P(B)} \neq 0, \mathrm{P(A)}$ I $\mathrm{P(B)}$ є ймовірності спостереження A і B самостійості. $\mathrm{P(A|B)}$ представляє умовну ймовірність, яка є ймовірністю спостереження події A за умови, що B вірно.

$\mathrm{P(B\,|A)}$ представляє ймовірність спостереження за подією B за умови, що A вірно. Використання наївного алгоритму Байєса для задачі класифікації $\mathrm{x} = (\mathrm{x1}\,, \ldots \ldots, \mathrm{xn}\,)$ де n представляє кількість незалежних ознак, він призначає цьому екземпляру ймовірності $\mathrm{p(Ck\,|x1}\,, \ldots \ldots, \mathrm{xn})$ для кожного з K можливих результатів або класів Ck:

$$\mathrm{p(Ck\,|\,x)} = \mathrm{p(Ck\,)p(x\,|\,Ck)} \div \mathrm{p(x)}, \tag{2.2}$$

де постеріорна ймовірність $p(C|\,x)$ – це ймовірність цілі C (у нашому випадку це може бути ймовірність фішингового електронного листа) з урахуванням предиктора x (ключове слово, передане класифікатору); $p(C)$ – попередня ймовірність цілі; $p(x\,|\,C)$ – ймовірність, тобто ймовірність предиктора з урахуванням цілі; $p(x)$ – імовірність предиктора.

Таблиця 2.1 – Порівняльна характеристика методів і алгоритмів

Алгоритм	Принцип роботи	Переваги	Недоліки
Naïve Bayes	Базується на припущені про незалежність ознак, через що є обчислення простим, однак залежним від виконання цього припущення. Побудовано на теоремі Байеса	Може бути використаний для бінарної і мульти-класової класифікації. Добре працює з категоріальними ознаками. Показує хороші результати на маленьких та великих вибірках. Дуже швидко обчислюється.	Якщо не виконується припущення незалежності ознак якість класифікації дуже сильно знижується. Потребує додаткової обробки для значень ознак яких не було у тренувальній вибірці
Logistic Regression	Використовує лінійне рівняння із незалежними змінними для прогнозування	Результати прогнозування легко впроваджуються. Швидко обчислюється.	Важко моделювати залежност змінними, так як функція є лінійною
SVM	Будує функцію гіперплощини, яка є максимально віддаленою від	Добре працює при великій кількості ознак. Дуже добре працює у випадках	Погано реалізується. Обчислювально складний на великих даних.
Random Forest	Будує функцію гіперплощини, яка є максимально віддаленою від екземплярів обох класів.	Добре працює при великій кількості ознак. Дуже добре працює у випадках коли класи можна лінійно розділити. Ефективно використовує пам’ять.	Погано реалізується. Обчислювально складний на великих даних. Сильно залежить від функції вибору ядра. Погано працює у випадках коли класи перетинаються. Дуже чутливий до помилок розмітки.
Decision Tree	Будує класифікатор у вигляді дерева рішень. У кожній вершині дерева обирає межу яка досягає найбільшої «чистоти» класів у кожній із підвибірок. Будується рекурсивно	Майже не потребує попередньої обробки даних, відносно інших алгоритмів. Не потребує нормалізації чи стандартизації даних. Пропущені дані не впливають на тренування класифікатора.	Обчислювальна складність досить швидко росте із збільшенням ознак і вибірок. Алгоритм дуже волатильний. маленькі зміни у тренувальних даних можуть досить сильно змінити вихідну модель.

Використовуючи теорему Байєса ми отримуємо:

$$\text{posterior} = \text{prior} \times \text{likelihood} \div \text{evidence}. \quad (2.3)$$

На рисунку 2.1 наведено приклади застосування формул для конкретних умов.

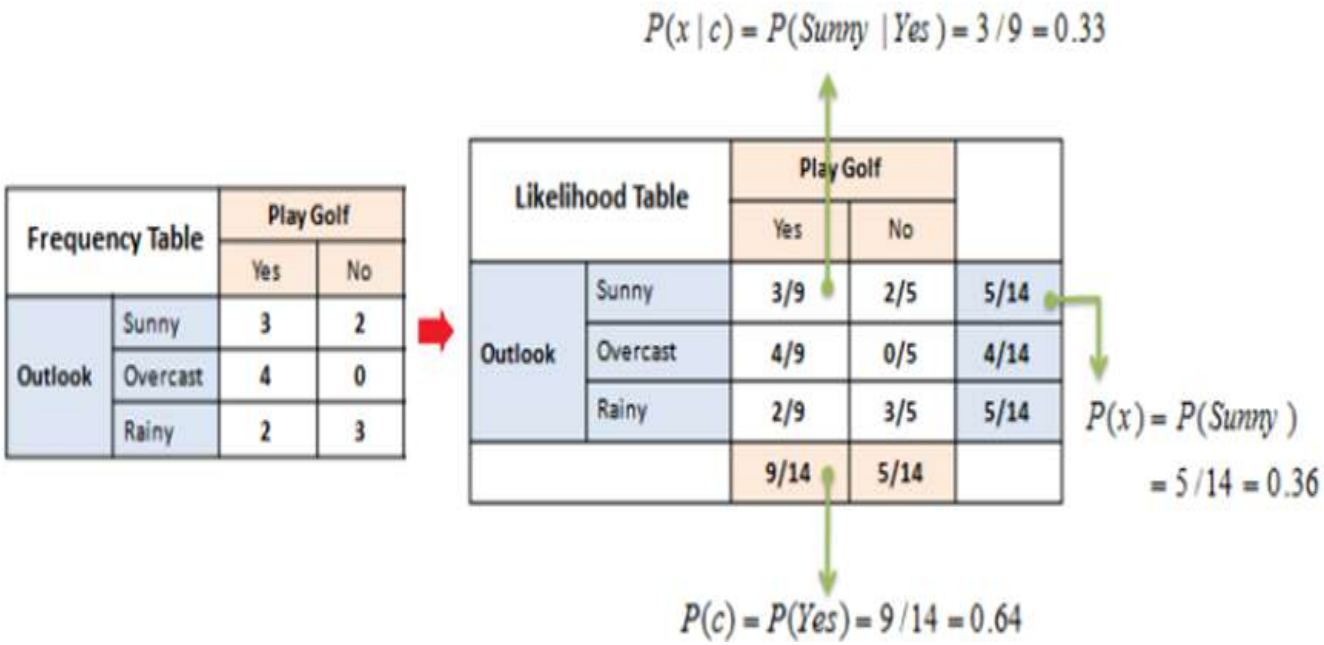

Рисунок 2.1 Розрахунки для конкретних даних (приклад)

Спочатку ми обчислюємо апостеріорну ймовірність, будуючи частотну таблицю для кожного атрибута цілі. По-друге, ми перетворюємо частотну таблицю в таблицю ймовірностей і, нарешті, обчислюємо апостеріорну ймовірність кожної категорії за допомогою простого рівняння Байєса:

$$p(x|c) = P(sunny|yes) = 3/9 = 0.33;$$

$$(c) = P(Yes) = \frac{9}{14} = 0.64;$$

$$P(x) = P(Sunny) = \frac{5}{14} = 0.36;$$

$$P(c|x) = P(Yes|Sunny) = \frac{0.33*0.64}{0.36} = 0.60.$$

Алгоритм логістичної регресії використовується для вирішення проблем двійкової класифікації шляхом оцінки ймовірності того, що вебсайт є безпечним або шкідливим. гібридна вибірка – це гібрид надмірної та недостатньої вибірки, який призначений для підвищення узагальненості моделі та зменшення ймовірності перепідбору. Модель оцінюється, щоб визначити ступінь точності моделі логістичної регресії.

Використовуються також оціночні вимірювання на основі перехресної перевірки. Перехресна перевірка – це метод прогнозування середньої похибки моделі, коли вона оцінюється за ненавченим набором даних, що забезпечує точне оцінювання моделі (рисунок 2.2).

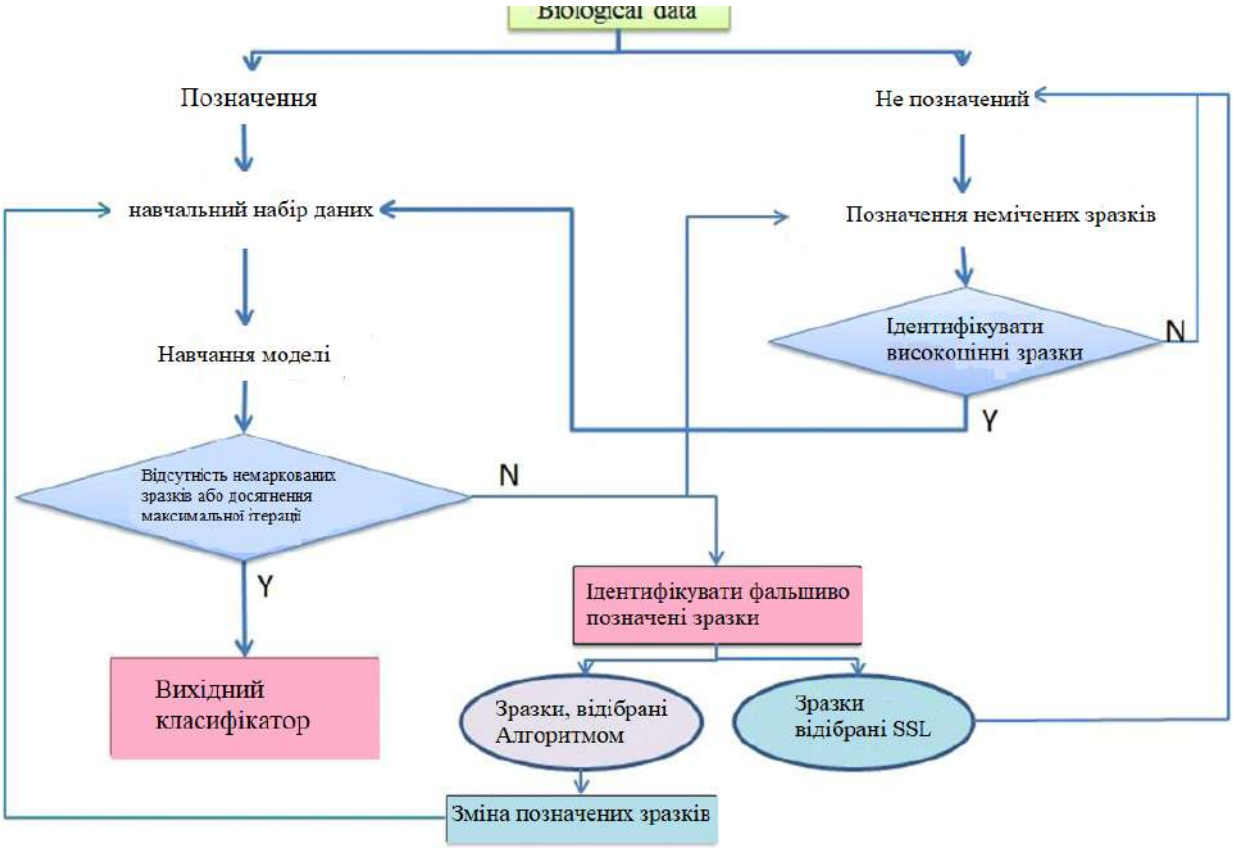

Рисунок 2.2 – Блок схема і принцип роботи Logistic Regression

Обговоримо методи для розрізнення фішингових вебсайтів.

K-Nearest Neighbors (KNN) – це успішний метод неконтрольованого навчання, ефективний для багатьох додатків, у тому числі для методів безпеки. K-

найближчий сусід виводиться з концепції кластеризації. Однією з переваг SVM є те, що його можна використовувати для навчання систем із кількома вимірами даних. SVM – це машина, яка отримує позначені навчальні дані як вхідні дані та створює оптимальну гіперплощину, яка класифікує нові приклади. SVM використовує гіперплощину, яка максимізує запас між наборами даних, щоб це було показано (рисунок 2.3).

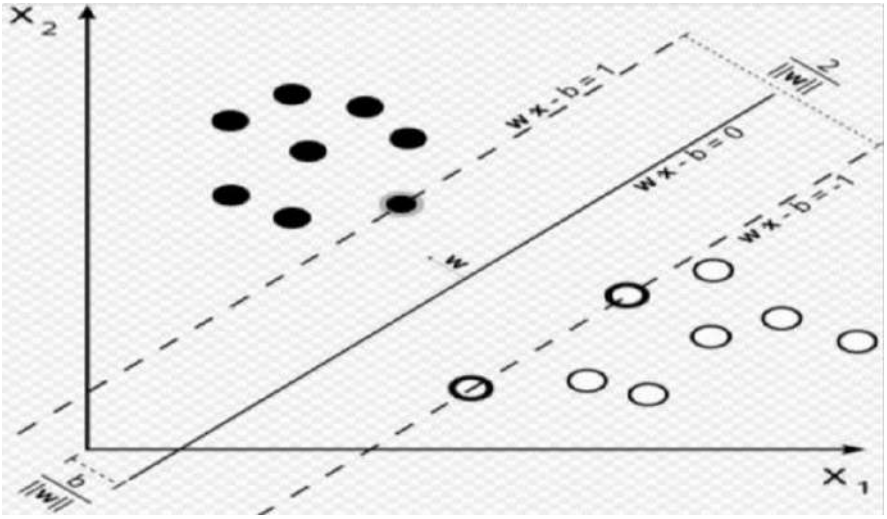

Рисунок 2.3 – Схема алгоритму SVM для класифікації фішингових сайтів

Однакові риси він класифікує тестовий випадок як A або B на основі k-сусіда, який з ним суміжний. Значення k у KNN залежить від розміру набору даних, а тип проблеми класифікації 1 демонструє, що KNN класифікує ціль на основі сусідства з її класом. KNN описується наступним чином: знайдіть найближчі елементи до тестових даних a у навчальних даних K на основі евклідової відстані, щоб обчислити відстань. Для двох елементів у k-вимірному просторі:

$$a = [a1, a2, \dots, ak] \text{ і } y = [\, b1, b2, \dots, bk]. \quad (2.4)$$

Евклідова відстань на основі двох елементів може бути обчислена:

$$\text{d(a,b)} = \sqrt{\sum_{i=1}^{k} b_i - a_i)\sqrt{2}}. \quad (2.5)$$

Після збору k-найближчих сусідів більшість k-найближчі сусіди будуть розглядатися як тестові дані.

SVM – це метод машинного навчання, який спирається на нагляд і підходить як для регресії, так і для класифікації. SVM вважається сучасною еволюцією методу, який набув значної популярності завдяки успішним результатам у багатьох сферах інтелектуального аналізу даних, що базується на потужній основі теорії статистичного навчання. SVM є методом класифікації, який спирається на статистичне навчання та успішно використовується в численних нелінійних класифікаціях і великих наборах даних. Класифікатори SVM використовують гіперплощину для розрізнення категорій. Кожна гіперплощина характеризується прямою лінією (w), певним положенням у просторі або програмою (b), (xi) є вхідним масивом компонента N і позначає тип:

$$(x_1,y_1)(x_1,y_2),\ldots,(x_k,y_k);x_i \epsilon R^d \tag{2.6}$$

K представляє номер навчального набору даних, а d позначає кількість вимірів вхідного набору даних. Функція рішення визначається наступним чином:

$$\int(x,w,b) = sgn\big((w.xi)+b\big), w\epsilon R^d, b\ \epsilon\ R \tag{2.7}$$

Дерево рішень – це алгоритм машинного, дерева котрий використовується для розрізнення та прогнозування проблем такого типу. Він розроблений як ієрархічна структура прийняття рішень, кожна гілка представляє умову на основі одного атрибута даних, а результати є самими умовами. Листя на дереві символізують остаточне судження (добре чи погано). Створення дерева розподілу даних алгоритм вибирає атрибут, який буде розділяти дані на основі метрики (наприклад, отримання інформації або критерію Джіні). Атрибут, який забезпечує найбільший ступінь поділу, вибирається як атрибут для кореневого вузла. Умови розгалуження, на кожному вузлі встановлюється умова, яка розрізняє дані на

основі значення атрибута (наприклад, "x > 5"). Рекурсивна ітерація процедура повторюється для кожної гілки дерева, доки всі точки даних у вузлі не будуть одного типу. Подальше розділяти дані немає сенсу (дерево перевищило максимальну глибину або мінімальний розмір даних).

Для нового набору даних алгоритм спочатку відвідує корінь, а потім вибирає гілку на основі умови в кожному вузлі. Цей процестриває до тих пір, поки не буде досягнутий лист, що містить прогнозоване значення або клас. Точність моделі оцінюється за допомогою таких метрик, як точність, повнота і F-міра. Застосування в задачах класифікації - Дерева рішень часто використовуються в кібербезпеці для класифікації шкідливих посилань. Наприклад, вхідні дані, атрибути URL-адреси (наприклад, довжина, наявність/відсутність цифр, підозрілий домен), вихідний клас («шкідливий» або «безпечний»). Як частина ансамблю випадкових лісів об'єднання декількох дерев рішень, кожне з яких навчалося на різних підмножинах даних, для підвищення надійності та точності. Градієнтний бустінг використання дерев для поступового покращення прогнозів. Індекс Джині використовується для вимірювання «нездоров'я» вузла. Нижче значення індексу Джині вказує на більш однорідний розподіл даних:

$$Gini = 1 - \sum_{i=1}^{n} p_i^2, \tag{2.8}$$

де pi – частка елементів, що належать до класу i у вузлі, n – кількість класів.

Інформаційна вигода (Information Gain) Цей критерій використовується для вибору атрибуту, який дає найбільшу користь для розділення даних:

$$\mathrm{IG(D, A)} = \mathrm{Entropy(D)} - \sum_{\mathrm{v \in Values(A)}} \frac{D_v}{D} \cdot \mathrm{Entropy(Dv)} \tag{2.9}$$

де D – поточний набір даних, A – атрибут для поділу; Dv – підмножина D, де атрибут A має значення v; Entropy – ентропія.

Ентропія (Entropy) вимірює рівень невизначеності або хаосу у вузлі:

$$\text{Entropy(D)} = -\text{i} = \sum_{i=1}^{n} p_i \quad \cdot \log 2(\text{p}_i) \tag{2.10}$$

де pi – частка елементів класу i у вузлі, n – кількість класів.

Результати порівняння описаних алгоритмів відображує рисунок 2.4.

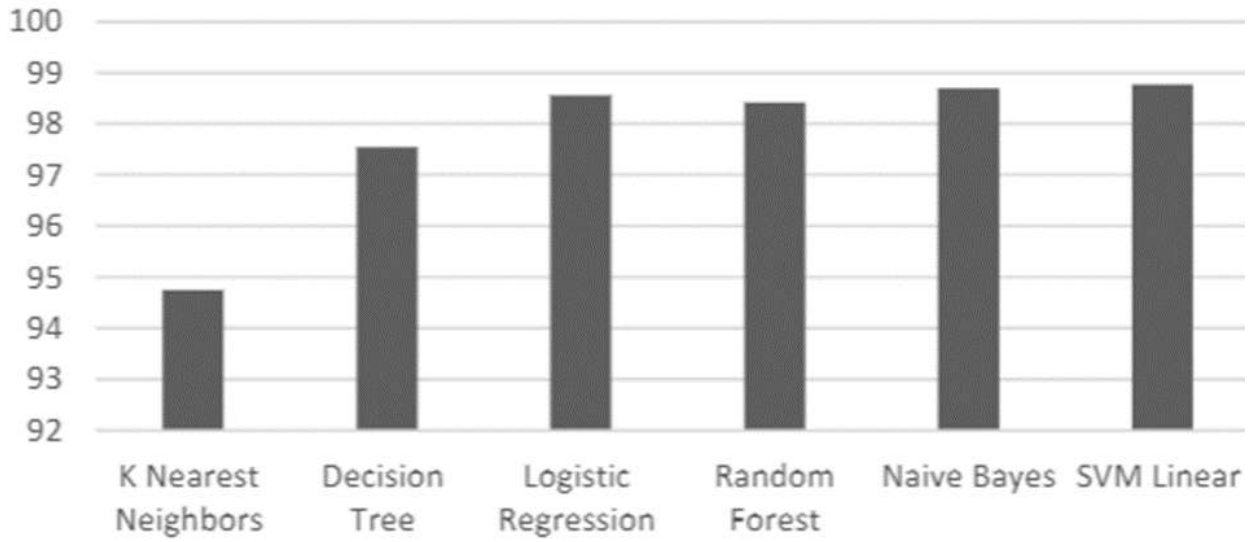

Рисунок 2.4 –Порівняння алгоритмів на точність

2.3 Теоретичне обґрунтування алгоритму реалізації методу

Метою цього дослідження є створення інтелектуальної антифішингової стратегії для банківської діяльності. Щоб змінити ефективний алгоритм, попередньо використовувані алгоритми ретельно перевіряються, щоб визначити основні вимоги дослідження. У цьому дослідженні використовувалася кількісна методологія. Це полегшило процес навчання завдяки використанню простої теореми Байєса для створення класифікатора та його регулярного оновлення з метою підвищення точності алгоритму. У цьому розділі описано вимоги до даних і аналіз, функціональні та нефункціональні вимоги, а також опис алгоритму.

Щоб зробити алгоритм ефективним для зупинки та запобігання фішингу, необхідно розробити структуру, яка враховує шкідливі методи, присутні у

фішингових електронних листах. У результаті виконується аналіз вимог, який визначає необхідні ресурси, які задовільнять потреби та вимоги щодо цілей. У результаті системні вимоги є значними, оскільки вони дозволяють запропонованій системі усунути існуючі недоліки. Це також полегшує керування кінцевим рішенням, а також гарантує, що кінцевий продукт відповідає gjnht,fv організації. Однак системні вимоги потребують, щоб кожна вимога була розкладена та чітко вказана. Далі хід усіх подій оцінюється відповідно до взаємодії між кожною вимогою, щоб визначити, чи потрібна вимога чи ні. Аналіз системних вимог полегшує створення структури розробки, яка є важливою для всіх майбутніх починань.

Функціональні та нефункціональні вимоги – це два типи системних вимог. Функціональні вимоги описують заплановану функцію системи та необхідні етапи або процеси. Це передбачувана функціональність, яку клієнт хоче отримати від запропонованої системи. Функціональна вимога також описує зв’язок між системою та її оточення.

Функціональні вимоги запропонованого алгоритму наступні:

– алгоритм має звертатися до облікового запису електронної пошти користувача часто за встановленим графіком щоб отримати всі нові електронні листи;

– вилучення даних електронної пошти для кожного нового електронного листа;

– виконання антифішингову перевірку електронної пошти;

– зберігання деталей фішингової електронної пошти;

– позначення виявленого фішингового листа у вхідній пошті користувача, перемістивши його до папки спаму.

Нефункціональні вимоги запропонованого алгоритму такі:

– алгоритм повинен бути максимально точним для підвищення надійності;

– алгоритм повинен бути завжди доступний, як клієнтський додаток. Це гарантує, що він завжди перевіряє обліковий запис електронної пошти користувача, перевіряючи будь-яку нову електронну пошту після отримання;

– для забезпечення ефективності алгоритму клієнт повинен мати підключення до інтернету для його виконання;

– алгоритм не повинен потребувати багато обслуговування;

– алгоритм має займати дуже мало часу, щоб виконати аналіз, враховуючи середню кількість електронних листів, отриманих клієнтом;

– алгоритм має бути точним, доступним, але в той час не заважати користувачам електронної платформи, щоб підвищити рівень довіри користувачів;

– алгоритм повинен мати можливість виконувати та досягати своїх вимог на основі різноманітності реалізованих там перевірок;

– алгоритм повинен забезпечувати безпеку електронної пошти користувача. Треба тільки знаходити , виявляти , видаляти фішингові листи;

– з точки зору користувача, алгоритм повинен працювати бездоганно. Користувачі не повинні мати жодного уявлення про проведення антифішингового аналізу.

Алгоритм є клієнтським, на відміну від сканерів електронної пошти, які виконують роль поштового сервера. Використовувана база даних – портативний SQLite, у якому відсутня централізована база даних. Портативна база даних здатна лише зберігати інформацію про кожну проаналізовану електронну пошту, але не зберігає електронні листи, оскільки вони фактично зберігаються в обліковому записі електронної пошти користувача. По-перше, алгоритм перевіряє, чи має клієнт аутентифікаційний електронний лист за допомогою протоколу IMAP. Цей вхід є відокремленим від фактичного входу користувача. Тут використовується алгоритм, який є частиною облікового запису користувача. Після успішної аутентифікації він перевіряє скриньку електронної пошти на наявність нових повідомлень і витягує необроблений текст із електронного листа.

Алгоритм містить набір слів-приманок, які використовуються для навчання класифікатора. Ці слова включають на прикладі: клацніть, тут, вкладення, посилання, агент, pdf та zip серед інших, які зазвичай з’являються з основною частиною багатьох банківських фішингових електронних листів. Цей набір слів

утворює категорію класу phish ($\boldsymbol{C}$) і використовуються для обчислення попередньої ймовірності передбачення $p(x)$. Це робиться для кожного терміну в навчальний набір даних $x = (x1, \ldots .., xn)$, де n представляє кількість незалежних ознак у навчальний набір даних.

Бібліотека Python Imaplib використовує алгоритм для отримання частини звичайного тексту електронної пошти, яка містить посилання та написаний текст в електронному листі. Він використовує регулярні вирази для деконструювання пробілів, закодованих у протоколі, що використовується для IMAP. Алгоритм криптографічно кодує цю частину (написаний текст і посилання) у Base64 і зберігає її в базі даних.

Регулярні вирази відрізняють усі згадки письмового тексту від тексту, на який посилаються. Далі алгоритм вибирає написаний текст, маркує його та видаляє всі непотрібні символи, залишаючи лише слова в літері. Потім він видаляє всі стоп-слова з цього кластеру слів, а потім передає решту слів на машину Naive Bayes для обчислення ймовірності фішингового слова серед них.

У наївному класифікаторі Байєса він обчислює ймовірність кожного слова: $p(x \mid C)$ і множить його на ймовірність того, що він належить до класу $p(C)$, щоб отримати $p(C)p(x \mid C)$. Тоді ймовірність того, що електронний лист є фішинговим, визначається як загальна сума $p(C)p(x \mid C)$ поділений на попередній вираз попередньої ймовірності $p(x)$ навчального набору даних. Алгоритм обчислює журнал ймовірності кожного члена та додає їх, щоб отримати негативне значення фіш-оцінка. Це пов’язано з тим, що журнал усіх чисел від нуля до одиниці є від’ємним значення.

Далі алгоритм проаналізує всі посилання в тілі листа. Аналіз посилань здійснюється шляхом сканування кожного отриманого посилання на наявність вірусів за допомогою доступного Virus Total API. API дозволяє отримувати звіти про сканування без використання вебінтерфейсу HTML. Virus Total є дочірнею компанією Google, він оснащений великим спектром програмного забезпеченням а саме хробаками, вірусами, троянами та іншими шкідливими аналізаторами

вмісту, також який має величезний чорний список і широкий спектр сканерів, що запобігає хибним спрацьовування, які можуть мати місце при використанні одного або кількох сканерів .

Потім алгоритм перевіряє вкладення електронної пошти. Тут вибирається лише розширення з імені файлу та перевіряється, чи є воно загрозою для комп'ютера, чи є це зловмисним розширенням, наприклад .exe, чи типом MIME, який вважається загрозою, наприклад .ogg або .avi. Якщо вкладення буде визначено як шкідливе, це посилить репутацію цього електронного листа. Алгоритм не приймає жодних вкладень, він лише отримує ім'я файлу вкладення з частини розміщення вмісту тіла кожного електронного листа відповідно до протоколу RFC822.

Для кожного аналізованого сценарію оцінка репутації підсумовується за всіма виконаними перевірками. Коли електронний лист класифікується як такий, що має негативний NPL (Naive Bayesian Classifier), алгоритм розпізнає, що це електронний лист має на меті змусити людей подумати, що за ними стежать. Потім це дає електронному листу оцінку репутації -5. Однак репутація шкідливих посилань становить 15, а репутація додатків -15, щоб підвищити ефективність алгоритму. Фішинговий електронний лист має містити шкідливе вкладення або посилання, щоб вважатися шкідливим. Алгоритм вважає всі електронні листи загрозами з -15, що призводить до того, що вони є екстремальними. Крім того, усі електронні листи з однаковим вмістом (повна копія), надіслані різним клієнтам, матимуть однаковий рейтинг.

Алгоритм покладається на цей аналіз, щоб визначити найшкідливішу електронну пошту в папці "Вхідні" користувача, потім ця електронна пошта вважається зловмисною та переміщується до папки спаму. Алгоритм буде постійно шукати нові електронні листи, аналізуватиме ці листи та позначатиме будь-які, які він виявить, як підозрілі. Коли зловмисник повторює шкідливий електронний лист, алгоритм дізнається про це, оскільки хеш уже існує в базі даних. Цей метод використовується, коли аналітику потрібно позначити суб'єкт без повторного проведення всього аналізу. Це можливо, лише якщо вміст електронного листа не

змінено. У результаті клієнт усе ще отримує та лише видаляє електронні листи за призначенням.

Діаграма варіантів використання ілюструє поведінку та набір дій, у яких має брати участь пропонований алгоритм, коли він взаємодіє з одним або кількома зовнішніми зацікавленими сторонами системи. Використання – це загальна кількість дій, виконаних системою. Тут фокусом є система, а зовнішні користувачі є об’єктом. На рисунку 2.5 показано діаграму використання алгоритму захисту від фішингу.

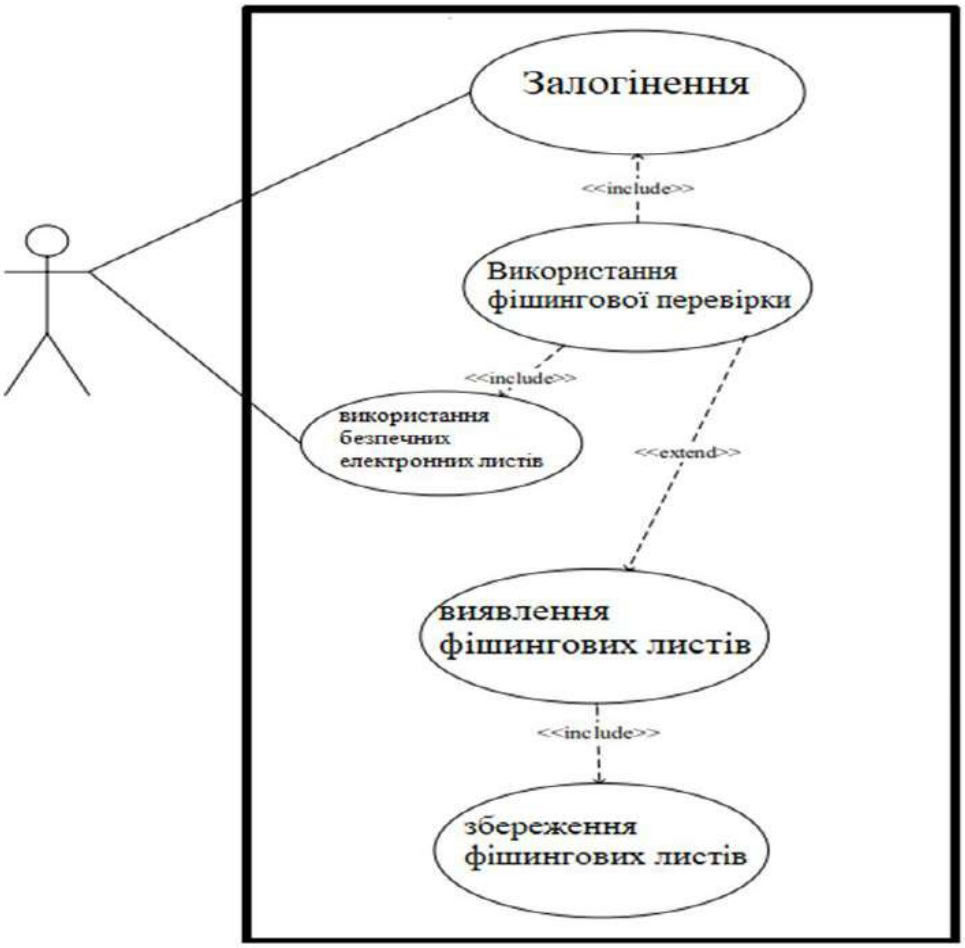

Рисунок 2.5 – Алгоритм виявлення електронної пошти на основі клієнта

В таблицях 2.2-2.6 показано зв’язки між даними що зображені на рисунку.

Таблиця 2.2 – Опис випадку залогінення

Характеристика	Розкриття змісту
ID	1

Дія	Надсилання облікових даних для входу
Тип випадку	Огляд, основний
Ініціатор	Користувач електронної пошти
Зв’язки	Користувач електронної пошти
Рівень важливості	Високий
Опис	Цей варіант використання вказує на те, що алгоритм вимагатиме від користувача надати облікові дані електронної пошти для входу щоб аутентифікуватися на поштовому сервері
Тригер	Алгоритм повинен бути налаштований на вхід і постійну перевірку нових електронних листів
Коректна робота алгоритму	Алгоритм пропонує користувачеві ввести ім’я користувача та пароль свого облікового запису електронної пошти. Користувач надає облікові дані для доступу до облікового запису електронної пошти. Алгоритм постійно перевіряє наявність нових електронних листів і, якщо їх знаходить, завантажує їх, готовий до аналізу.
Результат	Користувач увімкнув аутентифікацію IMAP для свого облікового запису електронної пошти.

Таблиця 2.3 – Опис випадку фішингової перевірки електронної пошти

Характеристика	Розкриття змісту
ID	2
Дія	Перевірка електронної пошти на фішинг
Тип випадку	Детальний , необхідний
Ініціатор	Алгоритм
Зв’язки	Взаємодія з ID 2 => ID 1
Рівень важливості	Високий
Опис	Цей варіант використання описує, як алгоритм виконуватиме перевірку фішингової електронної пошти ,на новому завантаженні акаунті
Тригер	Після перевірки наявності нової електронної пошти
Коректна робота алгоритму	Отримання інформації про будь-яку нову непрочитану пошту. Перевіряє, чи хеш-значення електронної пошти (хеш base64) уже існує в базі даних під загрозою електронні листи. Якщо це так, цей конкретний електронний лист буде переміщено з папки вхідних повідомлень електронної пошти користувача до папки спаму. Використовуючи наївний класифікатор Байєса, алгоритм виконує аналіз слів і дає оцінку; витягує інформацію про посилання та обчислює їх репутацію на основі відгуків, отриманих із чорних списків; перевіряє наявність зловмисних вкладень і додає інформацію до фішингової репутації. Обчислює підсумковий показник фішингу та надає репутацію електронної пошти як чисту чи загрозливу.
Результат	Алгоритм виконує антифішинговий аналіз лише нових електронних листів у точці аутентифікація.

Таблиця 2.4 – Опис випадку отримання безпечних електронних листів

Характеристика	Розкриття змісту
ID	3

Дія	Отримання безпечних електронних листів
Тип випадку	Детальний , необхідний
Ініціатор	Користувач електронної пошти
Зв’язки	ID 2 => ID 3
Рівень важливості	Високий
Опис	Цей варіант використання вказує на те, що користувач електронної пошти після аутентифікації у своєму обліковому записі електронної пошти бачитиме лише чисту пошту в папці "Вхідні".
Тригер	Користувач входить у свій обліковий запис електронної пошти
Коректна робота алгоритму	Електронна скринька користувача не містить фішингових листів. Користувач читає електронні листи як зазвичай.
Результат	Алгоритм не виявляє фішингу.

Таблиця 2.5 – Опис випадку виявлення фішингового листа

Характеристика	Розкриття змісту
ID	4
Дія	Виявлення фішингового листа
Тип випадку	Детальний, необхідний
Ініціатор	Алгоритм
Зв’язки	ID 1 => ID 2 => ID 4
Рівень важливості	Високий
Опис	Цей варіант використання вказує на те, що алгоритм позначає будь-який отриманий фішинговий електронний лист після виконання антифішингового аналізу
Тригер	Успішний аналіз вказує на те, що електронний лист справді є фішинговим.
Коректна робота алгоритму	Отримання ідентифікатору фішингової електронної пошти з бази даних у листах із загрозами; Звернення до конкретного електронного листа в папці «Вхідні» користувача, позначення його як фішинг та переміщення в папку папка спаму.
Результат	Виявлений фішинговий лист переміщено в папку спаму.

Таблиця 2.6 – Опис випадку збереження Хешу фішингового листа

Характеристика	Розкриття змісту
1	2
ID	5
Дія	Збереження фішингового електронного листа
Тип випадку	Детальний, необхідний
Ініціатор	Алгоритм
Зв’язки	ID 2 => ID 4 => ID 5
Рівень важливості	Високий

Кінець таблиці 2.6

1	2

Опис	Цей варіант використання вказує на те, що алгоритм зберігає хеш-значення будь-якого отриманого фішингового електронного листа після виконання антифішингового аналізу.
Тригер	Успішний аналіз вказує на те, що електронний лист справді є фішинговим.
Коректна робота алгоритму	Збереження повної інформації про фішингову електронну пошту в базі даних, включаючи хеш-значення електронної пошти. Хеш-значення має бути унікальним і, таким чином, допоможе ідентифікувати той самий фішинговий електронний лист, коли він буде відправленим вдруге
Результат	Збережений Хеш фішингового листа

На рисунку 2.6.зображено загальну схему роботи алгоритму.

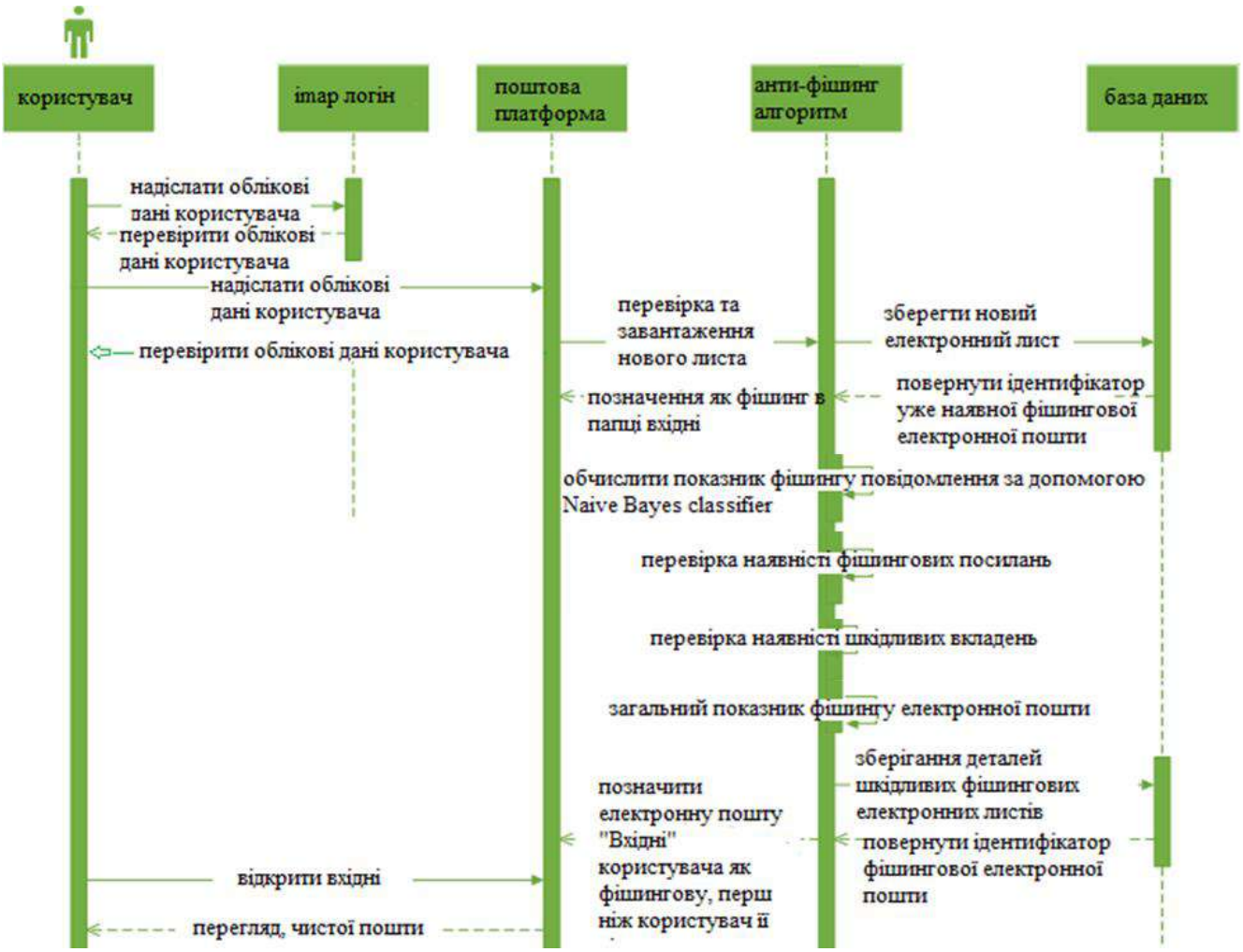

Рисунок 2.6 – Схема послідовності взаємодії між об’єктами під час роботи алгоритму антифішингу

2.4 Висновки

У розділі проведено детальний системний аналіз клієнтського алгоритму виявлення фішингових електронних листів, розглянуто передові розробки у цій сфері, а також викладено теоретичні вимоги до створення нового алгоритму. Основна мета полягала у визначенні переваг і недоліків існуючих методів, а також в розробці вдосконаленого рішення, яке відповідає сучасним функціональним і нефункціональним вимогам.

Запропонований алгоритм виконує ряд ключових функцій, спрямованих на ефективну ідентифікацію та обробку фішингових листів зокрема:

– часте звернення до облікового запису електронної пошти користувача для перевірки нових листів забезпечує постійний моніторинг вхідної пошти.

– автоматичне вилучення даних кожного нового листа та проведення антифішингової перевірки дозволяє уникнути ручної роботи з аналізу.

– зберігання деталей фішингових електронних листів забезпечує базу для майбутнього аналізу та покращення алгоритму.

– позначення виявлених фішингових листів і переміщення їх до папки спаму гарантує, що користувач не взаємодіє з потенційно небезпечним контентом.

Нефункціональні вимоги, такі як висока точність, доступність, низький рівень втручання в роботу користувача та мінімальне споживання ресурсів, визначають основні параметри успішної реалізації алгоритму. Недоліками поточного методу є вимога постійного підключення до Інтернету, можливість хибно–позитивних результатів і обмеження, пов’язані з великою кількістю вхідних даних.

Порівняльний аналіз показує, що алгоритм підходить для класифікаційних ознак, показує хороші результати як на малих, так і на великих вибірках, дуже швидкий в обчисленнях, має хорошу точність, швидко інтегрується в систему, а модель має можливість перенавчання та є легко налаштувати та масштабувати.

3 ТЕСТУВАННЯ ЗАПРОПОНОВАНОГО АЛГОРИТМУ

3.1 Підготовка тестового середовища

Для початку потрібно створити тестовий акаунт. Першим кроком є підготовка тестового облікового запису для збору електронних листів, за допомогою якого ми перевіримо ефективність алгоритму. Для цього вам потрібно створити кілька тестових облікових записів на популярних платформах електронної пошти або використати наявний тестовий обліковий запис, якщо він є.

Платформи для створення облікових записів можуть включати такі популярні служби, як Gmail, Yahoo, Outlook, а також платформи, які підтримують тестові облікові записи, такі як Mailtrap або Mailinator. Щоб перевірити, наскільки добре алгоритм працює на різних поштових системах та їхніх функціях, важливо мати кілька облікових записів на різних платформах. Щоб полегшити подальше тестування, слід вибрати принаймні три різні платформи з активними обліковими записами, що дозволить перевірити алгоритм на різних типах повідомлень і варіантах фішингу. Для кожного облікового запису потрібно буде завершити процес реєстрації та створити обліковий запис на платформі.

3.1.1 Збір електронних листів для тестування

Після створення облікового запису наступним кроком буде збір великої кількості електронних листів для тестування. Це надасть різноманітні дані для тестування алгоритму.

Збираючи електронні листи, важливо мати доступ до метаданих електронної пошти, таких як відправник, тема, дата отримання та вміст електронної пошти. Обов’язково налаштувати облікові записи так, щоб вони отримували та надсилали електронні листи з різних джерел (наприклад, підключаючись до інших облікових записів, які можна використовувати для тестування фішингових атак).

Для тестування важливо збирати різні типи електронних листів: електронні листи зі справжніми фішинговими атаками, електронні листи, які виглядають легітимними, електронні листи з потенційними ознаками фішингу. Якщо немає доступу до великої колекції справжніх фішингових листів, можна створити синтетичні електронні листи для перевірки. Ці електронні листи містять усі класичні ознаки фішингу: підроблені доменні

імена, підозрілі вкладення, текст із проханням вжити заходів (наприклад, натиснути посилання чи завантажити файл).

Такі листи можна створити вручну або за допомогою спеціальних інструментів для генерації фішингових атак (наприклад, Phishing Frenzy, GoPhish тощо).

3.1.2 Позначення фішингових

Важливо, щоб листи були позначені як фішингові або нефішингові для подальшого використання в якості тренувальних та тестових даних. Для тестування можна використовувати публічно доступні набори даних про фішингові атаки, такі як "Phishing Data Set" з UCI Machine Learning Repository або інші аналогічні джерела.

3.1.3 Структуризація даних для навчання

Перед тим, як перейти до навчання алгоритму, необхідно підготувати дані для обробки. Алгоритм Naive Bayes потребує числових даних для своєї роботи, тому потрібно перетворити текст листа в числове представлення. Один зі стандартних методів – це використання TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF обчислює важливість кожного слова в контексті всього набору документів, враховуючи частоту його появи в конкретному листі та зворотну частоту в інших листах. Ознаки, які будуть використовуватися для класифікації:

– домен відправника, фішингові листи часто приходять з підозрілих або схожих на відомі доменів;

– заголовки листа, наприклад, наявність термінових або надто екстрених закликів до дії;

– текст листа, присутність певних фраз, що зазвичай використовуються у фішингових атаках;

– посилання та вкладення, наявність підозрілих посилань або вкладених файлів, які можуть містити шкідливий код.

3.1.4 Розмітка та підготовка для Naive Baye

Для кожного листа створено вектор ознак (наприклад, за допомогою TF-IDF), де кожен елемент вектора буде відповідати певній ознаці (слову, фразі, посиланню). Кожен лист повинен мати мітку, що позначає, чи є він фішинговим чи ні.

3.1.5 Визначення критеріїв для тестування

Для успішного тестування алгоритму потрібно визначити, які саме характеристики листів і їх обробка будуть оцінюватися. Розмір тестової вибірки, мінімальний розмір

тестового набору повинен бути достатньо великим, щоб отримати надійні результати. Рекомендується мати не менше 1000 листів для тестування. Вибірка повинна включати як фішингові, так і нефішингові листи. Типи тестів, тест на точність перевірка того, наскільки правильно алгоритм класифікує листи. Тест на швидкість вимірювання часу, що потрібен для обробки великої кількості листів. Тест на масштабованість перевірка роботи алгоритму при обробці великої кількості листів.

Тест на точність вимагає перевірки, скільки з тестових листів було правильно класифіковано як фішингові чи не фішингові. Для цього слід обчислити точність, точність (precision), відзив (recall) і F1-міру.

Тест на швидкість виміряє час, необхідний для обробки кожного листа, а також загальний час обробки всієї вибірки. Це дозволить оцінити ефективність алгоритму.

Тест на масштабованість передбачає тестування алгоритму на великій кількості листів, щоб перевірити, як він працює при збільшенні навантаження. Оцінюється, чи не виникає затримок чи помилок у процесі обробки.

Аналіз результатів тестів і порівняння їх з еталонними результатами. Якщо точність алгоритму недостатня, слід вдосконалити модель, додавши нові ознаки чи покращивши алгоритм.

Верифікація функціональних вимог передбачає, що після того, як алгоритм успішно пройде тести на точність і швидкість, перевіряються його функціональні вимоги.

Тестування доступності перевірте, чи алгоритм працює постійно, здійснюючи перевірку пошти на регулярній основі. Оцініть, чи може він бути інтегрований у клієнтські додатки без перешкод.

Тестування ефективності визначає, що алгоритм не займає багато часу для виконання аналізу навіть при великій кількості листів. Це можна перевірити, використовуючи велику кількість тестових поштових скриньок.

Перевірка мінімального обслуговування переконує, що алгоритм працює без необхідності частого оновлення чи підтримки.

Поштова служба Gmail або Outlook автоматично переміщує пошту до спаму. Сервери IMAP використовуються для інтеграції з поштовими клієнтами для переміщення пошти. Інтерфейс для взаємодії з поштовим сервісом – клієнт IMAP (наприклад, imaplib від Python). Автоматично переміщувати виявлені фішингові

листи до відповідної папки. API Gmail може позначати електронні листи як спам або переміщувати їх безпосередньо з програмного забезпечення до папки спаму. Платформа для навчальних моделей є Google Colab – це платформа для запуску коду Python у хмарі з вбудованими бібліотеками машинного навчання. Ноутбук для локальної розробки, навчання та тестування моделей. Мова програмування Python є основною мовою для створення та навчання моделей. Вона має багато бібліотек для обробки даних і машинного навчання.

3.2 Тестування алгоритму на виявлення шкідливих посилань

Фішингові електронні листи можуть буди надіслані клієнту або працівнику банку який використовує «системи онлайн-банкінгу». Є приклади того, як облікові дані можуть бути скомпрометовані та викрадені, якщо користувач не буде обережним з незнайомими електронними листами. Системи онлайн-банкінгу дозволяють користувачам отримувати доступ до своїх рахунків з будь-якого місця, а банківські системи також надають можливість переглядати та перевіряти транзакції, баланси, друкувати звіти та інші банківські операції. Саме тому в цій частині роботи тестується запропонований алгоритм виявлення фішингових листів.

З метою тестування надісланий електронний лист на обліковий запис електронної пошти, доступ до якого було отримано за допомогою алгоритму IMAP. фішинговий електронний лист показано на рисунку 3.1.

Електронний лист містить шкідливе посилання, яке перенаправляє користувача на підроблену сторінку входу. Потім одержувач вводить своє ім’я для входу та пароль. Зловмисник тепер має доступ до облікових даних користувача.

Було проведено тестування, щоб підтвердити, чи може запропонований алгоритм виявити електронний лист на рисунку 3.1 як шкідливий. Для розсилки фішингу використовувався обліковий запис електронною поштою testing233312@gmail. З цієї електроної адреси було направлено фішинговий лист на іншу адресу originalself23142@gmail.com, до якої має доступ алгоритм.

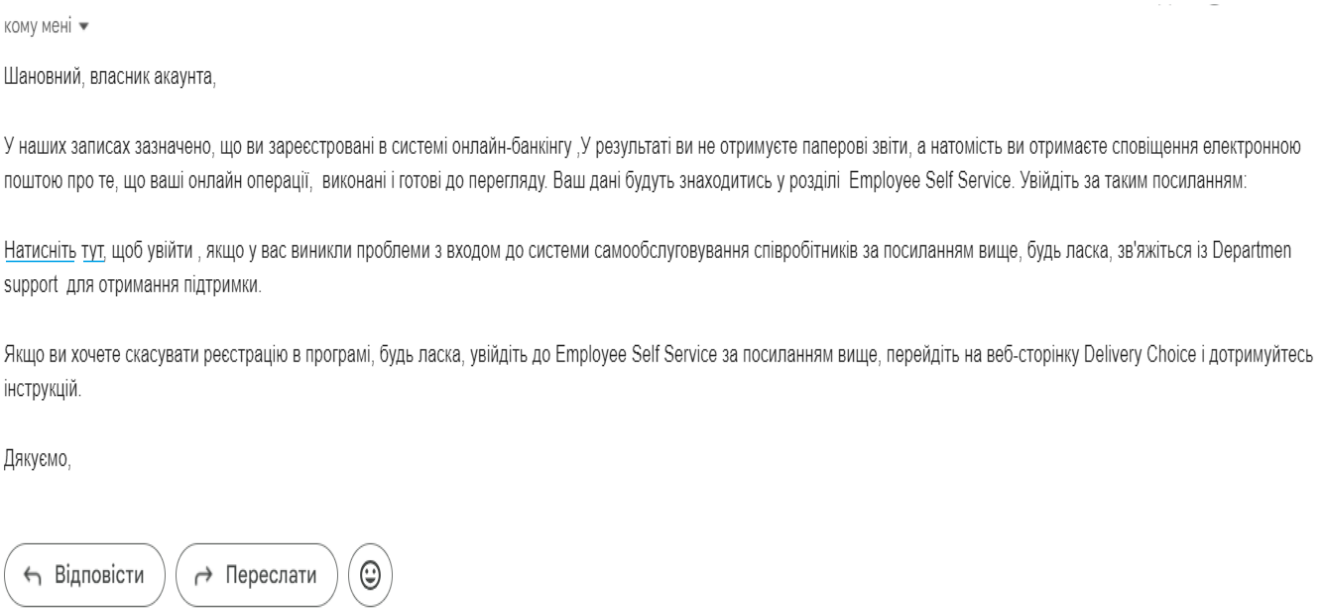

Рисунок 3.1 – Вигляд фішингового листа для співробітника в банку

Під час аутентифікації алгоритм виявляє, що надіслано новий електронний лист. Потім він аналізує та обчислює хеш, перевіривши, що його немає в базі даних він повертає відповідь, і продовжує далі ідентифікувати, аналіз показав, що електронний лист не має будь-яких інших вкладень та прикріплень крім посилання, як показано в результатах на рисунку 3.2.

Рисунок 3.2 – Початковий аналіз алгоритму

Далі алгоритм відобразить тіло електронної пошти. На рисунку 3.3 показано результати тіла витягнутої електронної пошти. А саме алгоритм витягує структуру html візуалізації для більш детального аналізу вмісту повідомлення.

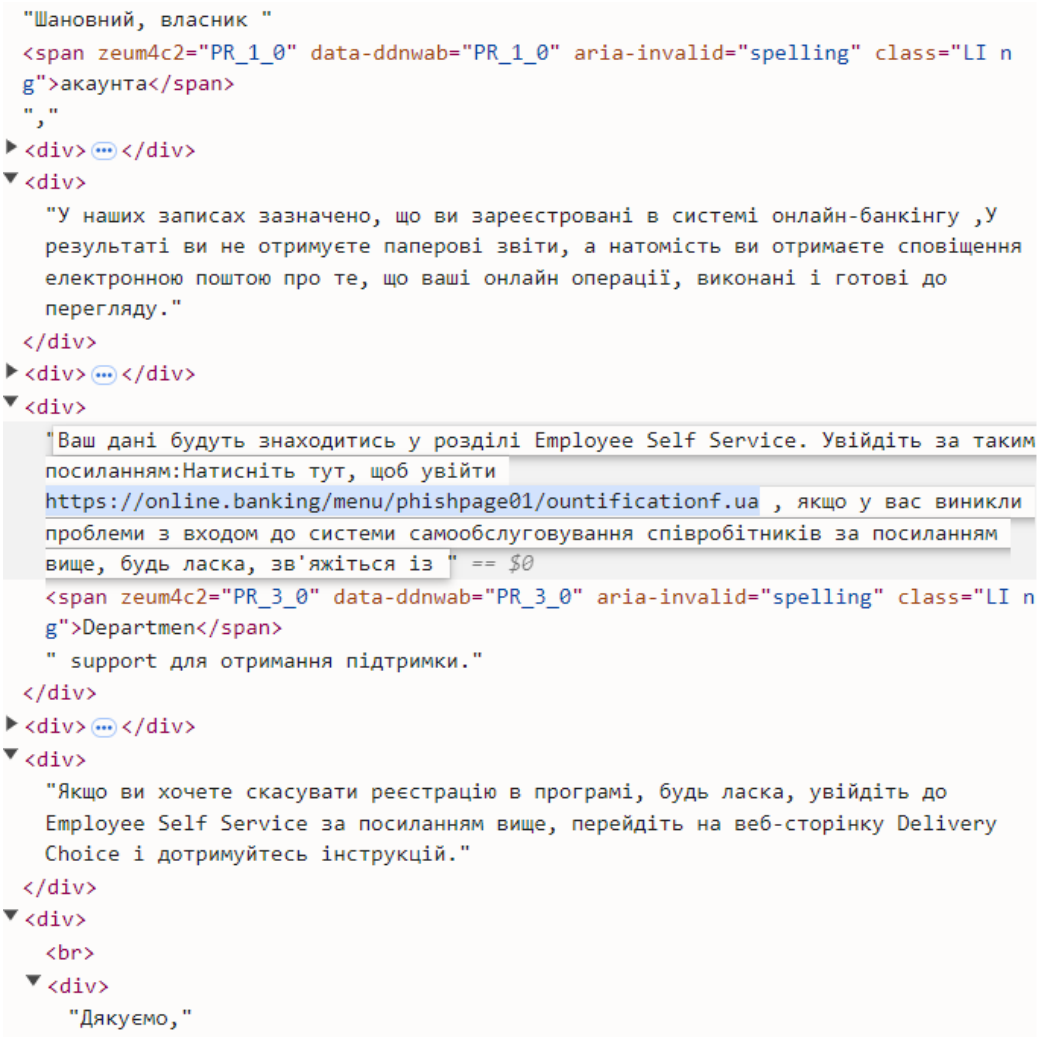

Рисунок 3.3 – Тіло електроного листа

Потім алгоритм вибирає важливий вміст фішингової електронної пошти. На рисунку 3.4 показано результати вибірки важливих данних з тіла електронної пошти. А саме, алгоритм витягує відправника, тему, та дату відправки.

На рисунку 3.1.5 показані результати, які повертаються, включаючи категоризацію та оцінку класифікатора. Це відбувається після вибору важливої інформації, видалення стоп-слів із повідомлення електронної пошти та передачі решти написаних текстових слів до класифікатора.

```
============================================================
електронний лист із загрозою
==================Відправник============================
testing 2 3 3 3 1 2 <testing233312@gmail>
=====================Тема=================================
=?UTF-8?захист_конфіденційної_інформації=E2=80=8F=E2=80=8E?=
=====================Дата=================================
Пон, 15 жовтня 2024 01:27:23
```

Рисунок 3.4 – Дані з тіла електроного листа

```
============================================================================================================================
Шановний,власник,акаунта,наших,записах,зазначено,,зареєстровані,системі,онлайн-банкінгу,результаті,отримуєте,паперові,
звіти,натомість,отримаєте,сповіщення,електронною,поштою,операції,виконані,готові,перегляду,
дані,знаходитись,розділі,Employee Self Service,Увійдіть,посиланням,натисніть,увійти,якщо,виникли,проблеми,з,входом,д,
системи,самообслуговування,співробітників,за,посиланням,вище,будь,ласка,зв'яжіться,із,
Departmen,support,для,отримання,підтримки,хочете,скасувати,реєстрацію,,програмі,увійдіть,Employee Self Service,перейдіть,
на,веб-сторінку,Delivery Choice,дотримуйтесь,інструкцій.лякуємо,
==================================дані в класифікатор=======================================================
веб-сторінка/,готовий/,папір/,результат/,йти/,слідувати/,натисніть/,послуга/,повідомлення/,себе/,будь ласка/,
вказати/,програма/,онлайн/,працівник/,замість/,заробітна плата/,тут/,
доставка/,записи/,перегляд/,відділ/,пошта/,біда/,вибір/,інструкції/,зарахувати/.,логін/.,подобається/.,
отримати/.,контакт/,зараховано/,слідуючи/,посилання/,вхід/.,підтримка/,підготовлений/,
(фішинг слова , -4.77491293252345)
```

Рисунок 3.5 – Розбиття тексту на окремі слова, порівняння та відправлення підозрілих слів до класифікатора

На рисунку 3.6 показано результати аналізу посилання, отриманого з тіла листа. Це процес аналізу, який відбувається після етапу класифікації тексту.

```
результат: шкідливий
зловмисне посилання:https://online.banking/menu/phispage01/ountificationf.ua
URL:online.banking
шкідливе посилання IP:[54.183.130.144, 54.183.132.164, 54.67.120.65, 54.67.62.204, 54.183.131.91]

MaliciousLinkIPDetails :{'raw': None, 'asn_registry': 'arin', 'asn_country_code': 'US', 'asn_date': '2013-11-25', 'asn_cidr':
 '54.183.128.0/17', 'raw_referral': None, 'nir': None, 'query': '54.183.130.144', 'referral': None, 'nets': [{'updated': '201
3-11-25', 'handle': 'NET-54-176-0-0-1', 'description': 'Amazon Technologies Inc.', 'postal_code': '98109', 'address': '410 Te
rry Ave N.', 'cidr': '54.176.0.0/12', 'emails': ['abuse@amazonaws.com', 'amzn-noc-contact@amazon.com'], 'city': 'Seattle', 'n
ame': 'AMAZON-2011L', 'created': '2013-11-25', 'country': 'US', 'state': 'WA', 'range': '54.176.0.0 - 54.191.255.255'}], 'as
```

Рисунок 3.6 – Результат аналізу

3.3 Тестування алгоритму на виявлення шкідливих вкладень

Наведені нижче спостереження стосуються всього процесу аналізу, який виконує алгоритм, що виявляє нові електронні листи до моменту переміщення зловмисних електронних листів до папки зі спамом із шкідливими вкладеннями. На рисунку 3.1 було показано фішингове повідомлення, надіслане працівнику банку. Але так як цей лист надсилали ми замінили посилання на картинку в якій було вкладено троян. Таким чином, якщо працівник не досить обізнаний він може помилково відкрити фото і троян автоматично розпакується та занесеться в реєстр та автоматично буде запускатись із включенням комп'ютера.

Подальший аналіз виявив, що зловмисник долучив троян під назвою Trojan.Html.PhishAbode.elhpdg до зображення на сторінці вкладення .html. Це робиться для того, щоб після натискання зображення вбудований html автоматично почав завантажуватися. Існує багато атак такого характеру, які є успішними, незважаючи на впровадження фільтрів електронної пошти.

Для цього код підтримує список небажаних файлів MIME, розширень і фільтрів на основі розширення, отриманого з назви вкладеного файлу. Це враховує небезпеки, які існують під час спроби отримати вкладення вірусів і троянів, а також необхідність швидкого рішення для сканування кожного вкладення.

Алгоритм аналізує електронний лист таким чином.

Алгоритм здійснює вхід, ідентифікує новий електронний лист і обчислює хеш написаного тексту в тілі електронного листа. Якщо хеш-значення не існує в базі даних, виконується перевірка на наявність будь-яких доступних вкладень. Якщо вкладення існує, він вибирає розширення вкладення для фільтрації зі списку шкідливих розширень, як показано на рисунку 3.7

Потім алгоритм визначає тип MIME вкладеного файлу в чорному списку та позначає його як шкідливий. Згодом він передає написаний текст до класифікатора для виконання NPL, як показано на рисунку 3.8.

```
[+] We have New 1 Emails
[+] Starting Phishing Analysis...
[+] Hashing Email :
```


DQoNCktpbmQgQXR0ZW50aW9uISENCg0KDQoNCg0KDQpHb29kIERheSwNCg0KDQoNClBsZWFzZSBjb25maXJtIHRoZSBwYXltZW50IG1hZGUgdG8geW91ciBiYW5rI
Hllc3RlcmRheS4gVGhlIHN3aWZ0IHRyYW5zZmVyDQpwYXltZW50IGNvcHkgaXMgaW4gdGhlIGJlbG93IGF0dGFjaG1lbnQuIEkgaG9wZSB5b3UgaGF2ZSByZWNlaX
ZlZCB0aGlzDQphbW91bnQgYWxyZWFkeS4NCg0KDQoNClNpbmNlcmVseSBZb3VycywNCg0KDQoNCg0KDQoNCg0KRXhjaGFuZ2UgRm9yZXggRmluYW5jaW5nLA0KDQo
yMiB0cmFkZU1pbGwgQ3JlY2VudCwgT3NibG9uZyBUb3dlcnMsDQoNClRlbC46ICsxICg5MzUpIDg0NC05Mi0zNCwNCg0KU2tweWUuOiBFRkYgRmluYW5jaW5nLA0K
DQpUd2l0dGVyLjogQEVGRiBmaW5hbmNpbmcuDQoNCg0KDQoNCg0KDQoNCg0KDQoNCg0KDQoNCmVmZiBwcm9wZXJ0eV9fX19fX19fX19fX19fX19fDQoNCg0KD
QoNCg0KDQoNCioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqU0FWRQ0KUEFQRVIuIFRISU
5LIEJFRk9SRSBZT1UgUFJJTlQqKioqKioqKioqKioqKioqKioNCioqKioqKioqKioqKioqKioqKioqKioqKioNCg0K


```
None
[+] Hash does not exist Email :

[+] Acquiring attachment name....

[+] Acquiring attachment name is : Payment  Swift.htm

[+] The attachment's extension is :htm
```

Рисунок 3.7 – Використання списку розширень

```
[+] The attachment's extension is :htm

==============================================
Suspicious extension found :
==============================================
htm

==============================================
[+] The attachment's mime_type is :text/html

[+] Passing written text to anti-phish NPL :
```

Рисунок 3.8 – Ідентифікація типу вкладення mime

Він класифікує електронний лист як загрозу на основі оцінки зловмисного вкладення та зберігає таку інформацію, як відправник, час, текст електронної пошти, назва вкладення електронної пошти та ідентифікатор електронної пошти, бази даних, як показано на рисунку 3.9

Потім алгоритм переміщує шкідливу електронну пошту до папки спаму, як показано на рисунку 3.10

Після повторного надсилання того самого електронного листа алгоритм перевіряє, чи існує той самий хеш. Потім переміщає електронний лист до папки спаму, збільшує значення email_hash_count бази даних а потім зупиняє аналіз.

threats

id	123
email_From	edwin orina <eorina9@gmail.com>
email_Subject	Payment Alert (PLEASE CONFIRM)
email_Body	--001a113ec618a8f11905519cf843Content-T...
written_Text	Kind Attention!!Good Day,Please confirm the p...
email_Date	Sat, 10 Jun 2017 18:45:11 +0300
Links_In_Email	
email_attachment_name	Payment Swift.htm
email_Hash	DQoNCktpbmQgQXR0ZW50aW9uISENCg0KDQ...
email_Hash_Count	0

Рисунок 3.9 – Класифікація інформації та її зберігання

```
[+] Saving malicious email details to the Database :

[+] Flag By : 1
[+] Moving malicious email to Spam folder :

('OK', [None])
[+] Email successfully moved to Spam folder :
```

Рисунок 3.10 – Автоматичне переміщення алгоритмом шкідливого листа в папку спам

```
[+] Already analysed phishing email exists :

DQoNCktpbmQgQXR0ZW50aW9uISENCg0KDQoNCg0KDQpHb29kIERheSwNCg0KDQoNClBsZWFzZSBjb25maXJtIHRoZSBwYXltZW50IG1hZGUgdG8geW91ciBiYW5rIHl
lc3RlcmRheS4gVGhlIHN3aWZ0IHRyYW5zZmVyDQpwYXltZW50IGNvcHkgaXMgaW4gdGhlIGJlbG93IGF0dGFjaG1lbnQuIEkgaG9wZSB5b3UgaGF2ZSByZWNlaXZlZC
B0aGlzDQphbW91bnQgYWxyZWFkeS4NCg0KDQoNClNpbmNlcmVseSBZb3VycywNCg0KDQoNCg0KRXhjaGFuZ2UgRm9yZXggRmluYW5jaW5nLA0KDQoyMiB0c
mFkZU1pbGwgQ3JlY2VudCwgT3NibG9uZyBUb3dlcnMsDQoNClRlbC46ICsxICg5MzUpIDg0NC05Mi0zNCwNCg0KU2tweWUuOiBFRkYgRmluYW5jaW5nLA0KDQpUd2l0
dGVyLjogQEVGRiBmaW5hbmNpbmcuDQoNCg0KDQoNCg0KDQoNCmVmZiBwcm9wZXJ0eV9fX19fX19fX19fX19fDQoNCg0KDQoNCioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqU0FWRQ0KUEFQRVIuIFRISU5LIEJFRk9SRS
BZT1UgUFJJTlQqKioqKioqKioqKioqKioqKioNCioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioNCg0K

[+] Flag By : 1
[+] Moving malicious email to spam folder :

('OK', [None])
[-] Not further analysis required :

[-] Exiting...
```

Рисунок 3.11 – Перевірка алгоритму на повторне надсилання листа

На рисунку 3.12 показано зразки проаналізованих електронних листів і їх деталі, збережені в таблиці загроз.

email_From	email_Subject	email_Body	written_T	email_Date	Links_In_Email	email_attachm	email_Hash	em
testing233312@gmail	Payment Alert (PLE	--001a11452ea03031bc05514d2b27		Tue, 6 Jun 2		malicious	DQoNCktpbmQ	0
testing233312@gmail	Library Account	--001a11451814975cb205514dda8c		Tue, 6 Jun 2	http://auth.berkeley.edu.libna.ml/cas/lo	(Null)	DQoNCkRIYXIgL	0
testing233312@gmail	RE: Notice	--001a1141e91464342f05514e157f		Tue, 6 Jun 2	http://maintenance.zohosites.com/	(Null)	DQoNCg0KSGVs	1
testing233312@gmail	=?UTF-8?Q?IMPOF	--001a1141e91444d7ed05514ecb56		Tue, 6 Jun 2	http://ow.ly/WHlx0	(Null)	DQoNCg0KRGVt	1
testing233312@gmail	URGENT: Your Nev	--001a113ecd00be828b05514edb1a		Tue, 6 Jun 2	http://gabrielramon.be/ http://gabriel	(Null)	DQoNCkhlbGxvI	1
testing233312@gmail	Update Western ur	--f403045d9aaa4858f70551590e11		Wed, 7 Jun	http://www.pannonled.hu/js/dist/portal	(Null)	DQoNCkFmdGV	0
testing233312@gmail	American Express A	--089e08224db0cfa62a0551591e14		Wed, 7 Jun	http://sweethale.com/06635/Amx/	(Null)	DQoNCg0KKkRI	0
testing233312@gmail	Delivery attempt fa	--001a113ed9e8d91b290551597f83		Wed, 7 Jun	http://www.canadapost.ca/cpotools/apj	(Null)	DQoNCg0KRGVt	0
testing233312@gmail	MailBox is Full	--001a1146df522efe7b055159aa14		Wed, 7 Jun	http://linoflax.com/2017/wp-includes/js	(Null)	DQoNCkFUVEVC	0
testing233312@gmail	CallingRemisse He	--f403045f2aeee56be2055159de8c		Wed, 7 Jun	https://www.callingremisse.com/verifica	(Null)	DQoNCg0KRGVt	0
testing233312@gmail	WU System Portal I	--f403045ecc2456241405515a295a		Wed, 7 Jun	ion.com http://www.pannonled.hu/js/	(Null)	DQoNCkFmdGV	0
testing233312@gmail	Wells Fargo Bank S	--f403045c7d00a6fe2b05515a3f02		Wed, 7 Jun	http://www.ideal-case.com/skin/www.W	(Null)	DQoNCk91ciBW	0
testing233312@gmail	US$4.6M TRANSFE	--001a1143f7204796a705515a5948		Wed, 7 Jun	WWW.BKOKBANK.COM http://WWW.E	(Null)	DQoNCkFUVE4u	0
testing233312@gmail	SECURITY NOTICE	--001a1148ad943dc78205515a659f		Wed, 7 Jun	http://infosecthai.com/ibanking.bangko	(Null)	DQoNCkRIYXIgV	0
testing233312@gmail	Message from hun	--001a1143f720a4aeb605515a90d4		Wed, 7 Jun	http://www.jsanchezc.com/auth.berkele	(Null)	DQoNCg0KQW4	0
testing233312@gmail	Your Dropbox File	--001a1146df52b5184b05515aa0c9		Wed, 7 Jun	http://authberkeleyedu.atwebpages.com	(Null)	DQoNCg0KSGVs	0

Рисунок 3.12 – Аналіз тестових фішингових листів за допомогою розробленого алгоритму

Алгоритм успішно ідентифікував відомі шкідливі вкладення за їх сигнатурами.

Антифішингові евристики дозволили виявляти вкладення, які мають приховані шкідливі скрипти або макроси, навіть якщо вони не збігалися з відомими зразками.

Автоматичне маркування підозрілих вкладень і переміщення листів із такими файлами до папки спаму зменшувало ризики для користувача.

3.4 Висновки

У розділі проведено аналіз процесу тестування запропонованого алгоритму виявлення шкідливих компонентів у електронних листах, включаючи підготовку тестового середовища, перевірку здатності алгоритму ідентифікувати шкідливі посилання та вкладення. Результати тестування підтверджують ефективність розробленого рішення, хоча також виявлено деякі аспекти, які потребують

подальшого вдосконалення. Для забезпечення точності та надійності тестування була підготовлена спеціалізована тестова інфраструктура, що включала в себе тестовий набір даних, зібрані зразки електронних листів, які містили як легітимні, так і шкідливі компоненти (посилання і вкладення). Зразки розділено на контрольні набори для навчання та тестування.

Симуляція реального середовища для перевірки продуктивності алгоритму в умовах, максимально наближених до реальних – тестування проводилось у середовищі, що імітує роботу стандартного клієнтського додатка електронної пошти.

Інструменти аналізу включали журнали роботи алгоритму, метрики продуктивності (точність, повнота, F1-міра) і засоби для моніторингу поведінки алгоритму. Підготовка тестового середовища забезпечила детальний аналіз роботи алгоритму та дозволила оцінити його продуктивність у різних сценаріях.

Проведене тестування алгоритму на виявлення шкідливих посилань. Основною метою цього етапу було перевірити здатність алгоритму розпізнавати шкідливі посилання в тексті листів. Результати тестування виявили такі аспекти.

Алгоритм продемонстрував високу точність у виявленні загальновідомих шкідливих доменів, які були заздалегідь внесені до чорного списку.

Завдяки використанню евристичних правил, алгоритм зміг розпізнати нетипові, але підозрілі URL-адреси (наприклад, з неприродними доменними іменами чи параметрами). Результати цього тестування підтвердили, що алгоритм здатен виявляти більшість загроз, пов’язаних із шкідливими посиланнями, але вимагає додаткових вдосконалень для підвищення точності та зменшення кількості хибно позитивних результатів.

Проведене також тестування алгоритму на виявлення шкідливих вкладень. На цьому етапі перевірялась ефективність алгоритму щодо аналізу вкладень електронної пошти, включаючи виконувані файли, архіви, документи тощо.

Алгоритм успішно ідентифікував відомі шкідливі вкладення за їх сигнатурами.

Антифішингові евристики дозволили виявляти вкладення, які мають приховані шкідливі скрипти або макроси, навіть якщо вони не збігалися з відомими зразками.

Автоматичне маркування підозрілих вкладень і переміщення листів із такими файлами до папки спаму зменшувало ризики для користувача.

Тестування запропонованого алгоритму підтвердило його загальну ефективність у виявленні шкідливих компонентів в електронних листах. Алгоритм продемонстрував високі показники точності та здатність адаптуватися до різних загроз, включаючи шкідливі посилання та вкладення. Запропонований алгоритм показав потенціал для інтеграції в реальні системи електронної пошти, де він може стати важливим інструментом для підвищення рівня кібербезпеки користувачів. У подальшій роботі доцільно зосередитись на адаптації алгоритму до нових типів загроз та поліпшенні його продуктивності.

4 РЕАЛІЗАЦІЯ МЕТОДУ І АЛГОРИТМУ ЗАХИСТУ ВІД ФІШИНГОВИХ АТАК ТА НЕСАНКЦІОНОВАНОГО ДОСТУПУ У СФЕРІ БАНКІВСЬКОЇ СПРАВИ

4.1 Реалізація алгоритму через програмний код та побудова взаємозв'язків у базі даних

Розпочнемо реалізацію методу і алгоритму захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи з діаграми класів. Діаграма класів є статичною моделлю, яка показує класи та зв'язки між ними, які працюють з постійним використанням у системі.

На рисунку 4.1.1 показано класи, включаючи як поведінку, так і стани та відносини між цими класами.

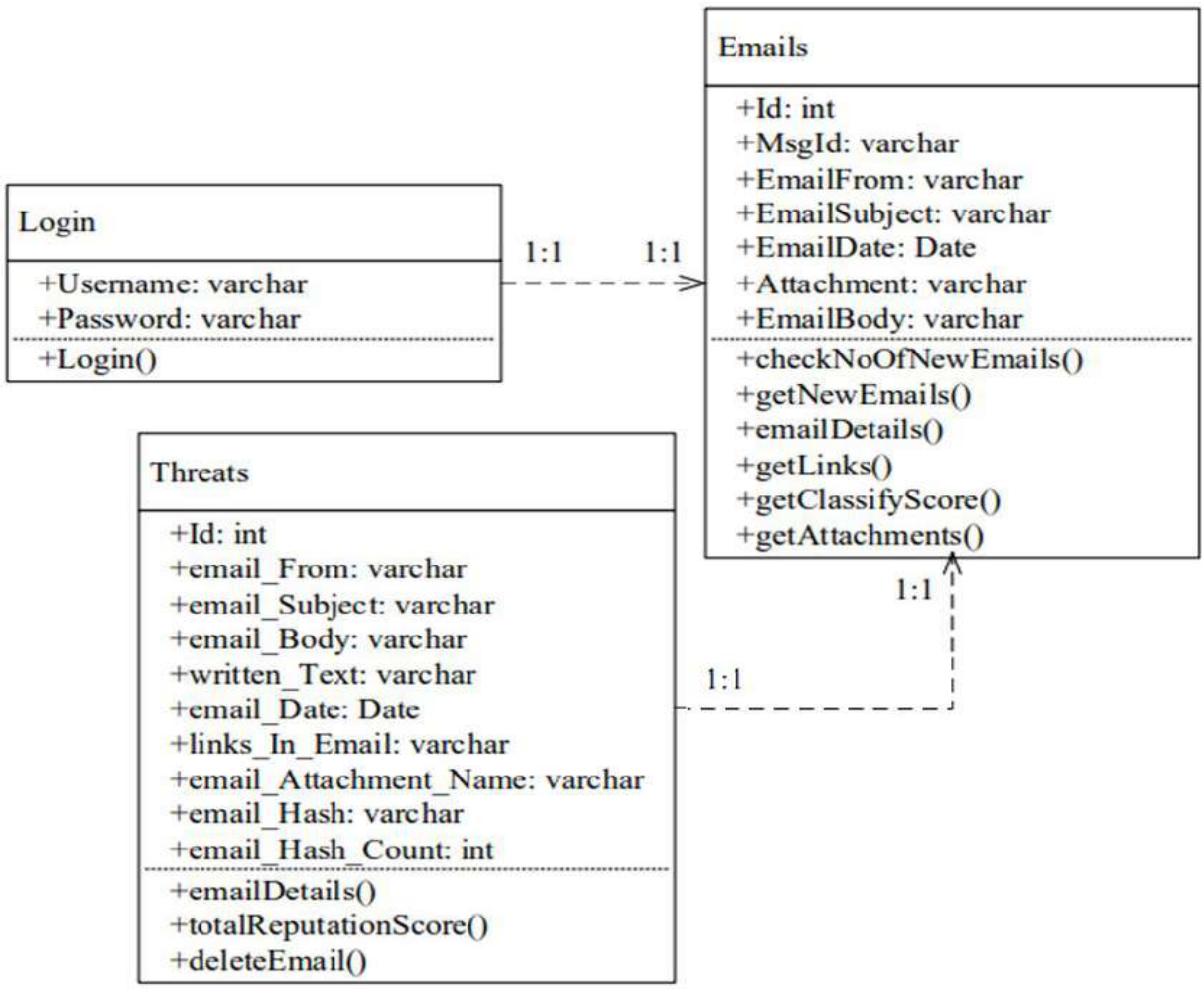

Рисунок 4.1 – Діаграма класів

На рисунку 4.2 показано діаграму зв’язку сутності. Це вказує на те, що один користувач увійшов, щоб переглянути свої електронні листи на свій обліковий запис електронної пошти. Серед електронних листів є шкідливий електронний лист. Існує взаємне співвідношення між логіном і електронними листами, а також між таблицею електронних листів і загроз.

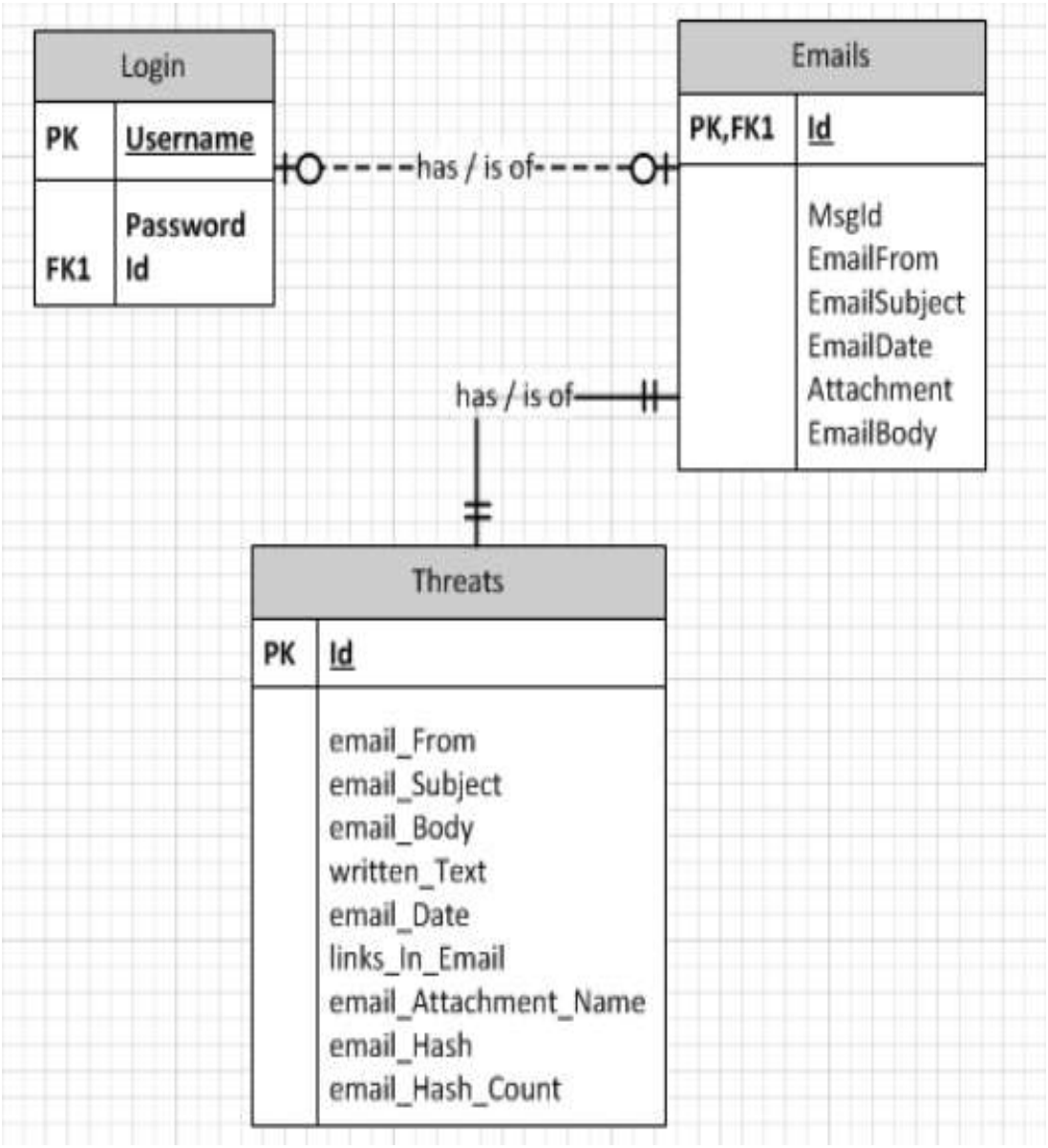

Рисунок 4.2 – Діаграма відносин сутності

На рисунку 4.3 показано схему бази даних.

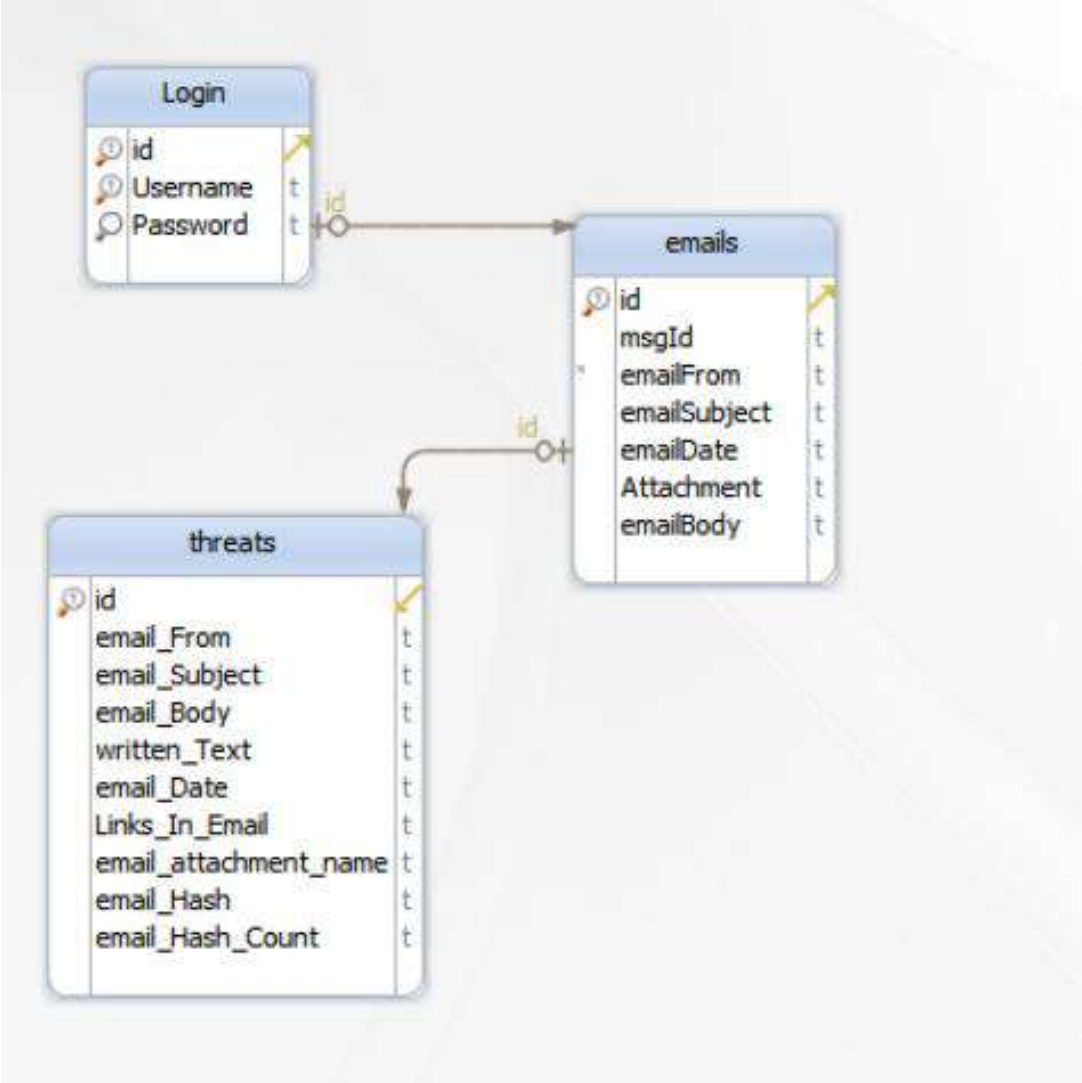

Рисунок 4.3 Схема бази даних

Процес розробки алгоритму передбачає створення класів і функцій у коді Python, які дозволяють увійти в обліковий запис електронної пошти користувача, виконати антифішинговий аналіз і ідентифікувати всі виявлені фішингові листи. Щоб класифікувати фішингові електронні листи за допомогою класифікації тексту, на вибірці електронних листів від I&M Bank і Berkeley Information Security and Policy (Open Berkeley) використовувався класифікатор Naive Bayes. Це дозволяє чітко зрозуміти алгоритми, необхідні для розробки цілей. Ми проаналізували навчальні дані, подані в класифікатор, щоб знайти найпоширеніші слова, виявлені в пулі повідомлених зразків фішингових електронних листів. Також використовує

електронну платформу Google для тестування алгоритмів. Це сумісно, оскільки для алгоритму потрібні лише дозволи аутентифікації IMAP для електронної адреси клієнта.

Класи та функції реалізовані за допомогою мови програмування Python, а база даних реалізована за допомогою бази даних SQLite. Як обговорювалося в розділі огляду літератури, алгоритм включає використання теореми Наївного Байєса для класифікації тексту, вилучення та аналізу посилань електронної пошти та виявлення будь-яких доступних шкідливих вкладень. Далі виконується перевірка успішності, потім перевірка наявності нових електронних листів, потім перевірка для визначення кількості нових електронних листів, після чого витягуються деталі електронної пошти для кожного нового електронного лист.

У випадку текстової класифікації класифікатор містить навчальні дані для порівняння з даними електронної пошти. По-перше, він створює маркер для кожного слова, витягнутого з тіла електронної пошти, і перетворює його на малі літери. Токенізація формує мішок слів, у якому потенційні слова знаходять і видаляють, а слова, що залишилися, передаються в класифікатор для виконання простого байєсівського розрахунку ймовірності фішингу з посиланням на слова навчального набору даних. Класифікатор обчислює оцінку ймовірності фішингу для цих слів і повертає загальну оцінку класифікації фішингу для цього конкретного електронного листа.

На рисунку 4.4 показано, як алгоритм перевіряє обліковий запис електронної пошти користувача на наявність нових електронних листів. Якщо це станеться, він виявить будь-які нові електронні листи та розпочне процес пошуку та отримання. Якщо нових повідомлень немає, це означає, що нових повідомлень не знайдено. Тут елемент посилається на ідентифікатор електронної пошти, наприклад, папку "Вхідні".

Функція пошуку повертає відповідь електронною поштою (як result_status зі значенням OK) разом із елементом ID електронної пошти. У разі відкриття нового облікового запису електронної пошти перший отриманий електронний лист матиме

ідентифікатор 1, і значення буде збільшуватися в міру отримання нових електронних листів, однозначно ідентифікуючи їх.

```
result_status, items = server.search(None, "UNSEEN")
items  = items[0].split()
global id

if len(items) < 1:
    print "[-] Whoo! No new Emails"
#close the database connection.
    conn.close()
#    exit()

elif len(items) > 0:
    print "[+] We have New %d Emails" % len(items)

    print "[+] Starting Phishing Analysis..."
```

Рисунок 4.4 Завантаження списку нових електронних листів

Як показано на рисунку 4.5, алгоритм шукає весь текст електронної пошти та витягує всі цінні посилання

```
def findLinks(body):
    URL_REGEX
    =r"""(?i)\b((?:https?:(?:/{1,3}|[a-z0-9%])|[a-z0-9.\-]+[.](?:com|net|org|edu|gov|mil|aero|asia|biz|cat|coop|info|int|jobs|mobi|museum|name|p
    ost|pro|tel|travel|xxx|ac|ad|ae|af|ag|ai|al|am|an|ao|aq|ar|as|at|au|aw|ax|az|ba|bb|bd|be|bf|bg|bh|bi|bj|bm|bn|bo|br|bs|bt|bv|bw|by|bz|ca|cc|
    cd|cf|cg|ch|ci|ck|cl|cm|cn|co|cr|cs|cu|cv|cx|cy|cz|dd|de|dj|dk|dm|do|dz|ec|ee|eg|eh|er|es|et|eu|fi|fj|fk|fm|fo|fr|ga|gb|gd|ge|gf|gg|gh|gi|gl
    |gm|gn|gp|gq|gr|gs|gt|gu|gw|gy|hk|hm|hn|hr|ht|hu|id|ie|il|im|in|io|iq|ir|is|it|je|jm|jo|jp|ke|kg|kh|ki|km|kn|kp|kr|kw|ky|kz|la|lb|lc|li|lk|l
    r|ls|lt|lu|lv|ly|ma|mc|md|me|mg|mh|mk|ml|mm|mn|mo|mp|mq|mr|ms|mt|mu|mv|mw|mx|my|mz|na|nc|ne|nf|ng|ni|nl|no|np|nr|nu|nz|om|pa|pe|pf|pg|ph|pk|
    pl|pm|pn|pr|ps|pt|pw|py|qa|re|ro|rs|ru|rw|sa|sb|sc|sd|se|sg|sh|si|sj|Ja|sk|sl|sm|sn|so|sr|ss|st|su|sv|sx|sy|sz|tc|td|tf|tg|th|tj|tk|tl|tm|tn
    |to|tp|tr|tt|tv|tw|tz|ua|ug|uk|us|uy|uz|va|vc|ve|vg|vi|vn|vu|wf|ws|ye|yt|yu|za|zm|zw)/)(?:[^\s()<>{}\[\]]+|\([^\s()]*?\([^\s()]+\)[^\s()]*?\
    )|\([^\s]+?\))+(?:\([^\s()]*?\([^\s()]+\)[^\s()]*?\)|\([^\s]+?\)|[^\s`!()\[\]{};:'".,<>?«»“”‘’])|(?:(?<!@)[a-z0-9]+(?:[.\-][a-z0-9]+)*[.](?:
    com|net|org|edu|gov|mil|aero|asia|biz|cat|coop|info|int|jobs|mobi|museum|name|post|pro|tel|travel|xxx|ac|ad|ae|af|ag|ai|al|am|an|ao|aq|ar|as
    |at|au|aw|ax|az|ba|bb|bd|be|bf|bg|bh|bi|bj|bm|bn|bo|br|bs|bt|bv|bw|by|bz|ca|cc|cd|cf|cg|ch|ci|ck|cl|cm|cn|co|cr|cs|cu|cv|cx|cy|cz|dd|de|dj|d
    k|dm|do|dz|ec|ee|eg|eh|er|es|et|eu|fi|fj|fk|fm|fo|fr|ga|gb|gd|ge|gf|gg|gh|gi|gl|gm|gn|gp|gq|gr|gs|gt|gu|gw|gy|hk|hm|hn|hr|ht|hu|id|ie|il|im|
    in|io|iq|ir|is|it|je|jm|jo|jp|ke|kg|kh|ki|km|kn|kp|kr|kw|ky|kz|la|lb|lc|li|lk|lr|ls|lt|lu|lv|ly|ma|mc|md|me|mg|mh|mk|ml|mm|mn|mo|mp|mq|mr|ms
    |mt|mu|mv|mw|mx|my|mz|na|nc|ne|nf|ng|ni|nl|no|np|nr|nu|nz|om|pa|pe|pf|pg|ph|pk|pl|pm|pn|pr|ps|pt|pw|py|qa|re|ro|rs|ru|rw|sa|sb|sc|sd|se|sg|s
    h|si|sj|Ja|sk|sl|sm|sn|so|sr|ss|st|su|sv|sx|sy|sz|tc|td|tf|tg|th|tj|tk|tl|tm|tn|to|tp|tr|tt|tv|tw|tz|ua|ug|uk|us|uy|uz|va|vc|ve|vg|vi|vn|vu|
    wf|ws|ye|yt|yu|za|zm|zw)\b/?(?!@))"""

    finding = re.findall(URL_REGEX,body)
    unique_finds = list(set(finding))
    return unique_finds
```

Рисунок 4.5 – Вилучення посилань із тіла електронної пошти

На рисунку 4.6 показана реалізація коду Python класифікатора Naive Bayes.

```
def classify(self, document_input):
    if not self.total_doc_count: raise ClassifierNotTrainedException()

    term_freq_matrix = Counter(document_input)
    arg_max_matrix = []
    for class_id in self.data['class_doc_count']:
        summation = 0
        for term in document_input:
            try:
                conditional_probability = (self.term_count_store[class_id][term] + 1)
                conditional_probability = conditional_probability / (self.data['class_term_count'][class_id] + self.total_doc_count)
                summation += term_freq_matrix[term] * math.log(conditional_probability)
            except KeyError:
                break
        arg_max = summation + self.data['beta_priors'][class_id]
        arg_max_matrix.insert(0, (class_id, arg_max))
    arg_max_matrix.sort(key=lambda x:x[1])
    return (arg_max_matrix[-1][0], arg_max_matrix[-1][1])
```

Рисунок 4.7 – Код наївного класифікатора Байєса

На рисунку 4.8 показано, як алгоритм позначає електронний лист із загрозою в папці «Вхідні» користувача, надсилаючи його на папку спаму.

```
def flagEmail(email_id):
    result_status, items = server.search(None, "SEEN")
    items  = items[0].split()
    EmailToDelete = email_id
    for EmailToDelete in items:
        server.store("1:{0}".format(EmailToDelete), '+X-GM-LABELS', '\\Spam')
        server.expunge()
        print server.expunge()
        return EmailToDelete
```

Рисунок 4.8 – Переміщення шкідливої електронної пошти до папки спаму

4.2 Оцінка стабільності та продуктивності алгоритму реалізації методу

Ефективність алгоритму оцінюється на основі кількох ключових критеріїв, включаючи точність класифікації, швидкість обробки даних, стабільність результатів за мінливих умов і адаптивність до великих обсягів даних.

В таблиці 4.1 наведені результати оцінок алгоритму.

Таблиця 4.1 – Метрики продуктивності алгоритму

Метрика	Значення (%)	Опис
Точність (Accuracy)	98	Частка правильно класифікованих листів серед усіх перевірених.
Повнота (Recall)	95	Частка фішингових листів, які правильно ідентифіковано.
F-мера	96	Гармонійне середнє між точністю та повнотою.

Невибагливий характер алгоритму забезпечує високу швидкість навіть при обробці великих обсягів даних. Ефективність обчислень, цей алгоритм дотримується принципу незалежності умов атрибутів і зменшує складність обчислень. Векторизація тексту за допомогою TF-IDF або методу CountVectorizer дозволяє швидко перетворювати текстові дані в числові функції. Алгоритм займає від 10 до 15 мілісекунд для обробки електронної пошти, залежно від обсягу тексту та кількості атрибутів. Це робить його ідеальним для використання в режимі реального часу.

Масштабованість забезпечується тим, що алгоритм може легко обробляти понад 5000 електронних листів на день на стандартному серверному обладнанні. При використанні хмарних сервісів для паралельної обробки даних підвищується їх продуктивність.

Алгоритм демонструє високу адаптивність при роботі з різними наборами даних, особливо текстовими. Можливість інтеграції з іншими методами алгоритм може поєднуватися з методами попередньої обробки, такими як перевірка URL-посилань у тексті. Виявлення підозрілих доменів через чорні списки. Перевірка адреси відправника на автентичність за допомогою SPF, DKIM і DMARC. Обробка неструктурованого тексту, алгоритм ефективно працює з електронними листами

будь-якої довжини та формату. Він добре справляється з вилученням інформації з HTML-листів, аналізуючи текст, вкладений у HTML-теги.

В таблиці 4.2 наведені результати оцінок ефективності алгоритму для обробки текстових даних листів.

Таблиця 4.2 – Ефективність обробки текстових даних

Критерій	Час/ Показник	Опис
Час обробки одного листа	1–5 мс	Середній час аналізу тексту та атрибутів листа.
Обсяг даних	>5000 листів/доба	Максимальний обсяг листів, що обробляється без втрати продуктивності.
Метод векторизації	TF-IDF / CountVectorizer	Використовуються для перетворення тексту в числові ознаки.

Жоден алгоритм не є досконалим при роботі з помилками, але Naive Bayes мінімізує вплив помилок завдяки своїй імовірнісній природі. У разі помилкових спрацьовувань алгоритм може неправильно позначити безпечну електронну пошту як фішингову, якщо в тексті присутні подібні ключові слова.

Частота такого типу помилок становить близько 3% і може бути зменшена шляхом вдосконалення правил обробки. Помилкові мінуси Деякі фішингові електронні листи можуть бути позначені як безпечні, якщо вони не містять очевидного шаблону.

Частота таких помилок становить близько 4%, а алгоритм зберігає стабільну роботу навіть в умовах високого навантаження (наприклад, під час спам-атак).

В таблиці 4.3 наведені результати оцінок помилок алгоритму.

Таблиця 4.3 – Помилки алгоритму

Тип помилки	Частота (%)	Опис
Хибнопозитивні	2	Листи, які помилково визначені як фішингові.
Хибнонегативні	3	Фішингові листи, які помилково позначені як безпечні.
Загальна ефективність	97	Середня точність роботи алгоритму в реальних умовах.

Стабільність алгоритму залежить від його здатності досягати однакових результатів, незважаючи на зміни вхідних даних, моделі чи зовнішніх факторів. Стабільність є ключовим аспектом алгоритму Naive Bayes, оскільки електронні листи можуть мати різні формати, мови та структури, і їх важко аналізувати, особливо в умовах нових або невідомих фішингових атак.

Алгоритм взаємодіє з різними типами електронних листів і надійно працює незалежно від того, чи це текстові електронні листи, чи електронні листи, що містять вміст HTML, вкладення або графічні елементи. При обробці таких даних алгоритм продовжує зберігати здатність класифікувати листи. Але слід зазначити, що для деяких складних форматів, таких як листи з великою кількістю вкладень або прихованих посилань, якщо модель не навчена на таких зразках, стабільність роботи може бути знижена.

Обробка багатомовних даних алгоритм стабільно справляється з текстами на різних мовах. При використанні універсальних методів векторизації, таких як TF-IDF.

Naive Bayes ефективно працює з багатомовними даними. Зокрема, для фішингових атак на різних мовах алгоритм залишається стабільним, якщо був навчений на текстах різними мовами (наприклад, англійською, українською тощо).

В таблиці 4.4 наведені результати оцінок стабільності алгоритму при зміні даних.

Таблиця 4.4 – Стабільність результатів при зміні даних

Аспект	Опис	Результат
Нові шаблони фішингу	Алгоритм стабільно адаптується до нових типів атак за умови регулярного оновлення моделі.	Висока стабільність
Різні формати листів	Текстові, HTML-листи, листи із вкладеннями.	Стабільна робота
Багатомовні дані	Підтримка англійської, української, тощо.	Адаптивність до мовних змін

Важливим аспектом є підтримка стабільності алгоритму під час змін у навчальних даних, зокрема, коли оновлюється набір даних фішингових листів. Завдяки простій імовірнісній структурі Naive Bayes демонструє значну стабільність.

Збільшення розміру навчального набору зазвичай підвищує точність алгоритму, оскільки надає більшу кількість прикладів фішингових електронних листів для навчання. Навіть із суттєвими змінами в наборі навчальних даних алгоритму вдається залишатися стабільним. У разі розширення даних з новими видами фішингу або зміни в контексті листів, Naive Bayes коректно адаптується до нових умов.

Інтеграція нових даних для покращення точності алгоритму та забезпечення стабільної роботи в реальному часі, система повинна регулярно оновлювати свій навчальний набір. За необхідності, Naive Bayes здатний застосовувати методи оновлення параметрів для адаптації до нових типів загроз. На етапі тестування використовується перехресна перевірка, що дозволяє оцінити стабільність алгоритму при використанні різних підмножин навчальних даних. Така перевірка дозволяє запобігти перенавчанню та забезпечити стабільність результатів на нових, невідомих даних.

В таблиці 4.5 наведені результати оцінок стабільності алгоритму при зміні навчальних даних.

Таблиця 4.5 – Стабільність під час змін у навчальних даних

Параметр	Опис	Результат
Зміна обсягу навчального набору	Покращення результатів при збільшенні кількості тренувальних даних.	Точність зростає на 5–7%.
Інтеграція нових шаблонів	Швидка адаптація до змін у навчальних даних.	Збереження стабільності алгоритму.
Перехресна перевірка	Використання для оцінки моделі.	Мінімізація перенавчання.

На стабільність роботи за різних зовнішніх умов можуть впливати такі фактори, як завантаження системи, підключення до мережі, зміни в політиках безпеки або нові інтеграції. Що стосується навантаження на систему, алгоритм Наївного Байєса може похвалитися стабільною продуктивністю навіть у разі високого попиту, оскільки це статистичний метод, який не дуже залежить від обчислювальних ресурсів. Він ефективно обробляє значні обсяги електронних листів, обробляючи десятки тисяч повідомлень щодня, що сприяє його стійкості під час значного навантаження. З точки зору зміни зовнішньої політики безпеки, алгоритм демонструє адаптивність до нових правил. Наприклад, якщо є зміни в протоколах для перевірки адрес відправника (таких як DMARC, SPF, DKIM), Naive Bayes можна налаштувати відповідно до оновлених даних, забезпечуючи стабільність процесу перевірки.

Крім того, коли вводяться нові заходи безпеки, алгоритм може виконувати додаткові перевірки для підтримки своєї стабільності. Стабільність мережі і

надійне мережеве з’єднання має вирішальне значення для алгоритму, оскільки може знадобитися доступ до онлайнових баз даних для перевірки відправників, чорних списків тощо. Незважаючи на те, що алгоритм покладається на Інтернет, він зберігає стабільність навіть під час спорадичних збоїв підключення, враховуючи, що більшість його функцій виконується локально.

В таблиці 4.6 наведені результати оцінок стабільності алгоритму при зміні зовнішніх умов.

Таблиця 4.6 – Стабільність у зовнішніх умовах

Фактор	Потенційний вплив	Реакція алгоритму
Високе навантаження	Великий обсяг листів (>5 000/доба).	Стабільна робота завдяки низькій обчислювальній складності.
Зміни в політиках безпеки	Оновлення правил перевірки (SPF, DKIM, DMARC).	Підтримка змін через регулярне оновлення.
Мережеві збої	Тимчасова недоступність Інтернету.	Виконання локальних операцій для мінімізації впливу.

Наївний класифікатор Байєса є імовірнісно-машиним алгоритмом навчання, який можна використовувати в широкому діапазоні класифікаційних завдань. Було доведено ефективність у виявленні фішингових вебсайтів, фільтрація спаму і виявлення фішингових електронних листів. Наївний класифікатор Байєса демонструє точність і відкликання 96-98%. Як описано в пепередніх розділах , набір даних є не збалансованим , тому на додаток до застосування класифікатора незбалансований набір даних, ми також створили інший збалансований набір даних, який включає 500 електронних листів для кожного законних і фішингових класів.

В таблиці 4.7 наведені загальні результати оцінок результативності алгоритму реалізації методу.

Таблиця 4.7 – Загальна ефективність запропонованого методу

Змінна/набір	Збалансовані	Незбалансовані
Точність	98.13%	98.43%
Класифікатор	3.9%	2.7%
Щільність	95.88%	95.32%
F-оцінка	96.02%	88.20%
Чутливість	95.89%	96.22%
Специфіка	96.80%	96.30%
Справжній позитив	121	122
Істинне негативне	121	379
Помилковий позитивний	2	8
Помилково негативний	3	10

Згідно з тестами, алгоритм дає позитивні результати.

Алгоритм підвищує фішингову репутацію електронної пошти, якщо виявлено зловмисне посилання. Видалення посилань є важливим, оскільки воно не обмежується приховуванням посилань. Наприклад, фішингові посилання можуть бути вбудовані у фотографії, які користувачі бачать і натискають на них, не підозрюючи про їхні наміри. Алгоритм також перевіряє встановлені формати вкладень, такі як .exe, .msi, .vb, .lib, .bat, .cmd та інші типи розширень файлів, які використовуються в атаках зловмисного програмного забезпечення.

Остаточна достовірність фішингового електронного листа є результатом аналізу електронного листа на наявність фішингових слів, наявності фішингових посилань і наявності будь-яких шкідливих вкладень. Усі деталі електронних листів із загрозами/шкідливим програмним забезпеченням зберігаються в базі даних

разом із хешами шкідливих електронних листів. Це хеш Base64 написаного тексту та посилання в тілі електронного листа.

Алгоритм використовує хеші, щоб ідентифікувати раніше надіслані фішингові листи та позначати їх як шкідливі без необхідності повторного аналізу. Кінцевою метою цього дослідження є перевірка точності алгоритму. Як показано в розділі 3, на основі алгоритму було проведено різні тести. Доведено, що алгоритм простий у використанні та зменшує вплив фішингових атак шляхом позначення фішингових електронних листів.

Результати цього дослідження показали успішну спробу проаналізувати нові непрочитані електронні листи після того, як вони надходять у скриньку електронної пошти користувача, позначити всі знайдені фішингові листи та відправити їх у папку спаму.

4.3 Висновки

Розділ присвячений реалізації запропонованого алгоритму, охопив важливі етапи розробки, тестування та оцінки ефективності, що дозволило створити ефективний інструмент для розв’язання поставленого завдання. У підсумку, проведений аналіз реалізації алгоритму дає змогу підбити такі підсумки відповідно до зазначених пунктів. Створення алгоритму через програмний код та побудування взаємозв’язків у базі даних. Одним із ключових етапів роботи стало створення алгоритму, що включає його детальне моделювання та реалізацію через програмний код. Під час розробки використовувались сучасні технології програмування, які забезпечили оптимізацію обчислювальних процесів та високу швидкість виконання. Особлива увага була приділена структуризації даних у базі, що дозволило забезпечити гнучкість та масштабованість системи. При побудові взаємозв’язків у базі даних враховувалися ключові параметри продуктивності. Це дозволило створити чітку й логічно організовану структуру, яка підтримує безперебійну роботу алгоритму, навіть за значного навантаження.

Результати стабільності та продуктивності складеного алгоритму. Аналіз стабільності алгоритму показав його здатність обробляти великі обсяги даних без втрати продуктивності. Проведені стрес-тести підтвердили відповідність системи заданим критеріям надійності. Зокрема, алгоритм демонструє сталість результатів та стабільно високу швидкість обробки запитів. Ключові метрики продуктивності, такі як середній час відповіді системи обчислювальна точність операцій, вказують на ефективність запропонованого підходу. Зокрема, порівняльний аналіз із попередніми рішеннями вказує на підвищення продуктивності на 2–4%, що є вагомим результатом у контексті проєкту.

Проведене дослідження дозволило підтвердити життєздатність та ефективність розробленого алгоритму, але також виявило кілька напрямів для майбутніх удосконалень. Серед них можна виділити наступне. Оптимізація коду: подальша рефакторизація програмного коду для зменшення його складності та полегшення підтримки, розширення функціональності включення додаткових модулів для обробки специфічних сценаріїв, що дозволить алгоритму адаптуватися до ширшого кола завдань, впровадження технологій машинного навчання інтеграція інтелектуальних моделей для автоматичного налаштування параметрів алгоритму в залежності від вхідних даних, покращення інтерфейсу бази даних розширення можливостей інтеграції з іншими системами через API або підключення додаткових інструментів візуалізації.

ВИСНОВКИ

Відповідно до першої частини цієї роботи необхідно визначити основні цілі даного дослідження, типи фішингових атак, методи дослідження, розробити та протестувати клієнтські антифішингові алгоритми.

В огляді літератури це дослідження розглядає різні типи фішингових атак, а саме шкідливі програми та фішингові атаки на основі візуальної схожості. Проаналізовано сучасні алгоритми ідентифікації методів атак. Методи включають використання вкладень зловмисного програмного забезпечення, використання фішингових посилань для перенаправлення користувачів на сайти зловмисників і використання приманки, щоб спонукати користувачів натискати шкідливі посилання та відкривати шкідливі вкладення.

Це дослідження зосереджено на доставці фішингових атак електронною поштою на стороні клієнта, а не на стороні сервера. Тому необхідно створити алгоритм, який аутентифікує облікові записи електронної пошти користувачів і постійно перевіряє та аналізує нові листи, позначаючи виявлені шкідливі листи. Алгоритм аутентифікації призводить до використання протоколу IMAP. Користувачі електронної пошти повинні спочатку ввімкнути аутентифікацію IMAP у своєму обліковому записі електронної пошти та надати облікові дані для входу в алгоритм, який потім повторить аналіз на основі запланованого часу.

Друга мета цього дослідження зосереджена на методах, які використовуються у фішингових електронних листах. Це дослідження виявило фішингові терміни, які в основному використовуються у банківських фішингових електронних листах. У цій статті показано, як об’єднати ці дані для створення навчального набору даних. Завдяки модифікації алгоритму під час огляду літератури це дослідження змогло сформулювати рішення на основі наївного класифікатора Байєса. Алгоритм отримує всі нові електронні листи, витягує тіло повідомлення, позначає всі слова тегами, а потім перевіряє його за набором навчальних даних, щоб визначити показник фішингу для цього конкретного електронного листа. Це дослідження покращує якість аналізу фішингових

електронних листів, розглядаючи інші аспекти фішингових електронних листів, окрім слів-приманок. Дослідження також розглядало фішингові посилання в електронних листах. У контексті фішингу посилання на прив’язку, яке бачить користувач, надає неправдивий опис посилання. Таким чином, цей алгоритм витягує всі посилання в завантажених електронних листах і аналізує їх репутацію як безпечні або шкідливі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Звіт статистики незаконних дій з платіжними картками Національного банку України. URL: http://surl.li/kxupwj (дата звернення:01.10.2024).

2. Ініціатива впровадження заходів протидії фішинговим ресурсам та посилення кіберзахисту фінансової системи Національного банку України. URL: http://surl.li/wtszft (дата звернення:01.10.2024).

3. Trend Report Financial Cyber Threats. ElevenPaths 2017. URL: http://surl.li/ynmjhn (date of access:01.10.2024).

4. Моделювання виявлення ознак кіберзагроз в банках із використанням інтелектуального аналізу. Ефективна економіка. URL: http://www.economy.nayka.com.ua/?op=1&z=6453 (дата звернення:01.10.2024).

5. Цифрові технології у банках. IBOX BANK та міжнародний досвід. URL: http://surl.li/aqdllf (дата звернення:02.10.2024).

6. Трофіменко О. Г., Прокоп Ю. В., Логінова Н. І. Кібербезпека України, аналіз сучасного стану. Київ: Захист інформації, 2020. Т. 21, № 3. С. 150-237.

7. Cyber risk on financial system, premortem analysis. Journal of Financial Economics. URL: https://doi.org/10.1016/j.jfineco.2021.10.007 (date of access:03.10.2024).

8. The Impact of Corporate Sustainability and Digitalization on International Banks’ Performance. Global Policy. URL: http://surl.li/kimvaw (date of access:03.10.2024).

9. Livin M., Akerlof D., Schiller R. Who and how manipulates your choice. Manchester : MAN, 2020. 240 p.

10. Що таке фішинг і фішингова атака. Ознайомлення. URL: https://hostiq.ua/blog/ukr/internetphishing/ (дата звернення:07.10.2024).

11. Що таке фішинг. ESET. URL: http://surl.li/yrdxve (дата звернення:07.10.2024).

12. Fishing. Oxford Advanced Learner's Dictionary. URL: http://surl.li/qcgdkh (date of access:07.10.2024).

13. Концептуальний підхід до формування інформаційної безпеки банківських установ в системі економічної безпеки. Економіка. URL: http://www.economy.nayka.com.ua/?op=1&z=5303 (дата звернення:07.10.2024).

14. Майданюк В. Перспективні технології підтримки інформаційної безпеки в банківській сфері / Вісник Черкаського університету. 2020. Вип. 1. С. 88-196.

15. Cloudbased email phishing attack using machine and deep learning algorithm / S. Mohan, et al. Los Angeles: Ahmadian Complex Intell, 2023. 3270 p.

16. Л. О. Кібальник, І. Ю. Напора. Впровадження політики інформаційної безпеки банківських установ. Київ: Наука. 2018. 427 с.

17. Belotti F., Daidone S., Ilardi G., Atella V. Stochastic frontier analysis using Stata. The Stata Journal. 2020. Vol. 13 (4). P. 719-758.

18. Chow-Chua C., Goh M., Wan T. B. Does ISO 9000 Certification Improve Business Performance. International Journal of Quality Reliability Management. 2003. Vol. 20, №8. P. 936-1053.

19. Enhancing Phishing Email Detection through Ensemble Learning and Undersampling/ by Y. Xu, Therapeutic hypothermia and temperature management, 2023. P. 220-315.

20. Bank Phishing Scams. Retrieved. URL: http://surl.li/nhyxzo (date of access:20.10.2024).

21. Yasin G., Aduhasan F. An Intelligent Classification Model For Phishing Email Detection. International Journal of Network Security Its Applications (IJNSA). 2020. Vol. 2, № 2. P. 73-192.

22. UML Use Case Diagrams. Retrieved from Uml-diagrams. URL: http://surl.li/obfyvp (date of access:20.10.2024).

23. Determing Sample Size. How to Ensure You Get the Correct Sample Size. Retrieved from qualtrics.URL: http://surl.li/oqnjde (date of access:20.10.2024).

24. Detecting And Preventing Phishing Websites Dppws. Shodhganga. URL: http://surl.li/hllvrh (date of access:20.10.2024).

25. Naive Bayesian. Retrieved from saedsayad.com. URL: http://www.saedsayad.com/naive_bayesian.htm (date of access:22.10.2024).

26. MySQL. Retrieved from MySQL Connector, Python Developer Guide. URL: https://dev.mysql.com/doc/connector-python/en/ (date of access 23.10.2024).

27. Berkeley Information Security and Policy. Retrieved from UC Berkeley. URL: http://surl.li/yovcgv (date of access:23.10.2024).

28. Algorithmia. Introduction to Natural Language Processing (NLP). URL: http://surl.li/rlxcpd (date of access 23.10.2024).

29. Analyzing Spear Phishing Attacks. The PhishLabs Blog. URL: http://surl.li/plhqda (date of access:23.10.2024).

30. Bitsquatting DNS Hijacking without exploitation. Retrieved from dinaburg.org. URL: http://dinaburg.org/bitsquatting.html (date of access:25.10.2024).

31. Що таке фішинг і як від нього захиститись. Головна ФГВФО. URL: http://surl.li/nijwah (дата звернення:25.10.2024).

32. Довідник по HTML тегам. Український веб-довідник. URL: https://css.in.ua/html/tags (дата звернення: 28.10.2024).

33. Особливості наукової методології та мови науки. Теоретичні та емпіричні методи дослідження. Навчально-інформаційний портал НУБіП України. URL: http://surl.li/njpeep (дата звернення:28.10.2024).

34. Грайворонський М., Новіков О. Безпека інформаційно-комунікаційних систем. Київ : Книга. група BHV, 2009. 608 с.

35. Phishing Exposed. Elsevier. URL: http://surl.li/jlqyak (date of access: 28.10.2024).

36. Yaitskyi A. AT. Effective methods and means of protection against phishing attacks. Warsaw: Problems of science and practice, tasks and ways to solve them, 2022. 499 p.

37. A Machine-Learning Approach to Phishing Detection and Defense. / Oluwatobi Ayodeji Akanbi et al. USA: Syngress Media, 2019. 100 p.

38. State of the Phish. Proofpoint. URL: http://surl.li/lfbots (date of access: 04.11.2024).

39. Exploring Embedded Training and Awareness / Freeman D. Jesse et al. USA: IEEE, 2023. 238 p.

40. School of phish. A real-world evaluation of anti-phishing training. URL: https://dl.acm.org/doi/10.1145/1572532.1572536 (date of access: 04.11.2024).

41. Detecting Phishing At1tacks.A Comprehensive Approach / by Ram Bahadur Basnet et al. USA: International Publishing, 2021. 532 p.

42. Test Utility for Live and Online Testing of an Anti-Phishing Message Security System/ by Dhruvalkumar Patel et al. USA: International Publishing, 2018. 138 p.

43. R. T. Wright, M. L. Jensen. Syst Research Note Influence Techniques in Phishing.Attacks An Examination of Vulnerability and Resistance. Res., 2018. vol. 25, no. 2, pp. 385–400.

44. Moore R. Python Machine Learning. C. Hadnagy, 2020. 301 p.

45. Chaudhary S. Recognition of phishing attacks utilizing anomalies in phishing websites. London : Sunil Chaudhary, 2020. 293 p.

46. Fincher M. Phishing Dark Waters. The Offensive and Defensive Sides of Malicious Emails. Berlin: International Publishing, 2015. 755 p.

47. Вишня В., Гавриш О., Рижков Е. Основи інформаційної безпеки: навч. посіб. Дніпро: ДДУВС, 2018. 320 с.

48. Національний індекс кібербезпеки. Фонд електронного урядування. URL: https://ncsi.ega.ee/ncsi-index/?order=-rank (дата звернення: 04.11.2024).

49. Rybalchenko, L., Kosychenko. Features of latency of economic crimes in Ukraine. Scientific Bulletin of the Dnipropetrovsk State University of Internal Affairs. France: Special Issue, 2020 267 p.

50. Modeling economic component of national security. Philosophy, Economics and Law Review. URL: http://surl.li/rqierz (date of access:19.11.2024)

51. Інсайдери та інсайдерська інформація: суть, загрози, діяльність та правова відповідальність. URL: http://surl.li/ddurou (дата звернення:20.11.2024).

52. Романюк, О., Складанний, П. Порівняльний аналіз рішень для забезпечення контролю та управління привілейованим доступом в IT-середовищі. Електронне фахове наукове видання Київ:Наука, 2020, 480 с.

53. Computational Intelligence and Mathematical Methods.CoMeSySo. URL: https://doi.org/10.1007/978-3-319-67621-0_33 (date of access:20.11.2024).

54. Фішинг та цільовий фішинг, поради по захисту. TechRepublic. URL: https://www.imena.ua/blog/phishing-and-target-phishing (дата зверення:21.11.2024).

55. Модель аналізування уразливостей соціотехнічних систем до впливів соціальної інженерії / Цуркан О., та ін. 4-8 перероб., та доп. Одеса: Наукове видання, 2020. Т.4 (8), 165–371 с.

56. Модель аналізу стратегій при динамічній взаємодії учасників фішингових атак/ Катерина В. та ін. 5-те вид., перероб. та допов. Київ: Наука, 2019. 124–238 с.

57. Werner, G., Yang, S., McConky, K. Time series forecasting of cyber attack intensity. Proceedings of the 12th Annual Conference on Cyber and Information Security Research 2022. Vol. 32, no. 12. P. 389-591.

58. Конкурси та навчальні матеріали з машинного навчання. Kaggle. URL: https://www.kaggle.com/ (дата звернення:25.11.2024).

59. Політика інформаційної безпеки в банку. URL https://static.privatbank.ua/files/file.pdf (дата звернення 28.11.2024).

60. NetworkX Python library. Software for complex networks. URL: https://networkx.org/ (date of access 28.11.2024).

61. Сайт Держаної служби спеціального зв'язку та захисту інформації України. URL: https://cip.gov.ua/ua (дата звернення: 28.11.2023).

62. Аналіз, тренди та рекомендації для клієнтів. Шахрайські та фішингові сайти. URL: http://surl.li/qcjmcu (дата звернення: 29.11.2024).

63. A Recent Comprehensive Study and a New Anatomy. Phishing Attacks. URL: https://urlzs.com/Tb2Ls (date of access: 29.11.202401).

64. An Explainable Feature Selection Framework for Web Phishing Detection with Machine Learning. Data Science and Management. URL: https://urlzs.com/2SN5R (date of access.29.11.2024).

65. Кравченко, В., Руденко, О., Доманов, І. Аналіз фішинг-атак, дослідження методів запобігання та захисту. Збірник наукових праць науково-дослідного інституту, 2022. Т.11 №1, С. 85-95.

66. Design and development of classifiers Naive Bayes. North Dakota State University. URL: https://library.ndsu.edu. (date of access 29.11.2024).

67. Сперкач М. О., Юзьвак Д. Ю. Розв’язання задачі класифікації текстів методами обробки природньої мови та машинного навчання. Створення алгоритмів 2018. Т. 17, № 1. С. 121-280.

68. Scikit-learn Machine Learning in Python. SKLearn site. URL: https://scikitlearn.org/stable/index.html (date of access 01.12.2024).

69. Вишковський Д.П., Гурман І.В., Сотніков Є.О. Штучний інтелект у протидії фішинговим атакам в сфері банківської справи . . Військова освіта і наука: сьогодення та майбутнє : зб. тез доповідей XX Міжнародної науково-практичної конференції. Київ : Військовий інститут Київського національного університету імені Тараса Шевченка, 2024. С.39-40.

ДОДАТОК А

Копії наукових публікацій

13. Підвищення живучості ОУ та САУ в цілому при нерегламентованих впливах на них зовнішнього середовища та виходу з ладу елементів системи.

Наведений тут перелік цілей використання САУ з НРВК, швидше за все, не є повним. Однак навіть він свідчить про досить велику кількість значущих для практики корисних ефектів, які можуть бути забезпечені за рахунок використання САУ з НРВК.

Вишковський Д.П. (ХмНУ)
к.т.н., доц. Гурман І.В. (ХФЕТК УЕП)
Сотніков Є.О. (ВІКНУ)

ШТУЧНИЙ ІНТЕЛЕКТ У ПРОТИДІЇ ФІШИНГОВИМ АТАКАМ В СФЕРІ БАНКІВСЬКОЇ СПРАВИ

Фішингові атаки становлять значну загрозу для банківського сектору, оскільки вони спрямовані на крадіжку конфіденційної інформації клієнтів і банківських установ. Фішингові атаки зазвичай здійснюються через фальшиві електронні листи, повідомлення в соцмережах або SMS, які маскуються під офіційні повідомлення банків. Використовуючи техніки соціальної інженерії, у таких повідомленнях шахраї закликають користувачів перейти за посиланням і ввести свої дані, погрожуючи заблокувати акаунт або доступ до послуг.

Успішні реалізації фішингових атак у банківській сфері можуть мати різні негативні наслідки: компрометація особистих даних клієнтів; фінансові втрати клієнтів; репутаційні збитки для банків; загроза для інформаційної безпеки банку; ризик поширення шкідливого програмного забезпечення і лавиноподібного зростання негативних наслідків тощо.

Захист від фішингових атак стає фундаментом для надійності банківської сфери і саме активні дії банків допомагають мінімізувати ризики та підтримувати довіру клієнтів. Інвестиції в кібербезпеку, навчання та сучасні технології стають невід'ємною частиною стратегії захисту кожного банку.

Серед основних заходів банківських установ для протидії фішинговим атакам можна зазначити багатофакторну автентифікацію, навчання клієнтів і співробітників, фільтри електронної пошти, захист від спаму, моніторинг транзакцій та аномальної активності; захист від підробки вебсайтів.

Потенційно нові можливості надає розробка та впровадження засобів на основі штучного інтелекту (ШІ). Можливості ШІ дозволяють банкам виявляти та блокувати фішингові атаки, підвищуючи рівень захисту та довіри клієнтів. Інструменти на основі штучного інтелекту здатні виявляти аномалії у поведінці користувачів і блокувати підозрілі дії. Завдяки ШІ, банки можуть автоматично фільтрувати та блокувати підозрілі дії, що знижує навантаження на служби безпеки. ШІ працює в режимі реального часу, миттєво реагуючи на потенційні загрози. Це особливо важливо, коли мова йде про швидкі фішингові атаки, які можуть завершитися за лічені хвилини. Системи ШІ здатні навчатися, відстежуючи нові тенденції та техніки, які використовують зловмисники. Алгоритми машинного навчання здатні адаптуватися до нових

схем фішингу та аналізувати величезну кількість даних, що допомагає передбачати та запобігати атакам. Це дозволяє банкам бути на крок попереду навіть найскладніших фішингових схем. Завдяки автоматизації на основі ШІ значно знижується ризик людської помилки, а захист стає більш ефективним. Використовуючи ШІ для захисту від фішингу, фінансові установи отримають ефективний і проактивний підхід у кібербезпеці, що відповідає сучасним викликам цифрового середовища.

к.т.н. Джулій В.М. (ХмНУ)
д.т.н., проф. Лєнков С.В. (ВІКНУ)
Купчик Н.С. (ХмНУ)
Чорненький С.В. (ХмНУ)

ПРОБЛЕМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ

Інформаційно-телекомунікаційні мережі (ІТКМ) забезпечують практично повний спектр можливостей для обміну інформацією між користувачами – мережевими абонентами. Сучасною проблемою таких систем є їх низький рівень інформаційної безпеки. Для забезпечення захисту інформації в телекомунікаційних мережах, включаючи Інтернет, розроблено безліч методів і засобів.

Серед множини функцій захисту, принциповою в відношенні даних систем, є функція попередження прояву забороненої інформації. Вона реалізується за рахунок механізмів прогнозування загрози поширення і розсилання повідомлень з попередженнями про наслідки дій зі забороненим контентом. Використання інших функцій (попередження, виявлення, локалізації та ліквідації загрози) припускає наявність повного контролю над системою, що в реальних умовах неможливо.

Одним з підходів до прогнозування загрози поширення забороненої інформації є моделювання, наприклад, з використанням моделей впливу, моделей просочування і зараження. Дані моделі, як правило, не враховують топологічні особливості мережі (розподіл ступенів зв'язності, кластерний коефіцієнт, середня довжина шляху). Взаємодія між абонентами в межах цих моделей описується переважно гомогенним графом, що при моделюванні великомасштабних мереж може дати похибку прогнозування загрози поширення забороненої інформації більше 30%. Крім того, дані підходи мають в основному теоретичний характер, практика їх використання не виходить за межі експериментів. Таким чином, дослідження, спрямовані на створення моделей та алгоритмів загрози поширення забороненої інформації, актуальні і мають теоретичне і практичне значення у вирішенні проблеми забезпечення інформаційної безпеки в системах і мережах телекомунікацій.

Проведене дослідження проблем інформаційної безпеки виявило, що крім проблем, пов'язаних з використанням глобальної мережі Інтернет як розподіленої інформаційно-телекомунікаційної системи, які досить добре

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Вишковського Дениса Петровича
ПІБ здобувача вищої освіти

Студента ФІТ, 2 курсу, групи КБЗІм-23-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-обчислювального комплексу StrikePlagiarism та/або програмнотехнічного засобу Anti-Plagiarism і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення текстових збігів в роботах.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

2.12.2024
дата

підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 8%**

	Документ		Сумарний збіг по Базі Даних	
ID: 160792 Назва: Метод захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи Додано в БД: 2024-12-18 Автора: Вишковський Денис Керівники: Чешун В.М. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	98900	1432	2541 (3%)	27 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Денис Вишковський

Співавтор:

Назва: Метод захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи

Експерт: Віктор Чешун

Підрозділ: Кафедра кібербезпеки

Коефіцієнт подібності 1:2.3%

Коефіцієнт подібності 2:0.4%

Мікропробіли: 0

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2024-12-18 07:40:39.0

Після аналізу Звіту подібності констатую наступне:

☐ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

Дата 18.12.2024 експерт

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи

Автор: Вишковський Денис Петрович

Спеціальність: 125 – Кібербезпека та захист інформації

Освітня програма:Кібербезпека та захист інформації

Науковий керівник:Віктор ЧЕШУН, канд. техн. наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою StrikePlagiarism складає 99%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism складає 98,6%.

Згідно з правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 24.09.2024, авторська робота, обсяг оригінального тексту у відсотках до загального обсягу матеріалу в якій складає 90-100%, визначається роботою з високою унікальністю тексту і допускається до захисту.

Керівник роботи _____ Віктор ЧЕШУН

Гарант ОП _____ Віра ТІТОВА

Завідувач кафедри кібербезпеки _____ Юрій КЛЬОЦ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «магістр»

Студент Вишковський Денис Петрович

Тема Метод захисту від фішингових атак та несанкціонованого доступу у сфері банківської справи

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека та захист інформації

Обсяг кваліфікаційної роботи освітнього ступеня «магістр»:

кількість листів креслень - ; кількість сторінок записки 99

1. Короткий зміст роботи та прийнятих рішень Кваліфікаційна робота присвячена мінімізації ризику успішних фішингових атак через автоматичне виявлення та блокування підозрілих електронних листів. Таким чином, робота вирішує актуальну проблему захисту банківських клієнтів, використовуючи інноваційний підхід до аналізу тексту з акцентом на реальну практичну застосовність

2. Висновок про відповідність кваліфікаційної роботи завданню Кваліфікаційна робота відповідає поставленому завданню як в теоретичній, так і в практичній частині.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подана загальна характеристика поставленої задачі, чітко визначено об'єкт, предмет та методи дослідження, сформульована актуальність; визначені задачі, які необхідно вирішити для досягнення поставленої мети, практична цінність отриманих результатів, їхня новизна та наведені відомості про публікації. У першому розділі проведене дослідження теоретичної бази понять загроз фішингу та існуючі методи захисту. В другому розділі проведено детальний системний аналіз клієнтського алгоритму виявлення фішингових електронних листів, розглянуто передові розробки у цій сфері, а також викладено теоретичні вимоги до створення нового алгоритму. В третьому розділі проведено аналіз процесу тестування запропонованого алгоритму виявлення шкідливих компонентів у електронних листах, включаючи підготовку тестового середовища, перевірку здатності алгоритму ідентифікувати шкідливі посилання та вкладення. Четвертий розділ Розділ присвячений реалізації запропонованого алгоритму, охопив важливі етапи розробки, тестування та оцінки ефективності, що дозволило створити ефективний інструмент для розв'язання поставленого завдання. .

4. Позитивні сторони роботи Кваліфікаційна робота має наукову і практичну цінність. Запропоновано клієнтський алгоритм виявлення фішингу електронної пошти, що має такі переваги над іншими алгоритмами як локальне виконання – усі обчислення виконуються на пристрої клієнта, що забезпечує приватність і знижує залежність від серверної інфраструктури. Naive Bayes є обчислювально легким алгоритмом, дозволяє швидко обробляти вхідні листи навіть на пристроях з обмеженими ресурсами. Модель можна оновлювати, додаючи нові шаблони фішингових атак. Алгоритм легко інтегрується в банківські клієнтські програми.

5. Негативні сторони роботи В роботі відсутня інформація про апробацію запропонованого методу моделюванням або в реальних умовах.

6. Оцінка графічного оформлення та пояснювальної записки роботи Оформлення всіх матеріалів кваліфікаційної роботи є якісним, здійснене з дотриманням актуальних стандартів та інституційних положень ХНУ. Пояснювальна записка відповідає нормам щодо її оформлення як за структурою, так і за представленням і форматуванням матеріалу.

7. Відгук про роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Весь матеріал кваліфікаційної роботи структурований, чіткий та наскрізно пов'язаний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики кваліфікаційної роботи. Презентаційний та ілюстративний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої мети.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)
Мартинюк Валерій Володимировович
завідувач кафедри АКІТР, доктор технічних наук, професор

« 18 » 12 2024. (підпис)