Машинне навчання як ключовий інструмент оборонних кібероперацій: ефективність виявлення фішингових загроз

Machine learning as a key tool in defensive cyber operations: the effectiveness of phishing threat detection

Надія Бурова [A]
Corresponding author бакалавр, студент, e-mail: nadiia.burova.mKBUI.2023@lpnu.ua, ORCID: 0009-0008-6539-6743

Роман Оприск [A]
бакалавр, студент, e-mail: roman.oprysk.mKBUI.2023@lpnu.ua, ORCID: 0009-0009-5646-2718

Євгеній Курій [A]
доктор філософії, асистент, e-mail: yevhenii.o.kurii@lpnu.ua, ORCID: 0000-0002-3423-5655

Юрій Лах [A]
кандидат фізико-математичних наук, доцент, e-mail: yurii.v.lakh@lpnu.ua, ORCID: 0000-0003-4153-8125

Віталій Сусукайло [A]
доктор філософії, асистент, e-mail: vitalii.a.susukailo@lpnu.ua, ORCID: 0000-0003-4431-9964

Nadiia Burova [A]
Corresponding author Bachelor, Student, e-mail: nadiia.burova.mKBUI.2023@lpnu.ua, ORCID: 0009-0008-6539-6743

Roman Oprysk [A]
Bachelor, Student, e-mail: roman.oprysk.mKBUI.2023@lpnu.ua, ORCID: 0009-0009-5646-2718

Yevhenii Kurii [A]
Doctor of Philosophy, Assistant, e-mail: yevhenii.o.kurii@lpnu.ua, ORCID: 0000-0002-3423-5655

Yuriy Lakh [A]
Candidate of Physical and Mathematical Sciences, Associate Professor, e-mail: yurii.v.lakh@lpnu.ua, ORCID: 0000-0003-4153-8125

Vitalii Susukailo [A]
Doctor of Philosophy, Assistant, e-mail: vitalii.a.susukailo@lpnu.ua, ORCID: 0000-0003-4431-9964

[A] Національний університет Львівська політехніка, м. Львів, Україна
[A] Lviv Polytechnic National University, Lviv, Ukraine

Received: October 18, 2024 | Revised: October 28, 2024 | Accepted: October 31, 2024

DOI: 10.33445/sds.2024.14.5.11

Мета роботи: визначити ефективність використання алгоритмів машинного навчання для виявлення фішингових загроз у межах оборонних кібероперацій.

Метод дослідження: використання алгоритмів випадкових лісів, логістичної регресії та методу опорних векторів для автоматизованого аналізу URL-адрес. Програма реалізована на мові Python з використанням фреймворку Flask.

Результати дослідження: розроблене програмне рішення виявило високу ефективність у виявленні фішингових посилань, продемонструвавши точність аналізу при тестуванні на наборах реальних даних.

Практична цінність дослідження: запропонована система може бути впроваджена як частина оборонних кібероперацій для автоматизованого виявлення шкідливих посилань та підвищення кібербезпеки.

Тип статті: теоретичний, практичний.

Purpose: to determine the effectiveness of using machine learning algorithms for detecting phishing threats within the scope of defensive cyber operations.

Method: utilization of random forest algorithms, logistic regression, and supporting vector machines for automated URL analysis. The program is implemented in Python using the Flask framework.

Findings: the developed solution demonstrated high effectiveness in detecting phishing links, showcasing accuracy in analysis when tested on real data sets.

Practical implications: the proposed system can be implemented as part of defensive cyber operations for automated detection of malicious links and enhancement of cybersecurity.

Papertype: theoretical, practical.

Ключові слова: кібербезпека, машинне навчання, фішинг, оборонні операції, аналіз URL.

Key words: cybersecurity, machine learning, phishing, defensive operations, URL analysis.

Вступ

Кібербезпека стає однією з найважливіших складових сучасної глобальної безпеки, особливо в контексті збройних конфліктів. Кібероперації, спрямовані на дестабілізацію державної і приватної інфраструктури, перетворилися на невід'ємний елемент гібридної війни. Фішингові атаки, що використовують методи соціальної інженерії для обману користувачів і отримання несанкціонованого доступу до інформації, є однією з найпоширеніших форм таких кібератак. В умовах сучасного конфлікту між Україною та Росією ці атаки набули нового значення, ставши

одним із основних інструментів для впливу на критичні системи та підривання оборонних можливостей.

Використання алгоритмів машинного навчання в оборонних кібероперацiях пропонує нові можливості для протидії фішинговим загрозам. Завдяки здатності цих алгоритмів автоматизувати аналіз даних та оперативно виявляти потенційні загрози, його інтеграція у системи кіберзахисту забезпечує більш ефективну та проактивну оборону. У цій статті розглядаються методи використання машинного навчання для перевірки посилань з метою запобігання фішингових атак, а також описуються практичні аспекти впровадження оборонних кібероперацій для підвищення стійкості до таких загроз.

Теоретичні основи дослідження

Фішинг був і залишається однією з найбільш поширених та ефективних тактик для проведення кібератак. Ці атаки націлені на обман користувачів з метою викрадення їхніх облікових даних, отримання доступу до конфіденційної інформації або встановлення шкідливого програмного забезпечення на їхніх пристроях.

Кібератаки з використанням фішингу проти України активно застосовувалися ще до повномасштабного вторгнення Росії в 2022 році та залишаються важливою частиною її кіберкампаній (Mueller та інші, 2023). Наприклад, на початку вторгнення у лютому 2022 року хакери, пов'язані з російськими та білоруськими урядами, організували фішингові атаки, націлені на українських держслужбовців, військових і навіть міжнародні організації, які координували допомогу біженцям (Lewis, 2022). Одна з таких атак включала використання зламаних електронних поштових скриньок, що ускладнювало її виявлення та підвищувало ймовірність успіху (Security Insider, 2022).

Під час війни, окрім традиційного використання фішингу для збору облікових даних, атакуючі часто застосовують шкідливі програми типу “вайпера” (wiper), такі як HermeticWiper та IsaacWiper (Fendorf та інші, 2022). Ці програми були спеціально розроблені для знищення даних на українських урядових та корпоративних системах, що ускладнювало роботу державних органів і знижувало можливості ефективного управління кризовими ситуаціями (Freedberg, 2023). Використання фішингу для поширення таких програм стало однією з найбільш небезпечних форм кібератак, оскільки дозволяло швидко отримувати доступ до критичних систем і розгортати руйнівні дії.

Оборонні кібероперації відіграють важливу роль у протидії фішинговим атакам, особливо під час військових конфліктів, коли кіберпростір стає ключовим елементом бойових дій. Такі операції спрямовані на виявлення, пом’якшення наслідків та запобігання шкідливим активностям у кіберпросторі (Huskaj, 2023) (Daniel, 2022) . В Україні оборонні кібероперації стали важливим інструментом у відбитті російських кібератак. Оборонні кібероперації виконують важливі функції у контексті протидії фішинговим атакам. Вони забезпечують ефективне виявлення загроз, де машинне навчання та інші інструменти використовуються для аналізу даних про підозрілі URL-адреси, поведінкові патерни користувачів і активність у мережі. Це дає змогу виявляти фішингові атаки на ранніх етапах та запобігати їх поширенню (Porche та інші, 2011).

Машинне навчання є однією з ключових технологій, яка дозволяє значно підвищити ефективність оборонних кібероперацій. Завдяки здатності до автоматизації аналізу даних, алгоритми машинного навчання можуть ідентифікувати аномалії та виявляти загрози швидше та точніше, ніж традиційні методи. Це особливо важливо у протидії фішинговим атакам, де зловмисники використовують нові техніки для обходу стандартних засобів захисту. Переваги використання машинного навчання в кібербезпеці включають здатність алгоритмів адаптуватися до змін у тактиці атак та виявляти нові загрози, які раніше не були відомі.

Існує кілька типів алгоритмів машинного навчання, що можуть бути застосовані для аналізу URL-адрес, таких як класифікація, кластеризація та регресія. Класифікаційні алгоритми, як-от “випадкові ліси” та логістична регресія, дозволяють моделювати ймовірність того, що URL є фішинговим. Кластеризаційні алгоритми, такі як метод опорних векторів, допомагають групувати дані на основі схожих ознак, що може бути корисним для виявлення нових фішингових кампаній (Sarker, 2024).

Варто зазначити, що ефективність алгоритмів машинного навчання залежить від якості та кількості даних, використаних для навчання моделей. Використання наборів даних, які включають як фішингові, так і безпечні URL-адреси, дозволяє забезпечити високу точність моделей. У той же час, застосування додаткових ознак, таких як аналіз метаданих, оцінка довжини домену, кількість субдоменів і валідність SSL-сертифікатів, значно підвищує здатність алгоритмів розпізнавати шкідливі посилання (Kapan та інші,2023).

Крім того, варто враховувати складність сучасних фішингових атак, де кіберзлочинці використовують техніки соціальної інженерії та намагаються імітувати легітимні веб-сайти. Це робить традиційні методи захисту менш ефективними, а застосування алгоритмів машинного навчання – необхідністю. У такому випадку системи на базі ШІ можуть аналізувати не лише URL, але й вміст сторінки, що дозволяє виявляти фішингові сайти, навіть якщо вони використовують легітимний SSL-сертифікат або виглядають автентично.

Машинне навчання варто активно застосовувати для перевірки посилань на фішинг, що значно підвищить ефективність оборонних кібероперацій. Використання алгоритмів машинного навчання дозволить автоматизувати аналіз URL-адрес і виявляти потенційні загрози в режимі реального часу. Системи на основі алгоритмів машинного навчання можуть запобігати фішинговим атакам, аналізуючи URL на наявність фішингових ознак. Вони здатні виявляти навіть добре замасковані фішингові сайти, які використовують техніки соціальної інженерії для обману користувачів. Таким чином, машинне навчання стає критично важливим компонентом оборонних кібероперацій, який допомагає не тільки реагувати на кіберзагрози, але й запобігати їм, що забезпечує більш проактивний підхід до захисту в сучасному кіберпросторі.

Постановка проблеми

Сучасні кіберзагрози стають дедалі складнішими, і фішингові атаки відіграють важливу роль у здійсненні шкідливої діяльності в кіберпросторі. Традиційні методи захисту від фішингу часто виявляються недостатньо ефективними, оскільки зловмисники використовують новітні техніки соціальної інженерії та швидко адаптуються до наявних засобів захисту. У зв'язку з цим актуальною є розробка та впровадження інноваційних методів, таких як використання ШІ, для виявлення і запобігання фішинговим атакам на ранніх стадіях.

Методологія дослідження

У цій роботі використані такі методи дослідження:

1. Аналіз наукових джерел: Огляд літератури, наукових статей та доповідей, що стосуються оборонних кібероперацій, використання алгоритмів машинного навчання у кібербезпеці, а також фішингових атак, дозволить отримати загальне розуміння поточного стану досліджень у цій галузі.

2. Систематизація та узагальнення інформації: Класифікація існуючих методів боротьби з фішингом та застосування алгоритмів машинного навчання для аналізу URL-адрес дозволяє структурувати знання з цієї тематики та визначити ключові напрями подальших досліджень.

3. Аналіз випадків фішингових атак проти України: Розгляд реальних прикладів використання фішингу під час російсько-української війни для оцінки ефективності різних методів захисту та виявлення слабких місць в існуючих системах.

Результати та обговорення

Для ефективної реалізації оборонних кібероперацій пропонується застосування інноваційного підходу до виявлення фішингових загроз, який включає розробку програмного забезпечення на основі штучного інтелекту. Цей підхід забезпечує автоматизований аналіз URL-адрес для виявлення потенційно небезпечних посилань, дозволяючи оперативно реагувати на нові загрози. Використання алгоритмів машинного навчання у рамках оборонних операцій підвищує точність і швидкість аналізу, що сприяє зниженню ризику успішних фішингових атак та забезпеченню проактивного захисту критичних систем.

Підкреслюється необхідність глибокого та ретельного аналізу. В основу програмної реалізації входить використання мови програмування Python з закладеним в неї фреймворком Flask для розробки веб-додатку, а також необхідних бібліотек для реалізації аналізу URL за допомогою комплексу алгоритмів машинного навчання та додаткових інструментів.

Аналіз включає в себе використання:

- Алгоритму випадкових лісів;
- Алгоритму логістичної регресії;
- Алгоритму опорно-векторного кластерування на основі методу опорних векторів;

Одним із початкових кроків структури машинного навчання є безпосередньо тренування моделі. Для демонстрації працездатності роботи цілком достатньо невеликого набору даних.

У реалізації програми звертається увагу на те, що фішингові посилання можуть містити в назві IP-адресу, що з високою ймовірністю свідчить про небезпеку такого URL. Для цього випадку передбачається функція, яка за допомогою регулярних виразів аналізує назву URL-адреси та повертає значення “1”, якщо в її назві виявлено послідовність символів, які використовуються у IP-адресах (Рис. 1).

```
def has_ip(url):
    pattern = re.compile(r"(?:\d{1,3}\.){3}\d{1,3}")
    return 1 if pattern.search(url) else 0
```

Рисунок 1 – Функція для виявлення наявності IP-адреси в назві URL

Також загально відомо, що веб-сайти використовують HTTPS для безпеки з’єднання. Разом з розвитком безпеки веб-сайтів продовжує розвиватися й світ кіберзлочинців, які навчилися використовувати сертифікати SSL для фішингових сайтів з метою їх маскування під безпечні. На основі цього існує потреба в перевірці URL-адреси на валідність SSL-сертифікату. Для цього створюється функція, що аналізує посилання на наявність та валідність SSL-сертифікату (Рис. 2). Вона дозволяє більш ефективно оцінювати рівень загрози URL-адрес, що підкреслює точність програмної реалізації.

```python
def has_valid_ssl(url):
    try:
        parsed_url = urlparse(url)
        hostname = parsed_url.hostname if parsed_url.hostname else parsed_url.path

        context = ssl.create_default_context()
        with socket.create_connection((hostname, 443)) as sock:
            with context.wrap_socket(sock, server_hostname=hostname) as ssock:
                ssl_info = ssock.getpeercert()

        return 1
    except Exception as e:
        return 0
```

Рисунок 2 – Функція для перевірки валідності SSL-сертифікату

Не менш важливим кроком є повне покриття можливих варіантів аналізу посилань на фішингові ознаки. На сьогоднішній день у сфері інформаційних технологій існує багато хороших рішень, проте усі вони лише частково покривають перевірку URL-адрес.

У цьому випадку створюється функція, що допоможе усунути цю проблему. Вона буде вирізнятися кількістю параметрів, згідно, яких буде аналізуватися URL-адреса (Рис.3).

Сюди входять:

- Визначення кількості підозрілих символів в назві URL;
- Перевірка на наявність символів IP-адреси в назві URL;
- Перевірка наявності протоколу HTTPS;
- Перевірка на наявність підозрілого доменного імені та його довжина;
- Перевірка кількості суб-доменів;
- Перевірка валідності SSL-сертифікату;

```python
def extract_features(url):
    suspicious_symbols = ['@', '%', '&', '=']
    has_ip_addr = has_ip(url)
    suspicious_count = sum([url.count(symbol) for symbol in suspicious_symbols])
    uses_https = 1 if url.startswith('https') else 0
    domain_info = tldextract.extract(url)
    domain_length = len(domain_info.domain)
    subdomain_count = len(domain_info.subdomain.split('.')) if domain_info.subdomain else 0
    dangerous_word_count = 1 if has_dangerous_words(url) else 0
    valid_ssl = has_valid_ssl(url)

    return [has_ip_addr, suspicious_count, uses_https, domain_length, subdomain_count, dangerous_word_count, valid_ssl]
```

Рисунок 3 –Функція для перевірки URL-адреси за кількома параметрами

Головною ідеєю програмної реалізації є використання комплексу алгоритмів машинного навчання. Перед безпосередньою його реалізацією важливо підготувати набір даних для подальшого аналізу. Тут мається на увазі розбиття даних з вибірки на навчальні та тестові. Важливим тут є правильне їх розділення. Оскільки, початковий набір даних не є великим і складається з шести URL-адрес, дані рекомендується ділити наступним чином: чотири URL з набору даних входять у тестову вибірку (70%), а два URL у навчальну (30%) (Рис. 4).

```python
features = np.array([extract_features(url) for url in data])
X_train, X_test, y_train, y_test = train_test_split( *arrays: features, labels, test_size=0.3, random_state=42)
```

Рисунок 4 – Підготовка даних

У зв’язку з використанням невеликих наборів даних задаються оптимальні значення для кожного з алгоритмів, що входять в комплекс. Вони не затримують процес виконання програми і разом з тим зумовлюють точні результати аналізу (Рис. 5).

```
model_rf = RandomForestClassifier(n_estimators=100, random_state=42)
model_lr = LogisticRegression(random_state=42)
model_svc = SVC(probability=True, random_state=42)
```

Рисунок 5 – Створення моделей

Для створення комплексу алгоритмів машинного навчання їх потрібно об’єднати. В такому випадку порівняння отриманих результатів відбувається внаслідок виконання кожного алгоритму та автоматичний вибір оптимальної оцінки аналізу (Рис. 6).

```
ensemble_model = VotingClassifier(estimators=[
    ('rf', model_rf), ('lr', model_lr), ('svc', model_svc)
], voting='soft')
```

Рисунок 6 – Створення комплексу алгоритмів

Тренування комплексу алгоритмів машинного навчання відбувається зразу після його створення. Також використовується функція, яка видає кінцевий результат аналізу URL-адреси про те чи є посилання безпечним чи фішинговим в залежності від того, яке було отримане на вхід.

Для більш зручної роботи використовується створений веб-додаток з використанням фреймворку Flask. Його суттю є введення користувачем URL-адреси для аналізу та отримання результатів аналізу і звіту про усі введені посилання.

Звідси виникає логічне твердження, що вже існує схожий веб-додаток під назвою “PhishTank”. В такому разі на прикладі розглядається програмна реалізація аналізу URL-адрес на фішингові ознаки з використанням комплексу алгоритмів машинного навчання.

На початку, користувач бачить інтуїтивну сторінку з можливістю вводу URL-адреси для аналізу. Тут зразу варто зазначити, що інтуїтивність є однією з переваг даної програми у порівнянні з “PhishTank”.

Для порівняння: користувач вводить один безпечний (Рис. 7) та два фішингових сайти.

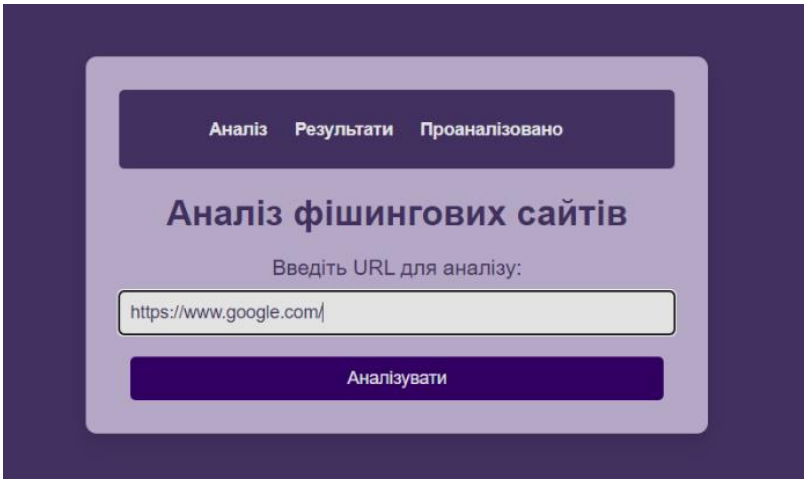

Рисунок 7 – Введення безпечного сайту для аналізу

Після підтвердження свого вибору одразу же можна побачити результат (Рис. 8).

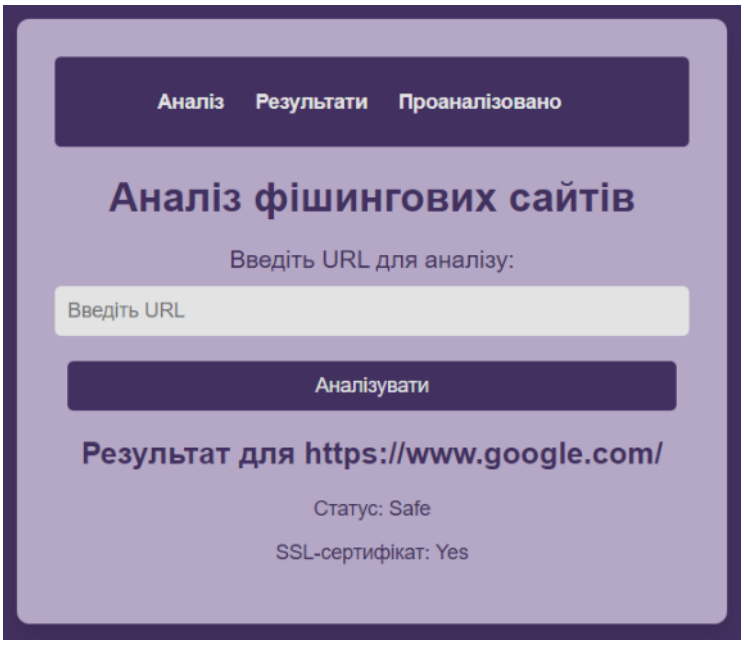

Рисунок 8 – Результат аналізу безпечного сайту

Тут підкреслюється швидкість виконання даної програми та актуальність перевірки на валідність SSL-сертифікату. “PhishTank” безумовно також має можливість вводу посилання для аналізу, проте отримання фінального результату займає тижні або місяці. Оскільки, усі посилання перевіряються людьми, то навіть за умови отримання кінцевого результату, він не є достатньо надійним, адже тут присутній людський фактор.

Така програмна реалізація також надає можливість користувачеві переглянути вже проаналізовані URL-адреси (Рис. 9). Як результат, очікувані результати роботи програми та наявні збігаються, адже було взято одну безпечну та дві фішингові URL-адреси.

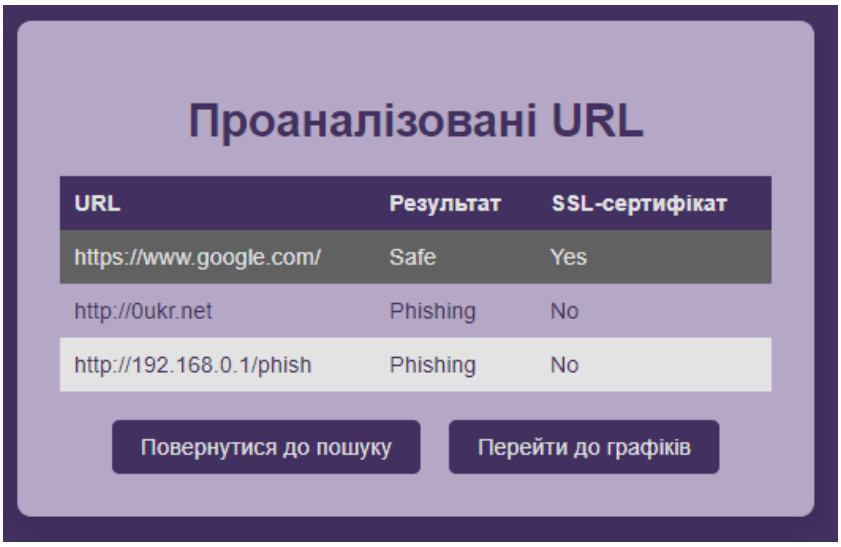

Рисунок 9 – Перегляд проаналізованих URL-адрес

Для вираження точності роботи програмної реалізації використовуються реальні фішингові адреси. Рекомендується використовувати підтверджений чорний список URL-адрес.

З нього обрано 20 фішингових посилань для проведення перевірки надійності результатів програмної реалізації (Рис. 10). Варто врахувати, що всі URL-адреси використовують протокол HTTPS, що ускладнює виявлення ознак фішингу.

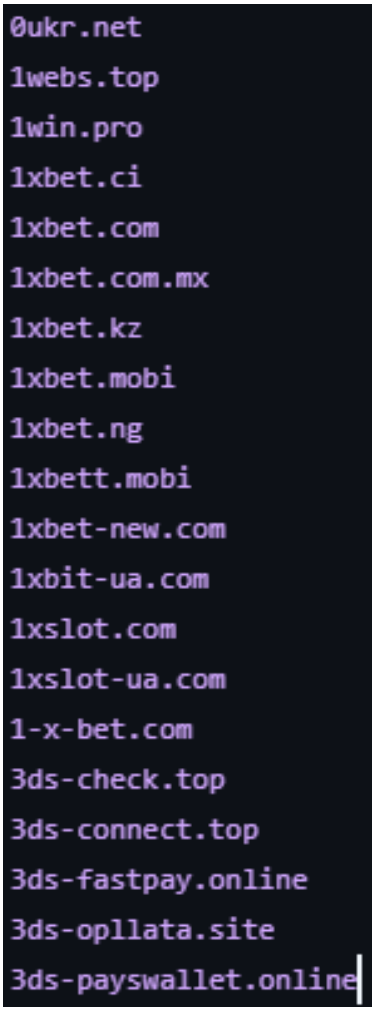

Рисунок 10 – Тестова вибірка чорного списку URL-адрес

Оскільки у програмній реалізації запобігання фішинговим атакам використовується функція перевірки SSL-сертифікату це тестування в свою чергу дає можливість показати її роботу.

Очкікується, що результатом програми, яка використовує комплекс алгоритмів машинного навчання будуть 20 URL-адрес з підтвердженням наявності фішингових ознак та невалідними SSL-сертифікатами.

Після введення усіх URL-адрес із чорного списку та їх перегляду після аналізу спостерігається, що усі посилання були позначені як фішингові, а перевірка SSL-сертифікатів показала, що вони не є валідними (Рис. 11). Таким чином, очікувані результати збігаються з наявними, що підкреслює точність роботи програмної реалізації аналізу URL-адрес на фішингові ознаки.

Проаналізовані URL

URL	Результат	SSL-сертифікат
https://0ukr.net	Phishing	No
https://1webs.top	Phishing	No
https://1win.pro	Phishing	No
https://1xbet.ci	Phishing	No
https://1xbet.com	Phishing	No
https://1xbet.com.mx	Phishing	No
https://1xbet.kz	Phishing	No
https://1xbet.mobi	Phishing	No
https://1xbet.ng	Phishing	No
https://1xbett.mobi	Phishing	No
https://1xbet-new.com	Phishing	No
https://1xbit-ua.com	Phishing	No
https://1xslot.com	Phishing	No
https://1xslot-ua.com	Phishing	No
https://1-x-bet.com	Phishing	No
https://3ds-check.top	Phishing	No
https://3ds-connect.top	Phishing	No
https://3ds-fastpay.online	Phishing	No
https://3ds-opllata.site	Phishing	No
https://3ds-payswallet.online	Phishing	No

Повернутися до пошуку | Перейти до графіків

Рисунок 11 – Демонстрація результатів

Після розгляду програми, можна порівняти її з “PhishTank” по швидкодії, точності та ефективності роботи (Рис. 12).

Програмна реалізація запобігання фішинговим атакам на основі комплексу алгоритмів машинного навчання суттєво виділяється на фоні “PhishTank”, адже ефективний аналіз URL-адрес не потребує великих часових затрат. Точність в свою чергу не залежить від людського фактору, а заснована безпосередньо на надійних алгоритмах машинного навчання, які працюючи в комплексі надають можливість ефективно запобігати фішинговим атакам.

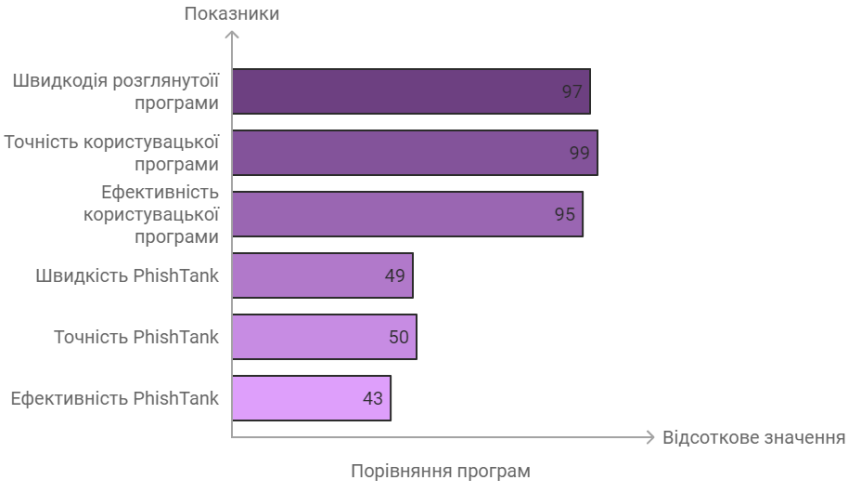

Рисунок 12 – Порівняльний графік

Висновки

Розробка і впровадження систем для автоматизованого виявлення фішингових атак є важливою складовою сучасних оборонних кібероперацій. Проведений аналіз показав, що застосування алгоритмів машинного навчання дозволяє підвищити ефективність захисту від фішингових загроз, забезпечуючи швидке та точне виявлення небезпечних URL-адрес. Використання таких алгоритмів, як випадкові ліси, логістична регресія та метод опорних векторів, дозволяє комплексно оцінювати ознаки фішингу, включаючи наявність IP-адрес у посиланні, валідність SSL-сертифікатів та кількість субдоменів.

Успішна реалізація запропонованого підходу підтверджується результатами тестування, де система виявила всі фішингові посилання із чорного списку. Це свідчить про високу точність програмного рішення, що ґрунтується на об'єднанні кількох алгоритмів машинного навчання. Додаткові переваги запропонованої системи включають її швидкодію та відсутність залежності від людського фактору, що є значним покращенням порівняно з існуючими рішеннями, такими як "PhishTank".

Подальший розвиток цієї програми передбачає збільшення набору даних для тренування моделей, а також впровадження нових методів аналізу, таких як глибоке навчання. Це дозволить адаптувати систему до змінних умов кіберзагроз та забезпечити її ефективність на довгострокову перспективу. Результати дослідження підтверджують, що інтеграція машинного навчання в оборонні кібероперації є перспективним напрямком для зміцнення кібербезпеки та протидії фішинговим атакам.

Фінансування

Це дослідження не отримало конкретної фінансової підтримки.

Конкуруючі інтереси

Автори заявляють, що у них немає конкуруючих інтересів.

Список використаних джерел

1. Mueller, G. B., Jensen, B., Valeriano, B., Maness, R. C., & Macias, J. M. (2023, 13 липня). *Cyber operations during the Russo-Ukrainian war*. Center for Strategic and International Studies. URL : https://www.csis.org/analysis/cyber-operations-during-russo-ukrainian-war
2. Lewis, J. A. (2022, 16 червня). *Cyber war and Ukraine*. Center for Strategic and International Studies. https://www.csis.org/analysis/cyber-war-and-ukraine
3. Russia's cyberattack activity in the Ukraine / *Security Insider*. (2022). URL : https://www.microsoft.com/en-us/security/security-insider/intelligence-reports/special-report-ukraine/
4. Freedberg Jr, S. J. (2023, 16 лютого). *Russian phishing attacks flooded Ukraine, tripled against NATO nations in 2022: Report – Breaking Defense*. Breaking Defense. URL : https://breakingdefense.com/2023/02/russian-phishing-attacks-flooded-ukraine-tripled-against-nato-nations-in-2022-report/
5. Fendorf, K., & Miller, J. (2022, 24 березня). *Tracking cyber operations and actors in the russia-ukraine war*. Council on Foreign Relations. URL : https://www.cfr.org/blog/tracking-cyber-operations-and-actors-russia-ukraine-war
6. Daniel, M. (2022). *Offensive cyber operations: Understanding intangible warfare*. Oxford University Press, Incorporated.
7. Huskaj, G. (2023). Offensive cyberspace operations for cyber security. *International Conference on Cyber Warfare and Security*, *18*(1), 476–479. https://doi.org/10.34190/iccws.18.1.1054

8. Porche, I. R., Sollinger, J. M., & McKay, S. (2011). *A cyberworm that knows no boundaries. RAND Corporation.* https://doi.org/10.7249/op342
9. Sarker, I. H. (2024). Learning technologies: Toward machine learning and deep learning for cybersecurity. У *AI-Driven cybersecurity and threat intelligence: Cyber automation, intelligent decision-making and explainability* (с. 43-59). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-54497-2_3
10. Kapan, S., & Sora Gunal, E. (2023). Improved phishing attack detection with machine learning: A comprehensive evaluation of classifiers and features. *Applied Sciences*, *13*(24). https://doi.org/10.3390/app132413269

References

1. Mueller, G. B., Jensen, B., Valeriano, B., Maness, R. C., & Macias, J. M. (2023, July 13). *Cyber operations during the Russo-Ukrainian war*. Center for Strategic and International Studies. Available from : https://www.csis.org/analysis/cyber-operations-during-russo-ukrainian-war
2. Lewis, J. A. (2022, June 16). *Cyber war and Ukraine*. Center for Strategic and International Studies. https://www.csis.org/analysis/cyber-war-and-ukraine
3. Russia's cyberattack activity in the Ukraine | *Security Insider*. (2022). Available from : https://www.microsoft.com/en-us/security/security-insider/intelligence-reports/special-report-ukraine/
4. Freedberg Jr, S. J. (2023, February 16). *Russian phishing attacks flooded Ukraine, tripled against NATO nations in 2022: Report – Breaking Defense*. Breaking Defense. Available from : https://breakingdefense.com/2023/02/russian-phishing-attacks-flooded-ukraine-tripled-against-nato-nations-in-2022-report/
5. Fendorf, K., & Miller, J. (2022, March 24). *Tracking cyber operations and actors in the russia-ukraine war*. Council on Foreign Relations. https://www.cfr.org/blog/tracking-cyber-operations-and-actors-russia-ukraine-war
6. Daniel, M. (2022). *Offensive cyber operations: Understanding intangible warfare*. Oxford University Press, Incorporated.
7. Huskaj, G. (2023). Offensive cyberspace operations for cyber security. *International Conference on Cyber Warfare and Security*, *18*(1), 476–479. https://doi.org/10.34190/iccws.18.1.1054
8. Porche, I. R., Sollinger, J. M., & McKay, S. (2011). *A cyberworm that knows no boundaries. RAND Corporation.* https://doi.org/10.7249/op342
9. Sarker, I. H. (2024). Learning technologies: Toward machine learning and deep learning for cybersecurity. У *AI-Driven cybersecurity and threat intelligence: Cyber automation, intelligent decision-making and explainability* (S. 43-59). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-54497-2_3
10. Kapan, S., & Sora Gunal, E. (2023). Improved phishing attack detection with machine learning: A comprehensive evaluation of classifiers and features. *Applied Sciences*, *13*(24). https://doi.org/10.3390/app132413269