# МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 38.00.00.000 ПЗ

**Група ІПм-24-2**

**Міхнович Михайло**

**2025**

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

**Міхнович Михайло Вікторович**

(прізвище, ім'я, по батькові)

УДК 004.9

(індекс)

# МАГІСТЕРСЬКА РОБОТА

**Засоби, методи та інструменти протидії мережевим атакам із застосуванням методологій машинного навчання**

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Міхнович М.В.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Мельник Віталій Дмитрович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

**Допущено до захисту**

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

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

Івано-Франківськ – 2025

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ:**
Зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ 04 ” вересня 2025 р.

# ЗАВДАННЯ

## НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

**Міхновичу Михайлу Вікторовичу**

(прізвище, ім’я, по-батькові)

**1. Тема магістерської роботи “ Засоби, методи та інструменти протидії мережевим атакам із застосуванням методологій машинного навчання”**

керівник проекту (роботи) Мельник Віталій Дмитрович, к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 05 ” листопада 2025 р. № 695/7

**2. Строк подання студентом проекту (роботи)** 15 грудня 2025 р.

**3. Вихідні дані до проекту (роботи)** Концепції, формальні моделі протидії мережевим атакам із застосуванням методологій машинного навчання

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

1. Аналіз предметної області дослідження та класифікації мережевих атак, загроз і вразливостей
2. Аналіз фішингових атак та поведінкових аспектів кібербезпеки
3. Дослідження методології виявлення шкідливого ПЗ на основі алгоритмів машинного навчання
4. Імплементація методів та інструментів протидії мережевим атакам та фішингу на основі МН

**5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)**

1. Ілюстрація алгоритму генерації доменів (рис. 1.1)
2. Доменні імена, згенеровані алгоритмами генерації доменів (рис. 1.2)
3. Спрощений формат кадру EAPOL (рис. 1.3)
4. Приклад фішингового електронного листа (рис. 1.4)
5. Фішинговий лист (імітація служби технічної підтримки) (рис. 1.5)

**6. Консультанти розділів проекту (роботи)**

| Розділ | Консультант | Підпис, дата |
|---|---|---|
| Перевірка на плагіат | доц., к.т.н. Вовк Р.Б. | |
| | | |

**7. Дата видачі завдання** 04 вересня 2025 р.

**Керівник** _____
(підпис)

**Завдання прийняв до виконання**_____
(підпис)

# КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назви етапів магістерської роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Підбір і вивчення літератури по темі магістерської роботи | 15.09.2025 | виконано |
| 2 | Аналіз предметної області дослідження та класифікації мережевих атак, загроз і вразливостей | 01.10.2025 | виконано |
| 3 | Аналіз фішингових атак та поведінкових аспектів кібербезпеки | 17.10.2025 | виконано |
| 4 | Дослідження методології виявлення шкідливого ПЗ на основі алгоритмів машинного навчання | 02.11.2025 | виконано |
| 5 | Імплементація методів та інструментів протидії мережевим атакам та фішингу на основі МН | 19.11.2025 | виконано |
| 6 | Аналіз продуктивності та оптимізація глибокого навчання | 02.12.2025 | виконано |
| 7 | Затвердження пояснювальної записки роботи завідувачем кафедри | 15.12.2025 | виконано |

**Студент – магістр** _____
(підпис)

**Керівник роботи** _____
(підпис)

# АНОТАЦІЯ

**Магістерська робота:** 78 с., 25 рис., 10 табл., 38 джерел.

**Тема:** Засоби, методи та інструменти протидії мережевим атакам із застосуванням методологій машинного навчання

**Мета роботи:** розроблення та дослідження методів, моделей і фреймворку виявлення та протидії мережевим атакам із застосуванням методів машинного.

**Об'єкт дослідження:** процеси виявлення, класифікації та запобігання мережевим атакам з використанням технологій машинного навчання.

**Предмет дослідження:** методи, засоби та інструменти протидії мережевим атакам, зокрема атакам, що реалізуються через алгоритми генерації доменів та фішингові механізми, із використанням алгоритмів машинного та глибокого навчання.

**Результати дослідження**

В роботі розроблено, теоретично обґрунтовано і практично перевірено підхід до підвищення ефективності протидії мережевим атакам із використанням технологій машинного навчання.

**Висновок**

У ході дослідження здійснено аналіз продуктивності моделей, налаштування гіперпараметрів і оптимізацію процесу навчання. Отримані результати свідчать, що застосування глибоких нейронних мереж дозволяє знизити кількість помилкових спрацьовувань, підвищити точність класифікації.

**МЕРЕЖЕВА БЕЗПЕКА; МАШИННЕ НАВЧАННЯ; ГЛИБОКЕ НАВЧАННЯ; АЛГОРИТМИ ГЕНЕРАЦІЇ ДОМЕНІВ; ФІШИНГ; КЛАСИФІКАЦІЯ; КЛАСТЕРИЗАЦІЯ; НЕЙРОННА МЕРЕЖА; ФРЕЙМВОРК КІБЕРЗАХИСТУ.**

# ABSTRACT

**Master Thesis:** 78 pp., 25 fig., 10 tab., 38 sources.

**Topic:** Means, methods and tools for countering network attacks using machine learning methodologies

**Purpose of the work**: development and research of methods, models and framework for detecting and countering network attacks using machine learning methods.

**Object of the research**: processes of detecting, classifying and preventing network attacks using machine learning technologies.

**Subject of the research**: methods, means and tools for countering network attacks, in particular attacks implemented through domain generation algorithms and phishing mechanisms, using machine and deep learning algorithms.

**Research results**

The paper developed, theoretically substantiated and practically tested an approach to increasing the effectiveness of countering network attacks using machine learning technologies.

**Conclusion**

During the study, an analysis of model performance, hyperparameter tuning and optimization of the learning process were carried out. The results obtained indicate that the use of deep neural networks allows reducing the number of false positives and increasing the accuracy of classification.

**NETWORK SECURITY; MACHINE LEARNING; DEEP LEARNING; DOMAIN GENERATION ALGORITHMS; PHISHING; CLASSIFICATION; CLUSTERIZATION; NEURAL NETWORK; CYBER DEFENSE FRAMEWORK.**

**ЗМІСТ**

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

DGA - Domain Generation Algorithms

DNN - Deep Neural Network

HMM- Hidden Markov Model

AUC - Area Under the Curve

FPR - False Positive Rate

J48 - Decision Tree (implementation of the C4.5 algorithm)

NB - Naive Bayes

MP - Multilayer Perceptron

SGD - Stochastic Gradient Descent

IRB - Institutional Review Board

# ВСТУП

**Актуальність теми.**

У сучасних умовах стрімкого розвитку інформаційних технологій та глобальної цифровізації суспільства питання забезпечення кібербезпеки набуває критичного значення. Швидке зростання кількості мережевих сервісів, хмарних платформ, інтернет-пристроїв та обмінів даними створює сприятливе середовище для появи нових типів кіберзагроз, які стають дедалі складнішими, динамічнішими й важчими для виявлення традиційними методами. Мережеві атаки націлені не лише на порушення конфіденційності або доступності інформації, але й на підрив довіри до інформаційної інфраструктури, що має критичні наслідки для економічної та національної безпеки держав.

В умовах постійної еволюції шкідливого програмного забезпечення, появи нових векторів атак і високого рівня автоматизації кіберзлочинності класичні підходи до виявлення та запобігання загрозам — сигнатурні, евристичні чи поведінкові методи — поступово втрачають свою ефективність. У зв’язку з цим виникає потреба у впровадженні інтелектуальних технологій аналізу, що здатні адаптуватися до нових сценаріїв атак, виявляти невідомі загрози та забезпечувати самонавчання систем. Машинне та глибоке навчання відкривають нові можливості у сфері кіберзахисту, дозволяючи виявляти приховані закономірності в даних трафіку, класифікувати шкідливі домени, прогнозувати аномальну активність і формувати адаптивні моделі захисту.

Дослідження, представлене в цій магістерській роботі, спрямоване на систематизацію засобів і методів протидії мережевим атакам та розробку практичного фреймворку, що поєднує алгоритми машинного навчання, кластеризації та прогнозування. Особлива увага приділяється виявленню шкідливого ПЗ, яке використовує алгоритми генерації доменів (DGA), а

також моделюванню фішингових атак із врахуванням поведінкових аспектів користувачів.

Актуальність теми зумовлена стрімким зростанням кількості та складності мережевих атак, які використовують інтелектуальні техніки обходу систем захисту, зокрема генерацію доменів, шифрування трафіку, соціотехнічні впливи та змагальні атаки на моделі штучного інтелекту. Згідно зі статистикою міжнародних аналітичних центрів, понад 70% сучасних атак мають автоматизований або частково інтелектуальний характер, що робить традиційні підходи малоефективними.

Використання методів машинного навчання для виявлення аномалій і класифікації загроз дозволяє створювати адаптивні, самонавчальні системи, здатні протидіяти новим типам атак у режимі реального часу. Особливу небезпеку становлять алгоритми генерації доменів (DGA), які активно застосовуються ботнетами для динамічного створення командних серверів, уникаючи блокування. Не менш критичними залишаються фішингові атаки, що експлуатують поведінкові слабкості користувачів і обходять технічні механізми захисту.

У цьому контексті розробка методології виявлення шкідливого ПЗ та мережевих атак із застосуванням машинного навчання має високу наукову та практичну значущість. Вона сприяє підвищенню ефективності захисту інформаційних систем, забезпеченню надійності критичної інфраструктури та формуванню нової парадигми кібербезпеки, орієнтованої на інтелектуальні аналітичні рішення.

**Метою роботи** є розроблення та дослідження методів, моделей і фреймворку виявлення та протидії мережевим атакам із застосуванням методів машинного.

**Об’єктом дослідження** є процеси виявлення, класифікації та запобігання мережевим атакам з використанням технологій машинного навчання.

**Предметом дослідження** є методи, засоби та інструменти протидії мережевим атакам, зокрема атакам, що реалізуються через алгоритми генерації доменів та фішингові механізми, із використанням алгоритмів машинного та глибокого навчання.

**Завдання дослідження**

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Провести аналіз сучасного стану мережевої безпеки, класифікації загроз, атак і вразливостей у комп'ютерних мережах.

2. Дослідити принципи роботи алгоритмів генерації доменів та існуючі методи їх виявлення.

3. Проаналізувати змагальні атаки на моделі машинного навчання та їхній вплив на безпеку систем.

4. Розробити методологію виявлення шкідливого ПЗ на основі алгоритмів машинного навчання.

5. Імплементувати модель глибокого навчання для класифікації DGA-доменів і поведінкових аномалій.

6. Провести експериментальну оцінку ефективності запропонованих методів та моделей.

**Методи дослідження**

У процесі дослідження використано комплекс наукових методів:

- аналітичні методи — для систематизації сучасних мережевих загроз і класифікації атак;

- математичне моделювання — для формалізації процесів виявлення аномалій у мережевому трафіку;

- методи машинного та глибокого навчання — для побудови моделей класифікації та прогнозування;

- кластерний аналіз (DBSCAN) — для виявлення нових типів атак;

- експериментальне моделювання — для перевірки ефективності запропонованого фреймворку.

**Наукова новизна отриманих результатів**

Запропоновано комплексний фреймворк виявлення мережевих атак на основі глибокого навчання з інтеграцією кластеризації та прогнозування часових рядів. Удосконалено методологію виявлення DGA-доменів шляхом розширення набору ознак, що включають статистичні, лінгвістичні та поведінкові характеристики. Розроблено багаторівневу модель аналізу мережевих загроз, яка поєднує алгоритми класифікації першого рівня, кластеризації другого рівня та прогнозування третього рівня.

**Практичне застосування результатів**

Розроблений фреймворк може бути використаний для побудови систем моніторингу кіберзагроз у корпоративних мережах, платформах кіберзахисту, сервісах DNS-моніторингу та навчальних системах кібербезпеки. Отримані результати можуть бути впроваджені у відомчі та промислові системи кіберзахисту для автоматичного виявлення шкідливої активності, виявлення фішингових кампаній і прогнозування аномалій у трафіку.

**Структура магістерської роботи.** Представлена робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 78 сторінок, і містить 25 рисунків, 10 таблиць, перелік використаних джерел із 38 позицій.

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ ТА КЛАСИФІКАЦІЇ МЕРЕЖЕВИХ АТАК, ЗАГРОЗ І ВРАЗЛИВОСТЕЙ

## 1.1. Аналіз та забезпечення мережевої безпеки в сучасних комп'ютерних системах

### *1.1.1. Актуальність проблематики мережевої безпеки*

Мережева безпека є критично важливим аспектом функціонування сучасних комп'ютерних мереж з моменту виникнення архітектури Інтернету. Гарантії безпеки необхідні для підтримки цілісності, конфіденційності та доступності даних у мережах, що використовуються для корпоративних транзакцій, урядових комунікацій та обміну інформацією між індивідуальними користувачами. Комп'ютерні мережі демонструють вразливість до різноманітних векторів кібератак, які традиційно класифікуються на два основних типи: пасивні та активні.

Пасивні атаки це коли зловмисники обмежуються перехопленням та моніторингом мережевого трафіку без внесення змін до даних чи порушення функціональності системи.

Активні атаки характеризуються здатністю зловмисників до порушення нормального функціонування мережі або несанкціонованого доступу до неї. Успішна активна атака може призвести до несанкціонованого доступу до корпоративних, державних або персональних активів, що потенційно може мати катастрофічні наслідки.

Управління мережевою безпекою є контекстно-залежним процесом, що визначається специфічними вимогами до безпеки, архітектурою мережі, типом загроз та поверхнями атаки. Зважаючи на глибоку інтеграцію Інтернету у повсякденне життя та зростання тенденції до зберігання даних в онлайнових сховищах, зокрема у хмарних середовищах, кількість потенційних поверхонь для експлуатації загроз значно збільшилася. Це вимагає розширених знань про механізми мережевих атак для вдосконалення

стратегій захисту. Швидкий розвиток мережевих технологій та зростання кількості взаємопов'язаних пристроїв створюють ризик каскадного проникнення: компрометація однієї точки доступу може надати зловмиснику плацдарм для проникнення в усю мережу та атаки на підключені пристрої. Отже, забезпечення захисту всіх потенційних поверхонь атаки є критичним імперативом.

### *1.1.2. Вразливості системи доменних імен*

Система доменних імен (DNS) є фундаментальним і життєво важливим компонентом архітектури Інтернету. Вона функціонує як розподілена система найменування для мережевих ресурсів, призначаючи доменні імена кожному підключеному пристрою та виконуючи функцію «телефонної книги» Інтернету, перетворюючи доменні імена у відповідні унікальні IP-адреси.

DNS є об'єктом різноманітних кібератак, включаючи атаки нульового дня, отруєння кешу (cache poisoning) та атаки типу "відмова в обслуговуванні" (DoS/DDoS). Забезпечення безпеки доменної інфраструктури є ключовим для загального захисту мережі.

Доменні імена, зазвичай, є рядками з літер і цифр, що використовуються для ідентифікації мережевого домену. Алгоритм генерації доменів (DGA) являє собою програмний механізм, який періодично генерує велику кількість доменних імен. Ці домени часто використовуються як динамічні точки комунікації для серверів командування та контролю (C2) шкідливого програмного забезпечення. Зловмисники застосовують DGA-базоване шкідливе ПЗ для маскування зловмисного трафіку під нормальний, уникаючи виявлення та отримуючи контроль над мережею. Це відкриває можливості для крадіжки особистої та корпоративної інформації. Виявлення такого шкідливого програмного забезпечення є нетривіальним завданням, оскільки зловмисники постійно змінюють місце розташування своїх C2-

сервісів, а традиційні методи, такі як використання чорних списків, стають неефективними через велику кількість динамічно згенерованих доменів.

### *1.1.3. Вразливості бездротових мереж*

З розвитком комп'ютерних мереж та зростанням потреби у мобільності, бездротові мережі набули широкого поширення. Однак, їх активне використання збільшило поверхню атаки, створюючи нові вектори для крадіжки персональних даних. Wi-Fi використовує бездротовий мережевий протокол на основі стандарту IEEE 802.11x. Сучасні захищені мережі Wi-Fi використовують протокол Wi-Fi Protected Access 2 (WPA2). Зв'язок у Wi-Fi мережах здійснюється за допомогою радіочастотної (RF) технології, де Точка Доступу (AP) є центральним компонентом, що передає бездротовий сигнал для підключення сумісних пристроїв.

Атака повторної встановлення ключа (KRACK) є серйозною атакою, що компрометує мережі Wi-Fi, експлуатуючи критичну вразливість у протоколі WPA2. Ця вразливість дозволяє зловмисникам потенційно викрадати конфіденційну інформацію користувачів, включаючи паролі, електронні листи та банківські реквізити. З огляду на зростаючу кількість мобільних пристроїв (ноутбуки, планшети, смартфони, підключені автомобілі), які використовують Wi-Fi, захист від KRACK є життєво необхідним для забезпечення конфіденційності користувачів.

### *1.1.4. Аналіз та протидія фішинговим атакам*

Фішингова атака є поширеним типом кібератаки, спрямованої на викрадення особистої інформації користувачів, такої як облікові дані, банківські рахунки та паролі. У цьому сценарії зловмисники маскуються під довірені суб'єкти, щоб обманом змусити користувача взаємодіяти зі шкідливим вмістом (наприклад, відкрити електронний лист або текстове повідомлення). Фішингові електронні листи є найбільш поширеним методом маскування через повсюдне використання електронної пошти.

Користувачі, як правило, стимулюються до переходу за шкідливим посиланням, що призводить до розголошення особистих даних, або до завантаження зловмисних вкладень, які встановлюють шкідливе програмне забезпечення. Хоча пильність користувача може допомогти у виявленні та запобіганні фішинговим атакам, значна кількість користувачів перевіряє пошту без належної уваги до деталей (адреса відправника, зміст листа). Для ефективнішого захисту користувачів від фішингу критично важливо дослідити та зрозуміти когнітивно-поведінкові фактори, які впливають на прийняття рішень користувачем під час зіткнення з такими загрозами.

### *1.1.5. Загрози для систем машинного навчання в мережевій безпеці*

З появою та стрімким розвитком машинного навчання та глибокого навчання було досягнуто значного прогресу у сферах комп'ютерного зору, обробки природної мови та розпізнавання мовлення. ML також демонструє високу ефективність у сфері мережевої безпеки, зокрема у системах виявлення вторгнень (IDS) та ідентифікації шкідливого програмного забезпечення.

Незважаючи на широке застосування, дослідники виявили, що алгоритми ML є вразливими до зворотних (adversarial) атак, таких як метод швидкого градієнтного знаку (Fast Gradient Sign Method) та градієнтний спуск з проекцією (Projected Gradient Descent, PGD) [3]. Зловмисники можуть ввести незначні, ледь помітні збурення у вхідні дані, які, проте, призводять до некоректної класифікації моделі, таким чином спричиняючи відмову в роботі алгоритмів ML.

У критично важливих для безпеки сферах, наприклад, у підключених та автономних транспортних засобах (CAVs), збій у виявленні атаки може мати фатальні наслідки. Наприклад, скомпрометована модель ML, що використовується в автономному транспортному засобі для розпізнавання дорожніх знаків, може призвести до неправильного рішення (наприклад,

ігнорування знаку "Стоп"), що потенційно може спричинити дорожньо-транспортну пригоду.

## 1.2. Науковий аналіз алгоритмів генерації доменів та методи їх виявлення

### *1.2.1. Функціональне призначення алгоритмів генерації доменів та виклики для кібербезпеки*

Алгоритми генерації доменів (DGA) є ключовим інструментом, який використовується зловмисниками для динамічного створення множини доменних імен. Основна функція DGA полягає у забезпеченні каналу комунікації для шкідливого програмного забезпечення (malware) із сервером командування та контролю (C2), створеним атакуючою стороною (як ілюстровано на рисунку 1.1). Цей механізм дозволяє маніпулювати мережевими комунікаціями, ефективно мімікруючи під нормальний трафік, і тим самим уникати виявлення. Успішна реалізація такої атаки призводить до крадіжки особистих даних або отримання повного контролю над мережею. Серед відомих зразків DGA-використовуючого шкідливого ПЗ є CryptoLocker, Tovar, Dyre, Nymaim та Locky.

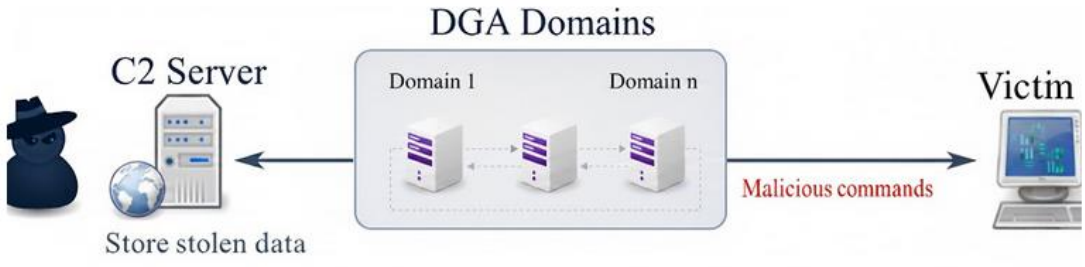



Рис. 1.1. Ілюстрація алгоритму генерації доменів

Традиційні методи контролю шкідливого програмного забезпечення, зокрема використання чорних списків (blacklisting), виявилися недостатніми для ефективної протидії загрозам на основі DGA. Це вимагає глибинного

аналізу та використання специфічних ознак (features) доменних імен, згенерованих DGA.

*1.2.2. Екстракція та аналіз ознак DGA-доменів*

Ключова ідея полягає у тому, що доменні імена, згенеровані DGA, містять статистично значущі ознаки, які можуть бути експлуатовані для їх надійної дискримінації від легітимних доменних імен [6]. Наприклад, доменні імена, згенеровані такими алгоритмами, як CryptoLocker та Tovar (рисунок 1.2), часто мають велику довжину та складаються з випадкових послідовностей символів. Ці характеристики є цінними для виявлення.


Рис. 1.2. Доменні імена, згенеровані алгоритмами генерації доменів

Проте, DGA-домени меншої довжини становлять більший виклик для традиційних методів. Моделі виявлення, як запропонована у [5], демонструють здатність динамічно ідентифікувати DGA-домени незалежно від їх довжини. Використовувані ознаки для виявлення DGA є наступними.

1. Лексичні та URL-ознаки:

- Аналізуються лексичні характеристики URL, включаючи довжину, кількість роздільників (крапок) та наявність спеціальних символів у шляху URL.

- DGA-домени часто характеризуються великою довжиною та низькою лінгвістичною осмисленістю (випадкові рядки). Виявлення може базуватися на порівнянні довжини домену та частотності символів англійського алфавіту [14].

- Метод сегментації слів (word segmentation) може бути використаний для виведення токенів із доменних імен, що дозволяє виявляти шкідливі домени на основі таких параметрів, як кількість символів, цифр та дефісів.

2. DNS-ознаки:

- Ознака NXDOMAIN (Non-Existent Domain) є високоінформативною, оскільки більшість DGA-доменів, що генеруються ботнетами, не зареєстровані та призводять до відповіді NXDOMAIN.

- Можуть бути вилучені статистичні ознаки, пов'язані з послідовностями NXDOMAIN, включаючи розподіл n-грам (послідовності символів довжиною n) для ідентифікації DGA-доменів [9].

- Комбінація лінгвістичних та DNS-ознак може бути експлуатована для підвищення точності виявлення DGA.

### *1.2.3. Застосування методів машинного та глибокого навчання*

Припускаючи, що DGA-домени мають значну кількість відмінних характеристик від легітимних, методи машинного навчання (ML), такі як метод опорних векторів (SVM) та нейронні мережі, можуть ефективно їх розрізняти.

1. Прогнозування та поведінковий аналіз. Деякі дослідження зосереджені на прогнозуванні майбутніх DGA-доменів на основі аналізу минулих зразків [9]. DNS-запити також використовуються як ознака для виявлення патернів у різних групах DGA [11].

2. Застосування методів глибокого навчання (DL) дозволяє безпосередньо здійснювати прогнозування DGA у реальному часі без необхідності у ручній екстракції ознак. Крім того, аналіз подібності та патернів DNS може бути використаний для прогнозування майбутніх DGA-доменів. Інтеграція DL-методів, як обговорюється в [15], дозволяє автоматично вивчати ознаки, усуваючи необхідність у людських зусиллях для їх аналізу.

Для розробки ефективних контрзаходів є критично важливим ретельне вивчення ботнетів, які застосовують DGA для генерації доменних імен. Низка досліджень зосереджена на розумінні внутрішньої логіки роботи ботнетів та їх зворотній інженерії (reverse-engineering) [26].

## 1.3. Глибинний аналіз вразливості протоколу WPA2 та методи протидії KRACK атаці

### *1.3.1. Вразливості WPA2 та механізм KRACK атаки*

Більшість сучасних бездротових мереж Wi-Fi використовують протокол Wi-Fi Protected Access 2 (WPA2), безпека якого забезпечується через чотиристоронніє рукостискання (4-way handshake). Численні вразливості протоколу WPA2 були виявлені та описані у науковій літературі [10]. З огляду на зростаючу кількість пристроїв, що підключаються до бездротових мереж, забезпечення надійності цих мереж набуває критичного значення.

Проте, у механізмі 4-стороннього рукостискання протоколу WPA2 були виявлені серйозні недоліки, які дозволяють зловмисникам, що перебувають у зоні дії точки доступу, здійснити атаку повторної встановлення ключа (Key Reinstallation Attack, KRACK) з метою перехоплення конфіденційної інформації.

Розглянемо механізм 4-стороннього рукостискання та вектор KRACK. Механізм 4-стороннього рукостискання є фундаментальною комунікаційною процедурою між користувацьким пристроєм та точкою доступу, що відбувається під час приєднання користувача до мережі. Його мета — підтвердити, що обидві сторони володіють коректними обліковими даними (credentials).

Розглянемо етапи алгоритму. Точка доступу ініціює рукостискання, надсилаючи повідомлення 1 користувачу. Користувач відповідає повідомленням 2, сигналізуючи про намір приєднатися. Після згоди

користувача, точка надсилає повідомлення 3, яке містить новий ключ шифрування. Цей ключ має бути встановлений та використаний лише один раз. На завершення користувач встановлює ключ і надсилає повідомлення 4 до точки доступу, підтверджуючи успішне з'єднання.

В атаці KRACK зловмисник діє як посередник (Man-in-the-Middle, MITM). Шляхом маніпуляцій, атакуючий може заблокувати доставку повідомлення 4 до AP. Це змушує точку доступу виконати повторну відправку повідомлення 3 з тим самим ключем шифрування, що змушує пристрій користувача багаторазово перевстановити ідентичний ключ. Експлуатація цієї вразливості дозволяє зловмисникам дешифрувати конфіденційні дані, що передаються мережею.

### *1.3.2. Масштаб впливу та наслідки KRACK атаки*

Оскільки всі бездротові мережі Wi-Fi використовують протокол 4-стороннього рукостискання, усі вони потенційно вразливі до KRACK. Практично вся інформація, передана користувачем через скомпрометовану мережу (включаючи облікові дані, фінансову інформацію, тощо), може бути дешифрована. Крім того, через архітектуру мережевого зв'язку та конфігурацію підключеного пристрою, дані, які надсилаються користувачеві, також можуть бути дешифровані.

Наведемо поведінковий аспект та IoT-загрози:

1. Аналіз поведінки користувачів.

Дослідження, що включало кількісне опитування 379 осіб [21], показало, що лише досвідчені користувачі своєчасно застосовують оновлення безпеки (security patches) для уникнення KRACK. Було встановлено, що більшість користувачів використовують слабкі паролі, які легко піддаються дешифруванню.

2. Проблема підтримки пристроїв.

Хоча оновлення безпеки є найкращим захистом, на деяких пристроях оновлення неможливе через завершення терміну підтримки ("end-of-life").

3. Довгостроковий вплив на IoT.

KRACK справила значний вплив на інтернет-інфраструктуру. Окрім крадіжки персональних даних, вона несе потенційну довгострокову загрозу для екосистеми Інтернету Речей (IoT), зокрема через компрометацію підключених систем контролю доступу.

*1.3.3. Захисні фреймворки та стратегії*

Забезпечення захисту від KRACK є імперативом для збереження приватності користувачів.

1. Фреймворк виявлення Krack-Cover

Krack-Cover є прикладом фреймворку бездротової безпеки, розробленого для протидії KRACK [23]. Його функціонал включає:

- Постійний моніторинг та захоплення всіх повідомлень стандарту 802.11x.

- Автоматичне спрацьовування сповіщення (alert) для користувача одразу після виявлення ознак KRACK.

Цей метод довів свою ефективність у реальних умовах Wi-Fi середовищ (наприклад, публічні місця, кафе).

2. Схема виявлення на основі EAPOL-кадрів

Повідомлення 4-стороннього рукостискання інкапсулюються у кадр EAPOL (як показано на рисунку 1.3). Детальне вивчення структури EAPOL-кадру дозволяє розробити схему виявлення.

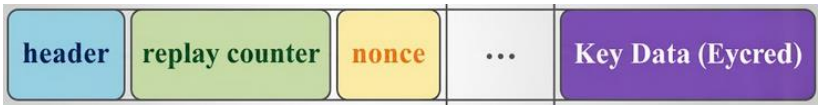



Рис. 1.3. Спрощений формат кадру EAPOL

Алгоритмічний підхід до виявлення наступний:

1) Екстракція Ethernet-рівня мережевого пакету.

2) Екстракція заголовка IEEE 802.11x.

3) Вилучення даних ключа WPA Key Data.

4) Аналіз на наявність дублікатів повідомлення 3.

5) При виявленні дублікату — надсилання попередження користувачу про атаку.

3. Рекомендації з мітігації

Для підвищення стійкості до KRACK рекомендуються такі заходи:

- Використання розширеного стандарту шифрування (Advanced Encryption Standard, AES).

- Вимкнення функціоналу швидкого роумінгу (fast roaming).

- Ручне оновлення програмного забезпечення Wi-Fi на всіх пристроях.

- Активний моніторинг мережі на предмет виявлення шахрайських точок доступу (rogue APs).

## 1.4. Аналіз фішингових атак та поведінкових аспектів кібербезпеки

### *1.4.1. Природа та механізм фішингових атак*

Фішинг (Phishing) — це тип кібератаки, основною метою якої є несанкціоноване заволодіння конфіденційною інформацією користувачів, такою як дані кредитних карток та облікові (логін) дані. З часом фішинг набув надзвичайно широкого розповсюдження.

У типовій фішинговій атаці зловмисники застосовують методи соціальної інженерії, маскуючись під довірену сторону (наприклад, банк, державну установу, відомий сервіс) з метою обманом змусити користувача розкрити приватну інформацію. Атака може бути реалізована через електронну пошту або текстові повідомлення (SMS).

Оскільки електронна пошта є широковживаним інструментом у повсякденній діяльності, вона стала найбільш поширеним методом здійснення фішингових атак.

Фішингові повідомлення зазвичай містять три основні ознаки, які використовуються для обману (рис. 1.4):

- Підозріла адреса електронної пошти відправника.

- Підозрілі гіперпосилання або вкладення.

- Підозрілий або нетиповий зміст листа.

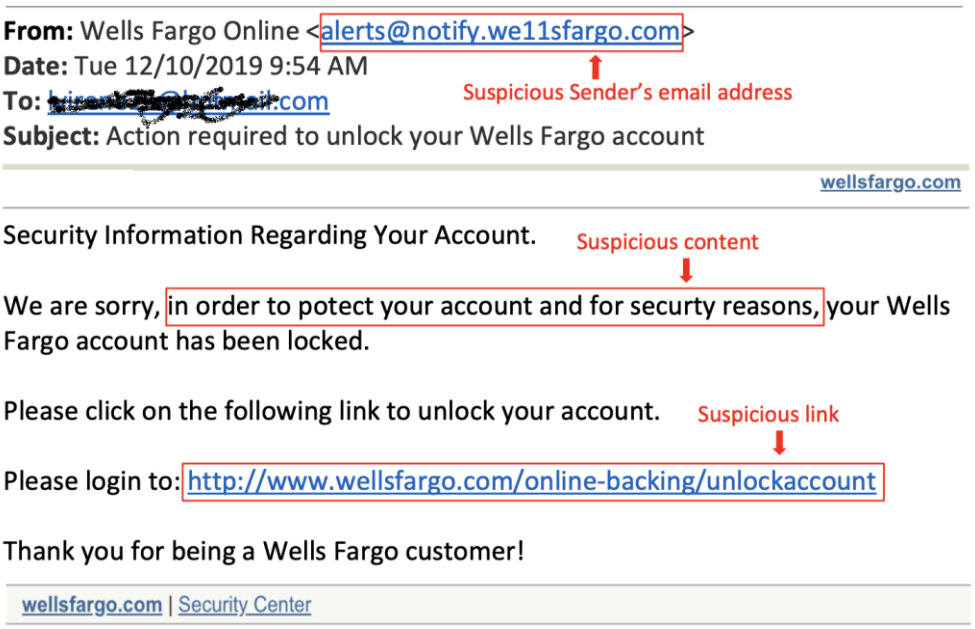



Рис. 1.4. Приклад фішингового електронного листа

Представлені нижче приклади ілюструють типові сценарії фішингових атак, класифікованих за вектором соціальної інженерії та цільовою аудиторією.

Імітація служби технічної підтримки (Helpdesk Phishing) - цей тип атаки є прикладом спірфішингу (spear phishing) — високотаргетованого підходу, спрямованого на конкретну особу або невелику групу. Імітація звернення від внутрішньої IT-служби або Helpdesk є ефективною тактикою для зниження пильності жертви.

Ключовий психологічний тригер це спроба спровокувати сильну емоційну реакцію (наприклад, страх, тривогу) шляхом використання теми, що сигналізує про «загрозу» або «підозрілу активність». Це вимагає негайної, некритичної реакції від користувача.

Ознаки виявлення фішингу є наступними. Попередження про зовнішнього відправника - наявність чітко видимого системного маркера

(наприклад, яскраво-жовтого тегу «EXTERNAL») свідчить про те, що повідомлення надійшло з-за меж корпоративної мережі. Легітимні внутрішні комунікації не містять такого маркування.

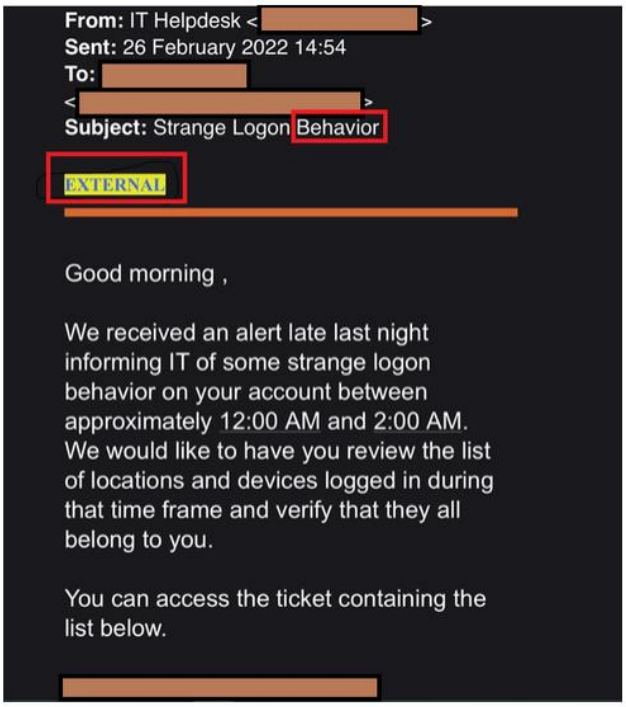



Рис. 1.5. Фішинговий лист (імітація служби технічної підтримки)

Ще ознакою є лексико-граматичні аномалії. В даному випадку це використання невідповідного регіонального варіанта орфографії (наприклад, американізоване написання 'behavior' замість 'behaviour' у британському контексті) або наявність типографських помилок. Це є поширеною ознакою фішингу, хоча очікується, що майбутні інструменти на базі штучного інтелекту можуть усунути ці «помилки». Також вимога відкрити вкладений документ (наприклад, файл Word) у такому контексті є аномальною для корпоративних сповіщень. Перехід за посиланням або відкриття вкладення у фішинговому листі науково інтерпретується як свідоме надання зловмиснику несанкціонованого доступу до системи.

Імітація фінансових сповіщень (Fake Payment Phishing) - це класичний сценарій фішингу, який експлуатує базові людські емоції, пов'язані з фінансовою винагородою.

Ключовий психологічний тригер це маніпуляція позитивною емоцією (радість, пов'язана з отриманням грошей — «Yay, I’ve been paid – money!»). Це створює сильний мотиваційний імпульс для негайної дії.

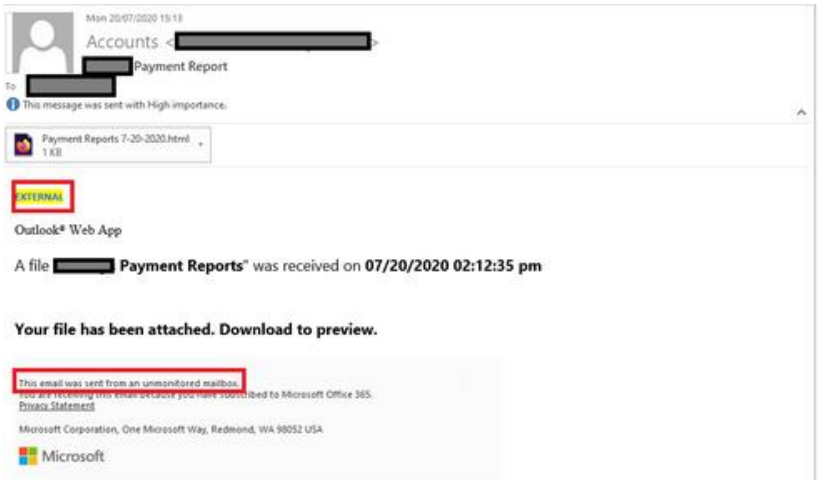



Рис. 1.6. Фішинговий лист (імітація фінансових сповіщень)

Ознаки виявлення фішингу:

- Повторна наявність маркера «EXTERNAL» у листі, який нібито походить від внутрішнього відділу (наприклад, Бухгалтерії/Accounts).

- Наявність системних повідомлень, як-от дисклеймер Microsoft щодо неконтрольованої поштової скриньки. Більшість легітимних внутрішніх корпоративних повідомлень не містять таких технічних попереджень.

- Низька персоналізація та анонімність, відсутність персоналізованого звернення (наприклад, імені одержувача), відсутність стандартного привітання/прощання або детальних даних відправника. Ці ознаки вказують на масовий характер атаки та спробу зловмисника приховати свою ідентичність.

### *1.4.2. Психологічні та поведінкові чинники вразливості*

Більшість фішингових листів можуть бути виявлені при прояві обережності та усвідомленні типових характеристик таких загроз. Однак, якщо електронний лист містить інформацію, що сприймається користувачем

як важлива та термінова, вірогідність ігнорування очевидних ознак фішингу значно зростає.

Цікавим є висновок, що користувачі, які мають більший досвід роботи з електронною поштою, частіше піддаються фішингу. Для ефективного запобігання фішингу критично необхідним є розуміння поведінкових реакцій користувачів на зіткнення з цими атаками.

Дослідники провели численні експерименти для пояснення високої сприйнятливості до фішингу:

1. Ігнорування індикаторів безпеки. Експеримент за участю 22 респондентів, які визначали фішингові вебсайти, показав, що ігнорування індикаторів безпеки є головною причиною неправильних рішень [20].

2. Індивідуальні звички. Інше дослідження виявило, що індивідуальні звички використання електронної пошти відіграють значну роль [21]. Особи, які мають звичку відкривати листи одразу після сповіщення, часто неусвідомлено, демонструють вищу вразливість до фішингу.

3. Недовіра до засобів захисту. Недовіра до індикаторів безпеки на вебсайтах також є чинником вразливості. Незалежно від наявності індикаторів, користувачі часто оцінюють вебсайт чи електронний лист, ґрунтуючись лише на суб'єктивному відчутті та зовнішньому вигляді.

### *1.4.3. Підходи до підвищення обізнаності та запобігання фішингу*

Для підвищення стійкості користувачів до фішингових атак необхідно досліджувати фактори, що сприяють покращенню обізнаності.

Навчальні інтервенції можуть бути важливим фактором для вивчення впливу на сприйнятливість користувачів до фішингу. Незважаючи на існування індикаторів безпеки та попереджувальних сповіщень, більшість користувачів ігнорують або не довіряють цим індикаторам. Більше того, значна частина користувачів не розуміє, що таке фішинг.

З огляду на вищезазначене, невідкладним завданням є:

1. Освіта користувачів щодо сутності фішингу.

2. Навчання довірі та увазі до індикаторів безпеки.

3. Розуміння того, як навчальні інтервенції можуть модифікувати поведінку користувачів при зіткненні з фішинговими загрозами.

Крім безпосередніх інтервенцій, необхідно проводити подальші дослідження інших поведінкових, когнітивних та технічних факторів, які впливають на сприйнятливість до фішингу.

## 1.5. Дослідження вразливостей моделей машинного навчання до змагальних атак

### *1.5.1. Концепція та класифікація змагальних атак*

Технології машинного навчання (ML) та глибокого навчання (DL) демонструють значні успіхи у різноманітних доменах, включаючи комп'ютерний зір та мережеву безпеку. Проте, дослідження виявляють вразливість ML-моделей до змагальних (adversarial) атак.

Механізм змагальної атаки наступний: Зловмисник (adversary) маніпулює оригінальним вхідним об'єктом (x) шляхом додавання незначного, ледь помітного збурення (r), створюючи новий вхідний об'єкт (x′), який є візуально або функціонально ідентичним x (x′≈x), але призводить до некоректного виходу класифікації моделі (рис. 1.7).

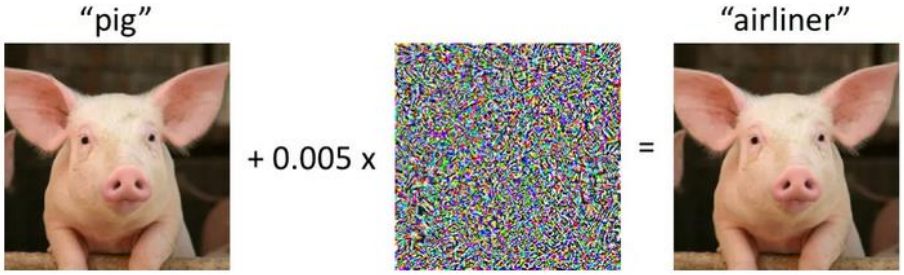



Рис. 1.7. Приклад змагальної атаки

Наведемо класифікацію за рівнем знань атакуючого:

1. Атака «Білого ящика» (White-Box Attack):

- Атакуючий володіє повною інформацією про цільову ML-модель. Це включає архітектуру (кількість шарів нейронної мережі), деталі оптимізаційного алгоритму та параметри навченої моделі (наприклад, ваги).

- Доступ до внутрішніх параметрів дозволяє здійснити високоефективні та сильні змагальні атаки.

2. Атака «Чорного ящика» (Black-Box Attack):

- Атакуючий не має попередніх знань про внутрішню структуру або параметри цільової ML-моделі.

- Зловмисник аналізує реакції моделі на минулі вхідні дані та вихідні результати, щоб виявити вразливості та розробити ефективні збурення.

*1.5.2. Вектори атак за фазами життєвого циклу моделі*

Змагальні атаки можуть бути реалізовані як у фазі навчання (training phase), так і у фазі тестування/виводу (testing/inference phase).

Таблиця 2.1.

Фази атак

| **Фаза атаки** | **Тип атаки** | **Механізм компрометації** |
|---|---|---|
| **Фаза навчання** | **Poisoning Attack** | Зловмисник компрометує модель шляхом:<br>1) **Ін'єкції неправдивих даних** у навчальний набір;<br>2) **Модифікації існуючих даних** у навчальному наборі.<br>Це призводить до **порушення логіки** алгоритму навчання, що ускладнює розробку контрзаходів. |
| **Фаза тестування** | **Evasion Attacks & Exploratory Attacks** | Зловмисник намагається змусити вже навчену модель генерувати некоректні вихідні результати. Вимагає знання про модель для ефективності. |

Класифікація за поверхнею атаки:

1. Атаки уникнення (Evasion Attacks) - найбільш поширений сценарій, що відбувається лише у фазі тестування. Зловмисник маніпулює шкідливими вхідними даними для уникнення виявлення, не впливаючи на навчальний

набір. Атаки «білого ящика» та «чорного ящика» зазвичай розглядаються як атаки уникнення, оскільки не припускають впливу на навчальні дані.

2. Атаки отруєння (poisoning attacks) здійснюються у фазі навчання шляхом маніпуляції навчальними даними. Отруєння навчального набору компрометує весь процес навчання, призводячи до створення низькоякісної ML-моделі.

3. Експлораторні атаки (exploratory attacks) є формою атаки «чорного ящика». Зловмисник компрометує модель шляхом збору додаткової інформації про патерни у навчальних даних та сам процес навчання, щоб виявити слабкі місця.

Серед відомих алгоритмів змагальних атак варто виділити: Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), Jacobian Based Method, Deepfool, та метод Carlini & Wagner (C&W) [17].

### *1.5.3. Критичні наслідки змагальних атак*

Більшість існуючих досліджень зосереджено на успішних змагальних атаках на зображення [19, 22], тоді як інші домени залишаються менш вивченими.

Оскільки багато критично важливих систем покладаються на точність ML-моделей, особливо у сферах, де безпека є пріоритетом (наприклад, автономні транспортні засоби), збій ML-моделі може призвести до катастрофічних наслідків.

Наведемо приклад у сфері автономних транспортних засобів (AVs). Добре навчена ML-модель може застосовуватися в AVs для високоточного виявлення дорожніх знаків (наприклад, знаку «Стоп»).

Якщо ця модель буде скомпрометована змагальною атакою, це призведе до некоректних рішень. Наприклад, автономний транспортний засіб може проігнорувати знак «Стоп», що потенційно може спричинити дорожньо-транспортну пригоду.

## Висновки до розділу

У першому розділі проведено комплексний аналіз сучасного стану мережевої безпеки, класифікації атак, загроз та вразливостей, що виникають у комп'ютерних системах. Визначено, що сучасні інформаційні інфраструктури характеризуються високою динамічністю та складністю, що створює сприятливе середовище для появи нових типів кіберзагроз. Досліджено вразливості систем доменних імен (DNS), бездротових мереж (WPA2), а також методи реалізації фішингових атак, які залишаються одним із найефективніших інструментів соціальної інженерії.

Проаналізовано алгоритми генерації доменів (DGA) як одну з ключових технологій розповсюдження шкідливого ПЗ, здатну обходити традиційні методи фільтрації. Особливу увагу приділено змагальним атакам на моделі машинного навчання, що становлять новий клас загроз у сфері кіберзахисту. Розглянуто поведінкові та психологічні чинники, що впливають на сприйнятливість користувачів до фішингових повідомлень. Визначено, що сучасні системи захисту повинні поєднувати технічні та когнітивні підходи, інтегруючи машинне навчання для автоматизації виявлення атак.

# РОЗДІЛ 2. ДОСЛІДЖЕННЯ МЕТОДОЛОГІЇ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

Представлений розділ присвячений дослідженню безпеки доменних імен та описує фреймворк для виявлення та прогнозування шкідливого програмного забезпечення, що використовує алгоритми генерації доменів (DGA).

## 2.1. Особливості фреймворку виявлення шкідливого ПЗ на основі алгоритму генерації доменів

### *2.1.1. Огляд проблематики*

Зловмисники постійно намагаються проникнути через багаторівневі захисні рішення, створюючи загрозу для комп'ютерних мереж та їхніх активів. Традиційні антивірусні рішення, які широко застосовуються в корпоративному секторі, часто покладаються на прості механізми, такі як хешування, статичне порівняння рядків та чорні списки. Ці рішення є недостатньо стійкими перед обличчям складних шкідливих програм, які використовують техніки ухилення для приховування своїх каналів комунікації та обходу більшості систем виявлення. Ця проблема становить серйозну загрозу для безпеки підприємств і є важливим викликом, що вимагає негайного вирішення.

Багато складних зловмисних програм використовують або статичний, або динамічний підхід для зв'язку зі своїм сервером командування та контролю (C2) [16].

Статичний підхід характеризується фіксованими параметрами (наприклад, постійна IP-адреса та незмінне доменне ім'я протягом усього життєвого циклу шкідливої програми). Виявлення такої загрози дозволяє застосувати прості правила блокування.

Широко використовується алгоритм генерації доменів (DGA). DGA — це алгоритм послідовності, який періодично генерує велику кількість доменних імен, що дозволяє шкідливій програмі ухилятися від доменних міжмережевих екранів. Згенеровані домени допомагають приховати C2-сервери, що ускладнює їх ідентифікацію. Домени, створені DGA, є короткочасними і, хоча можуть бути ідентифіковані людиною, їх автоматичне виявлення є складним для машинних систем.

*2.1.2. Методологія дослідження та архітектура фреймворку*

Це дослідження оцінює відомі алгоритми DGA та аналізує зловмисні домени, що ними генеруються. Ми застосовуємо підходи машинного навчання (ML), включаючи екстракцію множинних ознак, класифікацію, кластеризацію та методи прогнозування, для розуміння характеристик DGA-доменів. Додатково розроблено модель глибокої нейронної мережі (DNN) для класифікації великих наборів даних.

Ми здійснюємо моніторинг та аналіз кожного DNS-запиту в мережі (які часто виникають від запущених додатків і сервісів), щоб визначити, чи походить конкретний домен та запит від DGA, і ідентифікувати, який саме DGA його згенерував.

Запропонований ML-фреймворк складається з чотирьох основних компонентів для ідентифікації та класифікації DGA-доменів:

- Динамічний чорний список з фільтром патернів, що використовується для фільтрації вхідних DNS-запитів та отримання доменів, які потім зберігаються в чорному списку.

- Екстрактор ззнак (Feature Extractor) вилучає ознаки з вхідних доменів, які не присутні в чорному списку.

- Дворівнева ML-модель:

Перший рівень (класифікація) - використовуються різноманітні моделі класифікації для розрізнення DGA-доменів від нормальних.

Другий рівень (кластеризація) - застосовується метод кластеризації (зокрема, неконтрольований алгоритм DBSCAN) для групування DGA-доменів за їхнім алгоритмом генерації.

- Модель прогнозування часових рядів. Запропоновано приховану марковську модель (Hidden Markov Model, HMM) для прогнозування майбутніх ознак DGA-доменів.

Основна мета фреймворку це визначити, який алгоритм DGA використовується, для запобігання майбутнім комунікаціям з відповідним C2-сервером.

Таблиця 2.1.

Основні результати оцінки

| **Компонент фреймворку** | **Методологія** | **Точність (Accuracy)** |
|---|---|---|
| Перший рівень (класифікація) | ML-моделі | 95.89% |
| Другий рівень (кластеризація) | DBSCAN (Cluster Labeling) | 92.45% |
| Прогнозування | HMM | 95.21% |
| Класифікація великих наборів | Модель DNN | 97.79% |

Побудована модель глибокої нейронної мережі (DNN), що порівнюється з класичними ML-моделями. У DNN застосовувалися різні алгоритми оптимізації для досягнення вищої точності. Для запобігання перенавчанню (overfitting) використовувалося розділення даних на навчальний та валідаційний набори.

## 2.2. Дослідження загроз генерації доменів та методологія виявлення невідомих атак

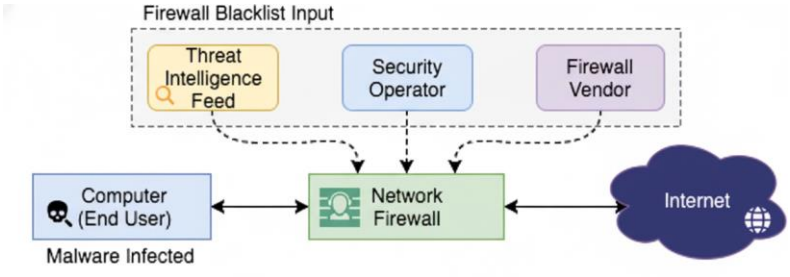
### *2.2.1. Обмеження традиційних методів захисту*

DGA (алгоритм генерації доменів) — це техніка, яку використовують зловмисники для періодичної генерації великої кількості доменних імен.

Ботнети повинні зв'язуватися з сервером управління (C2 — Command & Control), щоб отримувати команди. Якщо в коді вірусу прописати одну конкретну адресу (наприклад, evil-server.com), її швидко заблокують, і ботнет перестане працювати. DGA дозволяє вірусу щодня "вигадувати" тисячі нових доменів (наприклад, xys12-a.com, fgh99-b.net). Хакери реєструють лише один із них, а вірус перебирає згенеровані варіанти, доки не знайде активний. Це ускладнює роботу захисникам, оскільки не можна просто заблокувати одну адресу — потрібно блокувати сам алгоритм або тисячі потенційних доменів наперед

Незважаючи на постійне розширення чорних списків міжмережевих екранів (firewall blacklisting) за рахунок зростаючого обсягу вхідних даних (джерел загроз), послідовності, генеровані DGA, можуть бути невідомі цим джерелам на початкових етапах.

Більше того, для забезпечення ефективної комунікації шкідливого програмного забезпечення з функціональним доменом, зловмисник (threat actor) повинен послідовно реєструвати кожен відповідний домен у послідовності DGA для підтримки C2-інфраструктури або ризикує втратити контроль над скомпрометованими вузлами. Рисунок 2.1 ілюструє сценарій такої комунікації.




| DGA Sequence | Firewall Blacklist | Internet Domain | Filtered |
|---|---|---|---|
| swgqwmnpxuorh.com | swgqwmnpxuorh.com | swgqwmnpxuorh.com | Yes |
| ghluburrkboax.net | ghluburrkboax.net | ghluburrkboax.net | Yes |
| veivkqvvamlgx.biz | veivkqvvamlgx.biz | veivkqvvamlgx.biz | Yes |
| wykwohntbxrlh.org | | wykwohntbxrlh.org | No |
| kjpbsprvnertx.co.uk | | | ✗ |
| ahmsbkdcffkst.info | | ahmsbkdcffkst.info | No |

Рис. 2.1. Моделі загроз DGA

Основне завдання полягає у точній класифікації та кластеризації доменів, які походять від відомих DGA-методів. Мета — розробити підхід, що забезпечує автоматичне запобігання мережевому трафіку, пов'язаному з невідомими (новими) загрозами у послідовності DGA.

*2.2.2. Моделі загроз*

Зловмисники потребують надійного методу контролю та підтримки шкідливого програмного забезпечення в середовищі C2, забезпечуючи при цьому непомітність для систем мережевої безпеки.

Припущення:

- Успіх шкідливої програми не залежить від того, чи є домен зареєстрованим або дійсним.

- Ітерація послідовності DGA часто призводить до стану NXDOMAIN (Non-Existent Domain, незареєстрований домен).

Для запобігання зловмисній мережевій активності від шкідливого ПЗ зазвичай застосовуються такі стандартні методи, як чорні списки, встановлення DNS-сінхолів (sinkhole) та реалізація правил міжмережевого екрану. Сигнатури для цих методів часто надаються через канали розвідувальних даних про загрози (threat intelligence feeds).

У початкових етапах нашого аналізу дане дослідження свідомо не використовує будь-які попередньо визначені чорні списки з раніше відомими зловмисними доменами для блокування DGA-трафіку. Ця особливість є ключовою для нашої реалізації, оскільки:

- більшість каналів розвідувальних даних та евристичні дані надають сигнатури для шкідливих програм, які вже були виявлені у мережі або в публічному Інтернеті.

- складний зловмисник імплементує або використовує шкідливе програмне забезпечення у стилі «нульового дня» (0-day), яке є невідомим для громадськості.

Отже, використання чорних списків є недоцільним для нашого аналізу, спрямованого на виявлення нових загроз.

Запропонований фреймворк виявлення шкідливого програмного забезпечення на основі DGA прагне вирішити проблему ідентифікації послідовностей DGA за допомогою методів машинного навчання, які виводяться з спостережень у мережі.

## 2.3. Методологія збору даних та архітектура фреймворку виявлення загроз генерації доменів

Ключовими складовими даного розділу є: доменні імена, згенеровані алгоритмами генерації доменів (DGA); архітектура фреймворку виявлення шкідливого програмного забезпечення на основі DGA, яка охоплює комплекс методів екстракції характеристик, моделі для бінарної класифікації DGA- та легітимних доменів, кластеризації DGA-доменів та прогнозування їхніх характеристик. Нижче детально описано методологію збору DGA-даних та архітектуру запропонованого фреймворку для ідентифікації доменних імен, згенерованих DGA.

### *2.3.1. Збір даних DGA*

Доступність DGA-алгоритмів підтверджується численними відкритими зразками, що представлені у сховищах типу GitHub та загальнодоступних пошукових системах (наприклад, Google). Проте, складні кіберзагрози адаптивно генерують доменні імена DGA з метою обходу існуючих систем виявлення. Для емпіричної оцінки точності запропонованого підходу необхідне використання активних зловмисних доменів, отриманих з DGA у реальному Інтернет-середовищі.

Структура даних представлена у форматі CSV та містить такі атрибути, як доменні імена, походження шкідливого програмного забезпечення та ідентифікатор сімейства DGA. Середній щоденний обсяг файлів становив

приблизно 110 МБ. Приклад структури даних ілюструється на рисунку 2.2, де представлені доменні імена, джерело шкідливого ПЗ та схеми DGA.

```
vsnpkfq.net,Domain used by Cryptolocker - Flashback DGA for 16 Jan 2024,2024-01-16
lgjhsnp.biz,Domain used by Cryptolocker - Flashback DGA for 16 Jan 2024,2024-01-16
ndmsjoj.ru,Domain used by Cryptolocker - Flashback DGA for 16 Jan 2024,2024-01-16
dqikrwr.org,Domain used by Cryptolocker - Flashback DGA for 16 Jan 2024,2024-01-16
vwqsnyu.co.uk,Domain used by Cryptolocker - Flashback DGA for 16 Jan 2024,2024-01-16
```

Рис. 2.2. Приклад набору даних DGA

*2.3.2. Архітектура фреймворк виявлення шкідливого програмного забезпечення*

Ми пропонуємо архітектуру фреймворку для виявлення шкідливого програмного забезпечення на базі DGA, яка включає три основні процедури (рисунок 2.3).

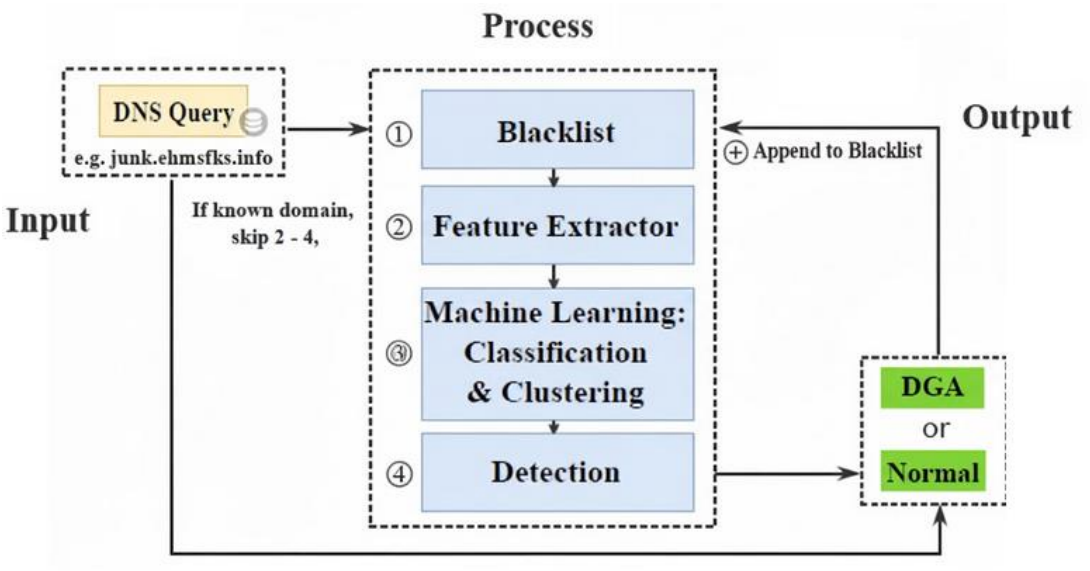



Рис. 2.3. Архітектура фреймворку

Вхідними даними слугують DNS-запити з відповідним корисним навантаженням. Далі ці запити надходять до етапу обробки, який складається з чотирьох ключових компонентів:

1. Фільтрація та динамічний чорний список. На початковому етапі доменні імена вилучаються за допомогою фільтра пакетів DNS-запитів і

зберігаються у динамічному чорному списку. Якщо вхідний домен вже ідентифікований як відомий (присутній у чорному списку), етапи (2)-(4) пропускаються, і система одразу переходить до виходу. В іншому випадку обробка продовжується.

2. Екстракція характеристик. Застосовується спеціалізований екстрактор для вилучення релевантних ознак домену.

3. Дворівнева модель ML: реалізується бінарна класифікація першого рівня для розрізнення DGA- та не-DGA доменів, а потім кластеризація другого рівня для групування подібних DGA-доменів.

4. Прогнозування часових рядів. Використовується модель часових рядів для прогнозування характеристик майбутніх доменів.

Після успішного завершення етапу обробки доменне ім'я додається до динамічного чорного списку. Наступні підрозділи присвячені детальному опису чотирьох компонентів етапу обробки.

Єдиною необхідною інформацією для подальших етапів класифікації та прогнозування є власне доменні імена. З метою видалення нерелевантних даних із сирого мережевого трафіку, ми застосовуємо фільтр пакетів DNS-запитів. Для реалізації процесу фільтрації використовується фільтр шаблонів. Весь мережевий трафік піддається цій процедурі фільтрації, в результаті чого отримуються виключно доменні імена. Відфільтровані домени зберігаються у динамічному чорному списку, який ініціалізується як порожній і підлягає постійному динамічному оновленню. Далі домени передаються до екстрактора характеристик. Використання динамічного чорного списку забезпечує зменшення обчислювального навантаження, оскільки наявність домену в списку дозволяє негайно перейти до етапу виводу, оминаючи подальші етапи аналізу.

Екстрактор характеристик призначений для вилучення ознак із доменних імен, отриманих після фільтрації на попередньому етапі. Доменне ім'я розглядається як послідовність символів (рядок). Для забезпечення

ефективної класифікації доменних імен ми застосовуємо два класи характеристик: лінгвістичні та DNS-характеристики.

Було визначено шість лінгвістичних характеристик: довжина домену, співвідношення значущих слів, частка цифрових символів, оцінка вимовності, частка довжини найдовшого значущого рядка (LMS) та відстань редагування Левенштейна.

Довжина (|d|) - загальна кількість символів у доменному імені.

Співвідношення значущих слів ($f_1$) - кількісно оцінює частку значущих слів у доменному імені за формулою:

$$\mathbf{f_1} = \sum_{i=1}^{n} \frac{|w_i|}{|d|}$$

Де $w_i$ - i-тий значущий підрядок, $w_i$ – його довжина. Низьке значення $f_1$ корелює з DGA-доменами. Довжина значущого підрядка встановлена не менше чотирьох літер.

Оцінка вимовності ($f_2$) характеризує легкість вимови слова, використовуючи таблицю пошуку n-грам ($n \in \{2,3\}$). DGA-домени, як правило, мають нижчу оцінку $f_2$. Розрахунок визначається як:

$$\mathbf{f_2} = \frac{\sum n - \text{gram}(d)}{|d| - n + 1}$$

де n — довжина n-грама.

Частка цифрових символів ($f_3$) - відношення кількості цифрових символів (m) до загальної довжини домену:

$$\mathbf{f_3} = \frac{|m|}{|d|}$$

Частка довжини LMS ($f_4$) - відносна довжина найдовшого значущого рядка (LMS):

$$\mathbf{f_4} = \frac{|l|}{|d|}$$

де $l$ – довжина LMS.

Відстань редагування Левенштейна вимірює мінімальну кількість односимвольних редагувань між поточним доменом та його безпосереднім попередником у послідовності DNS-запитів.

На додаток до лінгвістичних ознак, в моделі використовуються 33 DNS-характеристики (таблиця 2.2). Властивістю DGA-доменів є їхній короткий життєвий цикл та недавня генерація, що призводить до меншої кількості доступної інформації у порівнянні з легітимними доменами.

Таблиця 2.2.

DNS характеристики

| Категорія | Характеристика | Опис | Легітимність (+/-) |
|---|---|---|---|
| Лінгвістичні | Співвідношення значущих слів | Частка значущих слів у домені | + |
| Лінгвістичні | Вимовність | Легкість фонетичної вимови | + |
| Лінгвістичні | % цифрових символів | Частка цифрових символів | - |
| Лінгвістичні | % довжини LMS | Співвідношення довжини найдовшого значущого рядка | + |
| Лінгвістичні | Довжина доменного імені | Загальна довжина домену | - |
| Лінгвістичні | Відстань редагування Левенштейна | Мінімальна кількість редагувань від останнього домену | + |
| DNS | Термін дії | Термін дії 1 року | + |
| DNS | Дата створення | Дата створення 1 року | + |
| DNS | DNS-запис | Наявність задокументованого DNS-запису | + |
| DNS | Унікальні IP-адреси | Кількість пов'язаних унікальних IP-адрес | + |
| DNS | Кількість різних країн | Кількість країн, пов'язаних з доменом | + |
| DNS | IP, що використовується доменами | Кількість доменів, що використовують ту ж IP-адресу | - |
| DNS | Результати зворотного | Наявність домену (DN) у топ-3 | + |

| Категорія | Характеристика | Опис | Легітимність (+/-) |
|---|---|---|---|
| | DNS-запиту | результатів зворотного запиту | |
| DNS | Піддомен | Зв'язок домену з іншими піддоменами | + |
| DNS | Середній TTL | Середній час кешування DNS-даних серверами | + |
| DNS | Стандартне відхилення TTL | Розподіл стандартного відхилення TTL | - |
| DNS | % використання діапазонів TTL | Діапазон розподілу TTL | + |
| DNS | Кількість різних значень TTL | Різні значення TTL на сервері | - |
| DNS | Кількість змін TTL | Частота змін TTL | + |
| DNS | Дозвіл клієнта на видалення | Наявність дозволу клієнта на видалення | - |
| DNS | Дозвіл клієнта на оновлення | Наявність дозволу клієнта на оновлення | - |
| DNS | Дозвіл клієнта на передачу | Наявність дозволу клієнта на передачу | - |
| DNS | Дозвіл сервера на видалення | Наявність дозволу сервера на видалення | - |
| DNS | Дозвіл сервера на оновлення | Наявність дозволу сервера на оновлення | - |
| DNS | Дозвіл сервера на передачу | Наявність дозволу сервера на передачу | - |
| DNS | Реєстратор | Інформація про реєстратора доменного імені | + |
| DNS | Whois Guard | Використання Whois Guard для захисту конфіденційності | - |
| DNS | IP-адреса в тій же підмережі | Наявність IP-адреси в тій же підмережі | - |
| DNS | Назва бізнесу | Наявність назви корпорації, пов'язаної з доменом | + |
| DNS | Географічне розташування | Надання адреси | + |
| DNS | Номер телефону | Надання контактного номера телефону | + |
| DNS | Локальний хостинг | Використання локальної хост-машини | + |
| DNS | Популярність | Включення до топ-10000 списку доменів | + |

З метою підвищення глибини розуміння DGA-доменів, ми пропонуємо дворівневу модель машинного навчання, що складається з класифікації першого рівня та кластеризації другого рівня.

На першому етапі використовуються класифікатори ML для бінарного розрізнення DGA-доменів від легітимних. Серед протестованих класифікаторів (Decision Tree-J48, ANN, SVM, логістична регресія, NB, GBT, випадковий ліс), останній демонструє найкращу ефективність.

Кластеризація другого рівня - для сегментації DGA-доменів на групи відповідно до їхнього алгоритму генерації застосовується алгоритм DBSCAN.

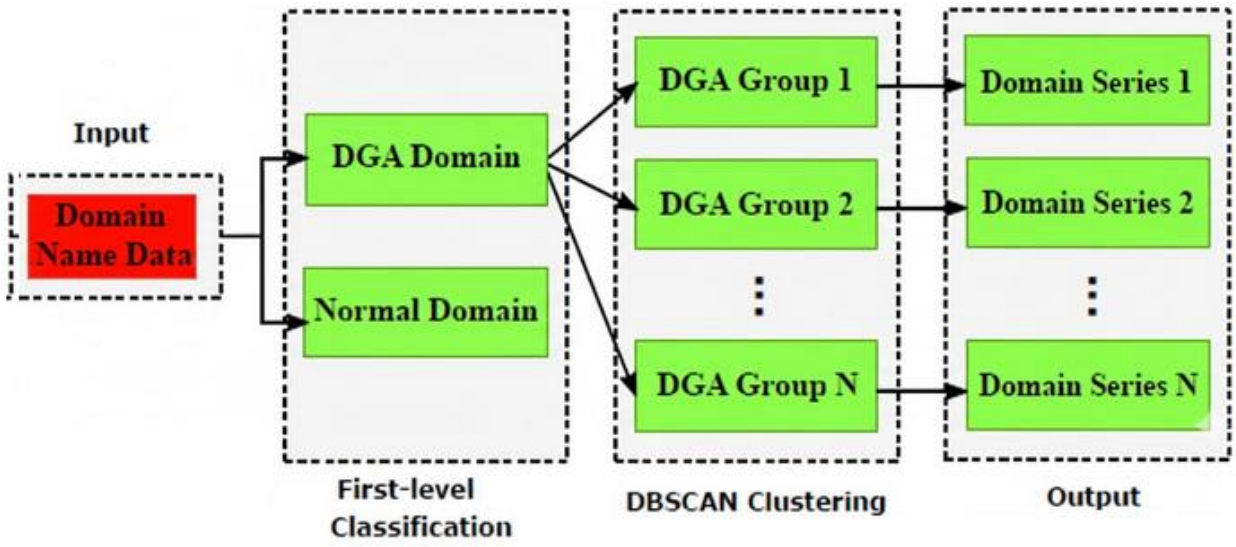



Рис. 2.4. Дворівнева модель класифікації та кластеризації

Лінгвістична відстань ($D_l$) обчислюється на основі шести лінгвістичних характеристик:

$$D_l(d_i, d_j) = \sqrt{\sum_{z=1}^{6} distance_z(d_i, d_j)},$$

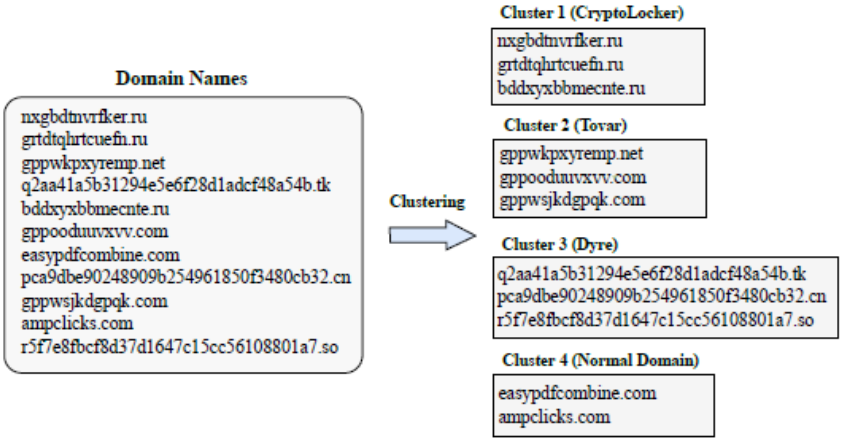
Подібність DNS (S) обчислюється через стовпцево-нормалізовану матрицю N:

$$\mathbf{S} = N^T \cdot N \in \mathbb{R}^{|L| \times |L|}$$

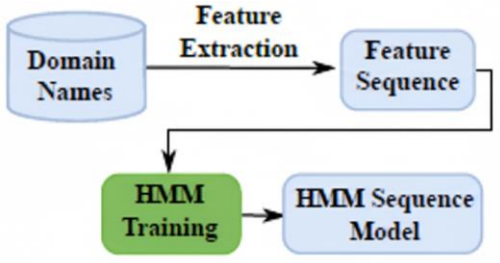
Загальна відстань (D) комбінує подібність DNS та лінгвістичну відстань:

$$D(d_i, d_j) = S_{d_i,d_j} + \log\left(\frac{1}{D_l(d_i, d_j)}\right)$$

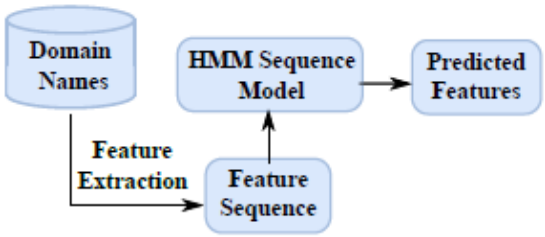
На основі загальної відстані та параметрів є і MinPts (встановлено 3) формуються кластери, що групують домени, згенеровані одним і тим же DGA.




а) Приклад групування доменних імен у множинні кластери




б) Приклад процедури навчання HMM




в) Прогнозування за допомогою моделі HMM

Рис. 2.5. Процедура роботи прогнозатора часових рядків на основі прихованої марковської моделі

Цей рисунок є ключовим для розуміння того, як фреймворк здійснює проактивне виявлення DGA-загроз. Він ілюструє, що після ідентифікації доменів DGA а), для кожного сімейства DGA (кластера) створюється індивідуальна модель HMM б), яка потім використовується для прогнозування характеристик наступних доменів у послідовності в). Цей механізм дозволяє системі блокувати ще не використані домени C2.

Перш ніж представити модель прогнозування часових рядів, ми наводимо приклад, який показує, що після кластеризації другого рівня доменні імена групуються в кілька кластерів, як показано на рисунку 2.5 а. З метою прогнозування характеристик майбутніх DGA-доменів, ми розробляємо модель прогнозування часових рядів на базі прихованої марковської моделі (HMM). Для кожного кластера DGA-доменів, отриманого на другому рівні, навчається окрема HMM-модель (рисунок 2.5 б).

Послідовність характеристик, витягнута з послідовності доменних імен, використовується для навчання моделі HMM.

Прогнозування (рисунок 2.5 с) - навчена HMM-модель застосовується для прогнозування майбутніх характеристик домену, які потім порівнюються з характеристиками нового спостережуваного DNS-запиту (рисунок 2.6).

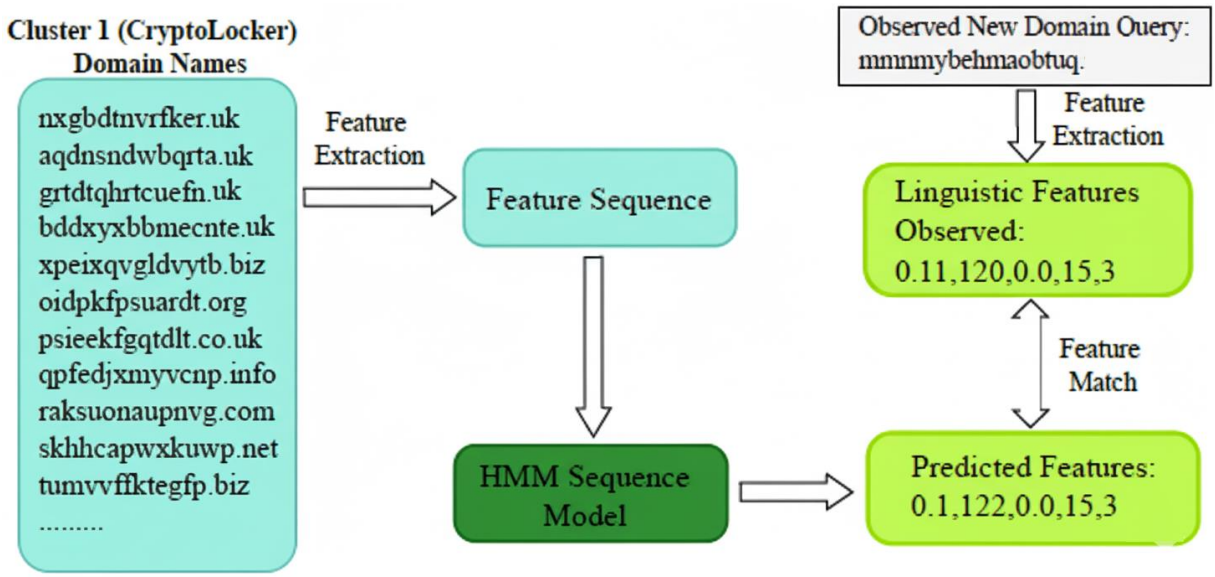



Рис. 2.6. Приклад прогнозування за допомогою моделі HMM

У моделі HMM передбачається, що кожен домен у момент часу t генерується прихованим станом $Z_t$. Використовується Марковський процес n-го порядку. Спільний розподіл прихованого стану та спостережуваних характеристик факторизується наступним чином:

$$P(Z_{1:T}, O_{1:T}) = P(Z_1)P(O_1|Z_1)\prod_{k=2}^{T} P(Z_k|Z_{k-1})P(O_k|Z_k)$$

Де Z — прихований стан, O — спостережувані характеристики. Довжина послідовності HMM (n) є критично важливою, оскільки вона визначає обсяг історичної інформації, доступної для виведення ймовірності наступного стану $Z_{t+1}$.

## Висновки до розділу

В даному розділі запропоновано DGA-базований фреймворк виявлення шкідливого ПЗ для розрізнення DGA-доменів від нормальних, ідентифікації алгоритму їх генерації та прогнозування ознак DGA за допомогою моделі часових рядів. Розроблено дворівневу модель (класифікація + кластеризація), де перший рівень ефективно ідентифікує DGA-домени, а другий рівень використовує DBSCAN для групування доменів за типом DGA. Створено HMM-прогнозатор часових рядів, який використовується для передбачення ознак DGA-доменів. Це дозволяє швидко блокувати DGA-домени, усуваючи ризик комунікації з C2-серверами під час перевірки DNS-запиту.

# РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ МЕТОДІВ ТА ІНСТРУМЕНТІВ ПРОТИДІЇ МЕРЕЖЕВИМ АТАКАМ ТА ФІШИНГУ ІЗ ЗАСТОСУВАННЯМ МЕТОДОЛОГІЙ МАШИННОГО НАВЧАННЯ

## 3.1. Модель глибокої нейронної мережі для класифікації DGA-доменів

Глибокі нейронні мережі (DNN) продемонстрували значні успіхи в обробці великих наборів даних у різноманітних прикладних галузях, включаючи комп'ютерний зір та обробку природної мови. У цьому розділі представлено фундаментальні концепції DNN та описано архітектуру запропонованого фреймворку глибокого навчання.

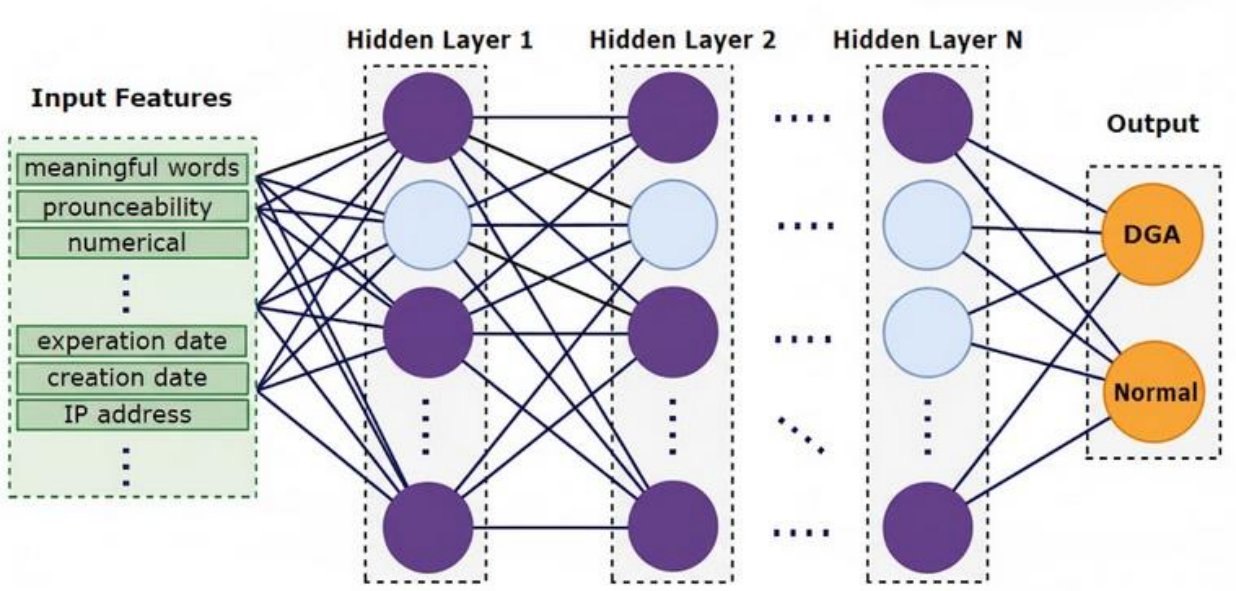



Рис. 3.1. Пропонована архітектура моделі глибокого навчання

DNN функціонально є поглибленою версією штучних нейронних мереж (ANN), що включає множинні приховані шари з кількома вузлами (нейронами) в кожному. Ця архітектура дозволяє DNN ефективно моделювати складні нелінійні взаємозв'язки. Для мінімізації функції втрат у DNN застосовуються спеціалізовані алгоритми оптимізації. З метою обробки великого масиву даних, ми розробили модель глибокого навчання для

класифікації DGA- та нормальних доменів, яку буде порівняно з раніше представленими методами машинного навчання. У поточному дослідженні розглядається виключно повністю з'єднана DNN (full-connected DNN). Архітектура моделі глибокого навчання ілюструється на рисунку 3.1.

### *3.1.1. Функція активації*

Для введення нелінійності до кожного прихованого шару застосовується нелінійна функція активації. Ця функція перетворює зважене значення кожного вузла попереднього шару перед його передачею на наступний шар.

Сигмоїдна функція перетворює зважену суму у значення в діапазоні [0,1]. Визначається як:

$$f_{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Одиниця випрямленого лінійного перетворення (ReLU, $f_{\mathrm{ReLU}}(x)$) встановлює нульове значення для всіх від'ємних вхідних даних і зберігає позитивні значення без змін:

$$f_{\mathrm{ReLU}}(x) = \max\{0, x\}$$

Функція ReLU є обчислювально більш ефективною та часто демонструє дещо кращі результати порівняно з гладкими функціями, такими як сигмоїдна. З огляду на ці переваги, у нашій моделі глибокого навчання застосовується активаційна функція ReLU.

### *3.1.2. Швидкість навчання*

Швидкість навчання є критично важливим гіперпараметром в алгоритмах оптимізації. Вона визначає величину кроку оновлення вектора

параметрів θ (ваг). Низька швидкість навчання забезпечує менші кроки вздовж градієнтного спуску, що запобігає пропуску локальних мінімумів, але значно подовжує час збіжності.

Загальне рівняння оновлення ваг:

$$w_{new} = w_{old} - lr \cdot g,$$

де w — вага, *lr* — швидкість навчання, а *g* — градієнт функції втрат.

Оптимальний вибір швидкості навчання має вирішальне значення. Результати аналізу ефективності різних значень швидкості навчання будуть представлені в наступному підрозділі.

*3.1.3. Алгоритми оптимізації*

Мінімізація функції втрат та оновлення тренованих параметрів (ваг) моделі DNN є ключовими завданнями глибокого навчання. Для виконання цієї мінімізації використовуються алгоритми оптимізації. Оскільки наша модель DNN виконує бінарну класифікацію (DGA або нормальний домен), вихідні дані прогнозу можуть бути інтерпретовані як значення ймовірності $p \in [0,1]$.

Для оцінки продуктивності моделі DNN застосовується функція логарифмічних втрат (*Log Loss*):

$$\text{LogLoss} = -(y \log(p) + (1 - y) \log(1 - p))$$

де y — істинний мічений вихід (1 для DGA, 0 для нормального домену), а p — прогнозована ймовірність.

У цьому дослідженні ми представляємо та порівнюємо три ключові алгоритми оптимізації:

1. Стохастична варіація градієнтного спуску, що оновлює параметр θ для кожного навчального екземпляра $(x^i, y^i)$.

$$\theta = \theta - lr \cdot g(\theta, x^i, y^i)$$

де $g(\theta, x^i, y^i)$ — градієнт функції логарифмічних втрат.

2. Градієнтно-орієнтований алгоритм, який адаптивно оновлює швидкість навчання для кожного параметра θ(i) на основі минулих градієнтів G(t,i). Це особливо корисно при роботі з розрідженими даними.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{lr}{\sqrt{G(t,i) + \epsilon}} g(t,i)$$

3. Подібно до Adagrad, Adam також адаптивно оновлює швидкості навчання. Він використовує оцінки першого моменту (середнього) та другого моменту (дисперсії) градієнтів M(t) та V(t) відповідно.

$$\theta_{t+1} = \theta_t - \frac{l}{\sqrt{V(t) + \epsilon}} M(t)$$

*3.1.4. Навчання та перевірка*

Для запобігання перенавчанню (overfitting) моделі DNN, ми розділяємо набір даних на навчальний та перевірочний (валідаційний) набори: 80% даних випадково обираються для навчання, а 20% — для перевірки. Навчання здійснюється виключно на навчальному наборі, тоді як оцінка продуктивності проводиться на перевірочному наборі.

Інші ключові параметри DNN включають:

- Епоху - одну повна ітерацію над усім навчальним набором даних.

- Розмір пакету (Batch Size, B) - кількість прикладів, які використовуються для одного кроку оновлення параметрів (для SGD часто B=1).

- Кількість кроків (Steps, S) - загальна кількість ітерацій навчання в межах однієї епохи. Розраховується як:

$$S = \frac{N}{B}$$

де N — загальна кількість навчальних прикладів.

### 3.2. Опис процесу проведення імітаційного моделювання

Реалізація методів машинного навчання та моделі глибокої нейронної мережі (DNN) здійснювалася із застосуванням інструментів з відкритим кодом, зокрема бібліотеки TensorFlow. В якості обчислювального середовища використовувався комп'ютерний кластер з високопродуктивними вузлами, що забезпечують можливості великомасштабних паралельних обчислень. Це обладнання критично необхідне для своєчасної обробки великих масивів даних у реальному часі.

Набори даних:

- для ретельної оцінки моделі використовувалися домени, згенеровані п'ятьма відомими сімействами DGA: CryptoLocker, Tovar, Dyre, Nymaim та Locky.

- контрольна група (легітимні домени). Для формування контрольної групи використовувався список топ-1 мільйона найпопулярніших інтернет-доменів.

Загалом у фреймворку було протестовано велику кількість доменних імен.

Конфігурація тестування наступна:

- Класифікація першого рівня. DGA-домени та контрольні домени змішувалися у співвідношенні 1:1 для бінарної класифікації.

- Кластеризація другого рівня (DBSCAN). Використовувалися лише домени, класифіковані як DGA на першому рівні, для їх групування за сімействами.

- Прогнозування HMM. Для кожного кластера DGA будувалася окрема модель HMM, використовуючи домени кластера як навчальні послідовності.

- Оцінка DNN. Для класифікації великих даних використовувалася модель DNN. Її продуктивність порівнювалася з результатами класифікації першого рівня на основі ML.

### 3.3. Технологія механізми захисту від фішингових атак

Фішингові атаки є однією з найбільш поширених форм кіберзагроз, головною метою яких є несанкціоноване вилучення конфіденційної інформації, зокрема облікових даних для входу, банківських реквізитів та паролів. У цих атаках зловмисники застосовують методи соціальної інженерії, маскуючись під довірених суб'єктів з метою обману користувачів та спонукання їх до взаємодії зі шкідливим контентом (електронними листами або текстовими повідомленнями).

Фішингові електронні листи залишаються домінуючим вектором атаки, що пояснюється широкою інтеграцією електронної пошти в повсякденну діяльність. Ключові індикатори, які використовуються для обману користувачів у фішингових повідомленнях, охоплюють три основні категорії:

1. Підозріла адреса відправника (невідповідність домену).

2. Наявність підозрілих гіперпосилань або вкладень.

3. Підозрілий або невідповідний контексту вміст повідомлення.

Хоча більшість фішингових листів містять легко помітні ознаки шахрайства, які можуть бути виявлені уважним користувачем, контекстна терміновість або важливість інформації в повідомленні підвищує ймовірність того, що користувачі ігноруватимуть ці очевидні підказки. Парадоксально,

але дослідження вказують, що користувачі з вищим рівнем досвіду у роботі з електронною поштою більш схильні стати жертвами фішингу.

### *3.3.1. Дослідження поведінкових механізмів користувачів під час фішингу*

Розуміння когнітивних та поведінкових патернів користувачів під час зіткнення з фішингом є фундаментальним для розробки ефективних захисних стратегій.

1. Незнання індикаторів безпеки.

Експериментальні дослідження [16] з використанням 20 учасників показали, що необізнаність щодо індикаторів безпеки веб-сайтів була основною причиною неправильного визначення фішингових ресурсів.

2. Індивідуальні звички та оперативність.

Дослідження [17] виявило кореляцію між звичною поведінкою роботи з поштою та сприйнятливістю. Користувачі, які мають звичку негайно відкривати електронні листи після сповіщення, демонструють вищу вразливість.

3. Суб'єктивна оцінка та недовіра.

Недовіра або ігнорування стандартних індикаторів безпеки на веб-сайтах є ще однією причиною вразливості. Користувачі часто приймають рішення про довіру, ґрунтуючись лише на візуальній естетиці та суб'єктивному сприйнятті електронного листа або веб-сайту.

Ці висновки підкреслюють необхідність вивчення факторів, що сприяють підвищенню обізнаності та довіри до захисних механізмів. Важливим напрямком є дослідження ролі втручання (intervention) [8] як фактора, що впливає на поведінку користувачів. Оскільки багато користувачів не розуміють значення індикаторів безпеки або не знають самого поняття фішингу, критично важливою є освіта користувачів щодо фішингу, а також навчання їх довіряти та усвідомлювати ці індикатори.

*3.3.2. Дослідження поведінки користувачів під час фішингових атак*

В умовах сучасності електронна пошта набула інтегрального значення у повсякденному та професійному житті, забезпечуючи широке спілкування через Інтернет. Це призводить до високої ймовірності того, що значна частина користувачів (зокрема співробітники та студенти) стикалася з ситуаціями ненавмисного натискання на посилання, що видається легітимним, але насправді є фішинговим.

З метою глибинного розуміння поведінкових патернів користувачів під час фішингових атак та формування ефективних програм навчання, представлено два типи досліджень: лабораторне та онлайн-дослідження.

Оцінюючи електронну пошту, користувачі фактично виконують завдання сортування: спочатку вони визначають легітимність листа, потім, навіть якщо його відкрито, оцінюють його вміст на надійність та корисність. Обидва типи досліджень імітують атмосферу відкриття, читання та прийняття рішення щодо електронної пошти.

Було використано спеціалізований емулятор електронної пошти, де користувачі сортували листи на дві категорії: «фішингові» та «не-фішингові» (легітимні). Від учасників вимагалося лише сортування, без необхідності відповідати на листи.

Джерела даних:

- Фішингові листи отримані з бази даних «Phish Bowl» [4] з необхідними модифікаціями для захисту особистої інформації.

- Легітимні листи - зібрані з реальних листів.

Лабораторне дослідження складалося з трьох раундів сортування пошти, кожен з яких містив унікальний набір листів.

Загальна кількість листів: 60 унікальних листів.

Фішингові: 45 листів (по 15 листів кожного з трьох типів фішингу).

Легітимні: 15 листів.

Мета завдання: диференціювати фішингові листи від легітимних шляхом їх сортування.

Емулятор електронної пошти включав три основні компоненти (рисунок 3.2):

- RoundCube Email Client - браузерний IMAP-клієнт, що слугував інтерфейсом для користувачів для перегляду та сортування листів.

- Postfix Virtual Mail Server - віртуальний поштовий сервер, що забезпечував можливість розміщення доменів, через який надсилалися попередньо завантажені листи.

- BurpSuite Proxy Listener - використовувався як HTTP-проксі для перехоплення, інспектування та модифікації HTTP-запитів та відповідей між RoundCube та Postfix, дозволяючи захоплювати трафік та зберігати логи у форматі XML для подальшого аналізу.

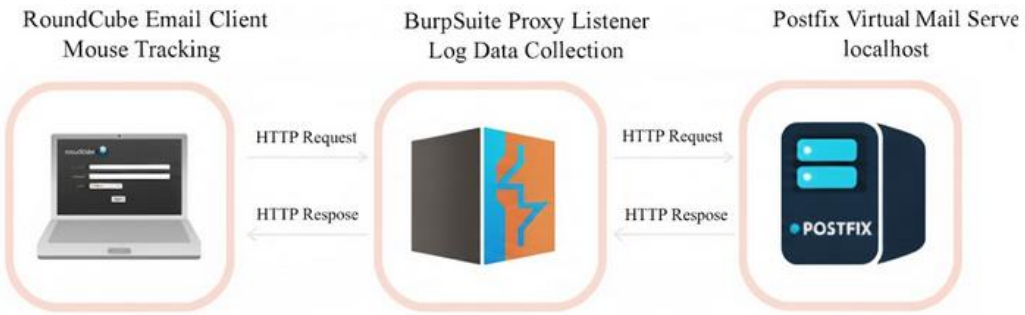



Рис. 3.2. Архітектура емулятора електронної пошти для виявлення фішингу

Крім того, був розроблений Python-код для відстеження рухів миші учасників, включаючи координати розташування та тривалість перебування в певних точках. Для оцінки нерішучості (hesitation) фіксувався час t (у секундах), протягом якого курсор залишався нерухомим. Якщо t перевищував встановлений поріг α, лічильник нерішучості h збільшувався на 1.

У дослідженні було виділено та протестовано три різні типи фішингових атак, щоб визначити, який із них є найбільш ефективним у

схилянні користувачів до помилкового сортування фішингових листів як «не-фішингових».

1. Підозріла адреса відправника - атака, заснована на підробці адреси. Зловмисники маніпулюють адресою (наприклад, paypal@on1ine.service.org), користуючись тим, що увага користувачів зазвичай сфокусована на вмісті, а не на деталях відправника.

2. Підозрілі посилання або вкладення - атака, що містить шкідливі гіперпосилання (з маніпуляціями, орфографічними помилками або неіснуючими адресами) або зловмисні вкладення (виконувані файли, PDF), які можуть інсталювати шкідливе програмне забезпечення.

3. Зловмисний вміст електронного листа - фішинг, що використовує приховані елементи або низько помітні помилки (орфографічні, граматичні), які важко ідентифікувати, особливо для користувачів, які не є носіями мови.

### *3.3.3. Використання машинного навчання для прогнозування*

Метою застосування машинного навчання є прогнозування кінцевої продуктивності користувача (успішно/неуспішно) під час фішингової атаки.

Продуктивність класифікується на дві бінарні категорії: 'Good' (Добре) та 'Poor' (Погано). Ця класифікація базується на критичній точці c (порогове значення загального балу):

$$\text{Performance Class} = \begin{cases} \text{'Good'}, & \text{якщо Score} \geq c \\ \text{'Poor'}, & \text{якщо Score} < c \end{cases}$$

Оскільки дослідження даних має велику кількість атрибутів при відносно невеликій кількості екземплярів, що ускладнює класифікацію, виникає потреба у скороченні розмірності та ретельному відборі ознак.

Для підвищення ефективності моделі та подолання проблеми "малого набору даних з великою кількістю ознак" застосовувався поетапний відбір:

- аналіз кореляції Персона використовувався для оцінки важливості кожного окремого атрибута щодо загального показника продуктивності.

- лінійна регресія та прямий відбір - атрибути були включені в модель лінійної регресії. Для відбору найбільш значущих ознак застосовувався метод прямого поетапного відбору, де ознаки додавалися до моделі по одній, ґрунтуючись на пороговому значенні p-value (встановленому на рівні p=0.25).

Для прогнозування продуктивності на основі скороченого набору ознак (16 атрибутів) було побудовано чотири різні моделі ML:

- Decision Tree (J48)

- Naive Bayes (NB)

- Support Vector Machine (SVM)

- Multilayer Perceptron (MP)

Для забезпечення робастності моделі та обліку малої вибірки використовувався метод перехресного відбору ознак (cross-selection):

- з m=16 відібраних ознак, випадковим чином вибиралися n=12 ознак (що становить 75%×m).

- цей процес повторювався k=4 рази (k=⌈m−nm⌉=⌈16−1216⌉=4), забезпечуючи, що кожна ознака має ймовірність бути обраною ≥99.6%.

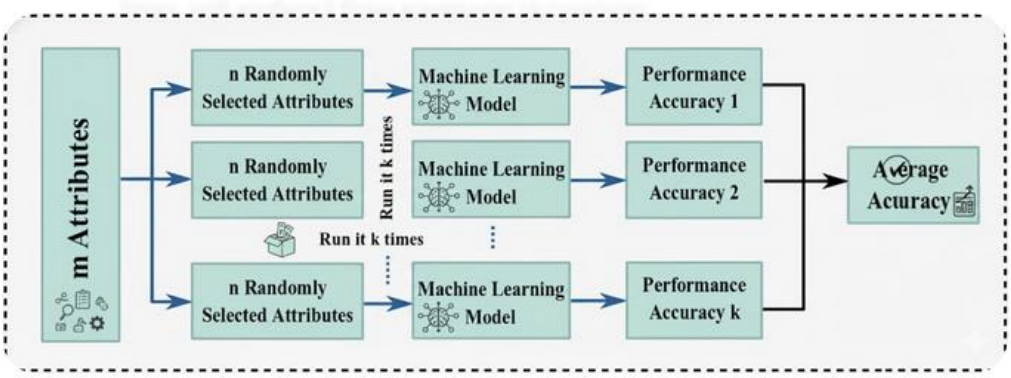



Рис. 3.3. Процедура використання машинного навчання визначення фішингу

Процедура (рис. 3.3) включала: випадковий вибір n атрибутів → навчання моделі ML з 10-кратною перехресною валідацією → розрахунок точності. Кінцева точність є усередненим значенням k запусків.

Визначення оптимальної критичної точки c є вирішальним кроком для ефективної класифікації 'Good'/'Poor'. Для цього використовувався жадібний метод (greedy method):

- система перевіряла всі можливі значення критичної точки.

- вибиралася та критична точка c, яка забезпечувала найкращу точність прогнозування у класифікаційній моделі.

Цей підхід гарантує, що поріг c не є довільним, а є емпірично визначеним для максимізації точності моделі.

*3.3.4. Оцінка результатів*

Перед застосуванням моделей ML для бінарної класифікації ('Good'/'Poor') необхідно визначити оптимальну критичну точку c (поріг балу). Оцінка проводилася з використанням 10-кратної перехресної валідації на 16 відібраних атрибутах.

Таблиця 3.1.

Результати застосування алгоритмів

| **Модель ML** | **Оптимальна c** | **Точність (%)** |
|---|---|---|
| J48 | 32 | 97.78% |
| Multilayer Perceptron | 30 | 96.67% |
| SVM | 30 | 92.22% |
| Naive Bayes | 31 | 88.89% |

Висновок наступний:

- Оптимальна c: хоча висока точність досягається при низьких значеннях $c \in [15;20]$, ці значення призводять до незбалансованого розподілу класів ('Good' vs. 'Poor').

- Найкраща прогностична точність досягається при c=32 для J48 (97.78%) та c=30 для SVM і MP, забезпечуючи більш обґрунтований поділ вибірки.

Нижче проаналізовано, чому модель J48 (Decision Tree), яка є відносно простою, перевершила більш складні моделі, такі як Multilayer Perceptron

(MP) та Support Vector Machine (SVM), у завданні прогнозування продуктивності користувачів на фішингових атаках.

У класифікаційній задачі, що ґрунтується на невеликому наборі даних (90 екземплярів) з відібраними атрибутами (16 ознак), J48 проявила виняткову продуктивність (97,78%) завдяки кільком ключовим факторам:

1. Характеристики даних та відбір ознак

J48 чудово працює з даними, які можна чітко розділити за допомогою порогових значень (що і робить дерево рішень). Після поетапного відбору атрибутів у моделі залишилися ознаки з високою кореляцією з кінцевим балом. Ці ознаки є прямими і сильними індикаторами кінцевого результату. Ці ознаки, ймовірно, дозволяють лінійно розділити простір даних. Наприклад, якщо phishing accuracy > певного порогу, клас майже завжди 'Good'. Дерево рішень ефективно знаходить ці оптимальні пороги. Моделі, як-от J48, часто показують відмінні результати на малих наборах даних, де більш складні моделі (наприклад, DNN/MP) можуть не мати достатньо даних для ефективного навчання своїх численних параметрів, що призводить до гіршої узагальнюючої здатності.

2. Ефективність бінарної класифікації

Завдання було зведено до бінарної класифікації ('Good' або 'Poor') з оптимізованою критичною точкою (c=32):

- Точка c=32, обрана жадібним методом, гарантує, що класи 'Good' та 'Poor' є максимально роздільними на основі наявних атрибутів.

- J48 інтуїтивно використовує атрибути для створення серії простих правил "якщо-то" (наприклад: Якщо P. Accuracy > X I Sort Agreement 4 < Y → Good). Оскільки оптимальні розділяючі пороги вже існують у відібраних ознаках, J48 будує прозорий і високоточний класифікатор.

3. Обмеження складних моделей

Нейронні мережі вимагають великої кількості даних для тонкого налаштування ваг і вивчення складних нелінійних залежностей. Якщо залежність у даних є простою або лінійно роздільною, MP часто

перенавчається на невеликій вибірці або просто не може знайти більш точного рішення, ніж просте дерево.

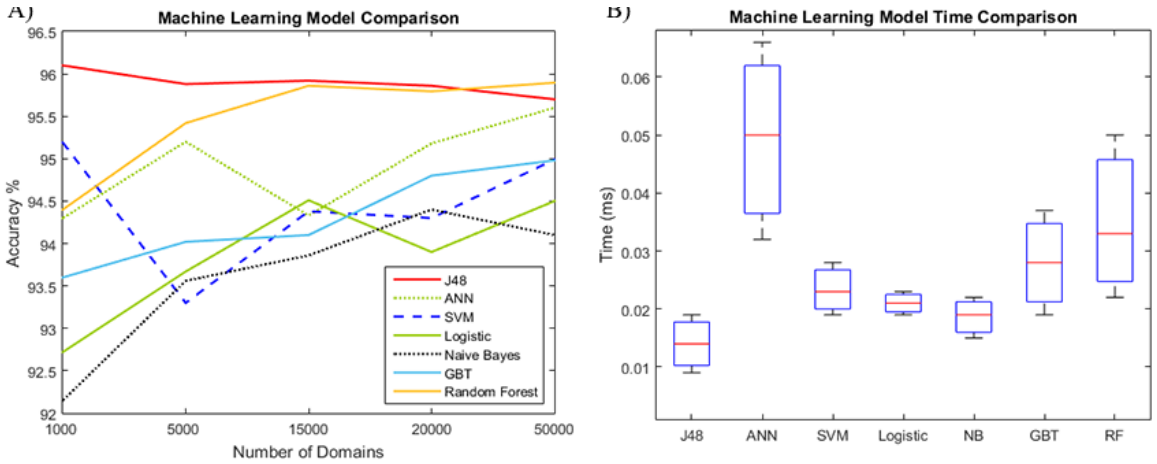
SVM ефективний при високій розмірності, але для досягнення високої точності він сильно залежить від правильного вибору функції ядра (kernel) та параметрів регуляризації. У цій задачі, ймовірно, оптимальна гіперплощина, знайдена SVM, виявилася менш ефективною, ніж ієрархічні пороги J48.

Таким чином, у даному контексті — прогнозування поведінки користувача на основі сильно корельованих поведінкових та демографічних ознак — J48 виявився найкращим прогностичним інструментом завдяки своїй здатності знаходити прості, але високоефективні правила поділу класу.

## 3.4. Результати оцінки запропонованого фреймворку

### *3.4.1. Класифікація першого рівня (ML-моделі)*

Для визначення найбільш оптимальної моделі класифікації було протестовано сім алгоритмів ML (J48, ANN, SVM, логістична регресія, наївний байєс, дерево градієнтного бустингу, випадковий ліс). Оцінка проводилася за допомогою 10-кратної перехресної перевірки (90% для навчання, 10% для перевірки) і подано на рисунку 3.4.



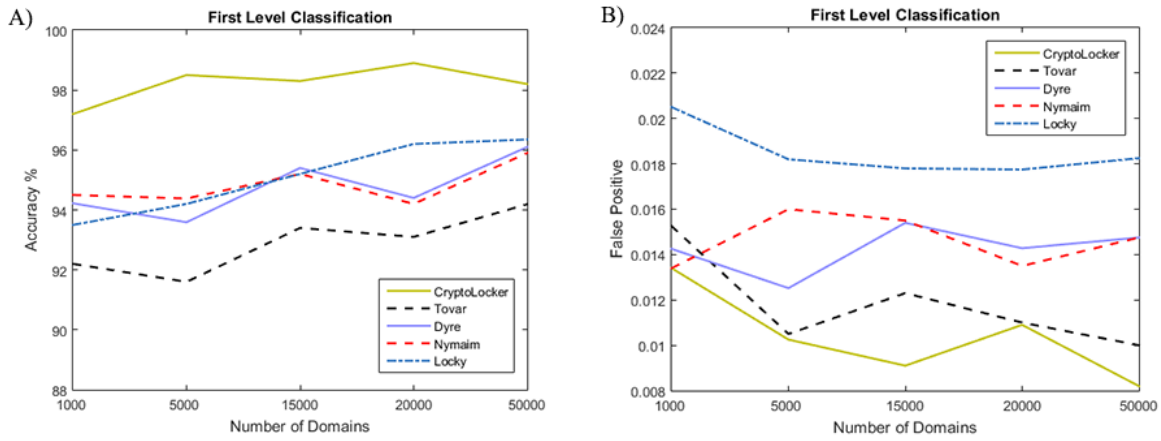

а) б)

Рисунок 3.4. Порівняння різних алгоритмів машинного навчання
а) Точність різних алгоритмів МН, б) Час класифікації різних алгоритмів МН

Точність: Random Forest продемонстрував найкращу середню точність — 95,47%.

Швидкість: Алгоритм J48 виявився найшвидшим (середній час 0,0144 мс на класифікацію).

Аналіз продуктивності J48 (масштабованість): продуктивність J48 тестувалася на п'яти групах зразків DGA-доменів (від 1000 до 50000 доменних імен).




а) б)

Рис. 3.5. Результати класифікації за допомогою J48.

а) Точність класифікації J48 для кожного домену DGA з різними обсягами даних б) Рівень хибнопозитивних спрацьовувань класифікації J48

Найвища точність: J48 найкраще працював для доменів CryptoLocker, досягаючи середньої точності 98,22% (пікове значення 98,9%), тоді як точність для інших DGA варіювалася від 92% до 95% (рисунок 3.5 а).

Рівень хибнопозитивних результатів (False Positive Rate, FPR): для CryptoLocker FPR становив найнижче значення — 0,010. (рис. 3.5 б).

*3.4.2. Кластеризація другого рівня (DBSCAN)*

Для кількісної оцінки успіху кластеризації використовувалася оцінка точності (accuracy score), де міткою кластера обиралася найпоширеніша мітка DGA серед його елементів.

Проведемо порівняння метрик відстані:

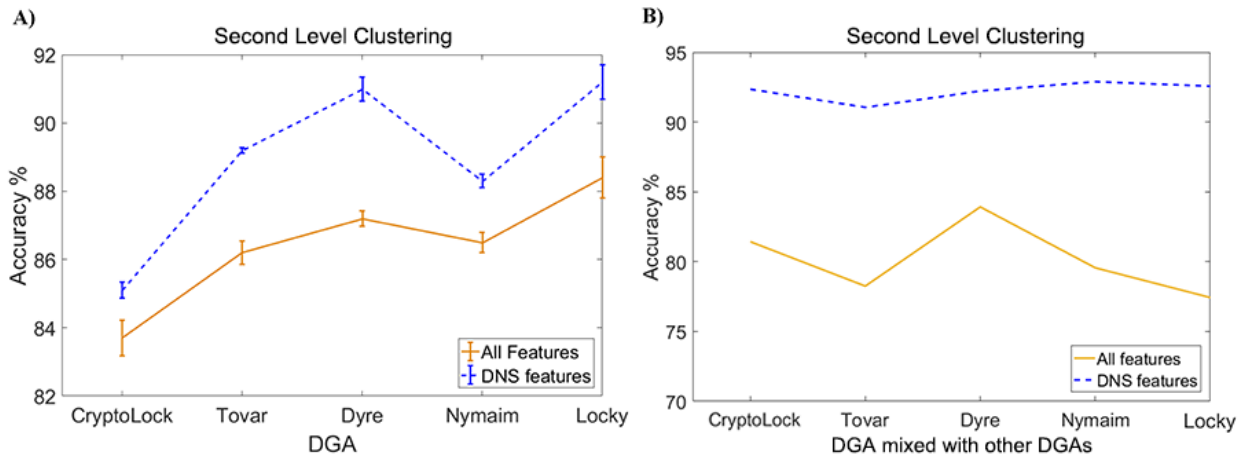
- Використання лінгвістичної відстані та подібності DNS як загальної метрики: середня точність 87,64%.

- Використання лише подібності DNS як загальної метрики: середня точність 89,02%.

Низька ефективність лінгвістичних ознак у кластеризації пояснюється тим, що більшість DGA мають схожий склад рядків та довжину, що знижує їхню дискримінаційну здатність.

Кластеризація двох змішаних груп: тестування всіх попарних комбінацій DGA.

При змішуванні CryptoLocker з іншими DGA, середня точність кластеризації при використанні всіх характеристик становила 81,43%.




а) б)

Рис. 3.6. Експериментальні результати кластеризації

а) Точність кластеризації для кожного DGA

б) Точність кластеризації для кожних двох груп DGA

Рисунок 3.6 а показує, як алгоритм кластеризації другого рівня працює з різними DGA. При використанні лише DNS-характеристик як відстані DBSCAN, точність зросла до 92,45% (для CryptoLocker), що показано на рисунку 3.6 б.

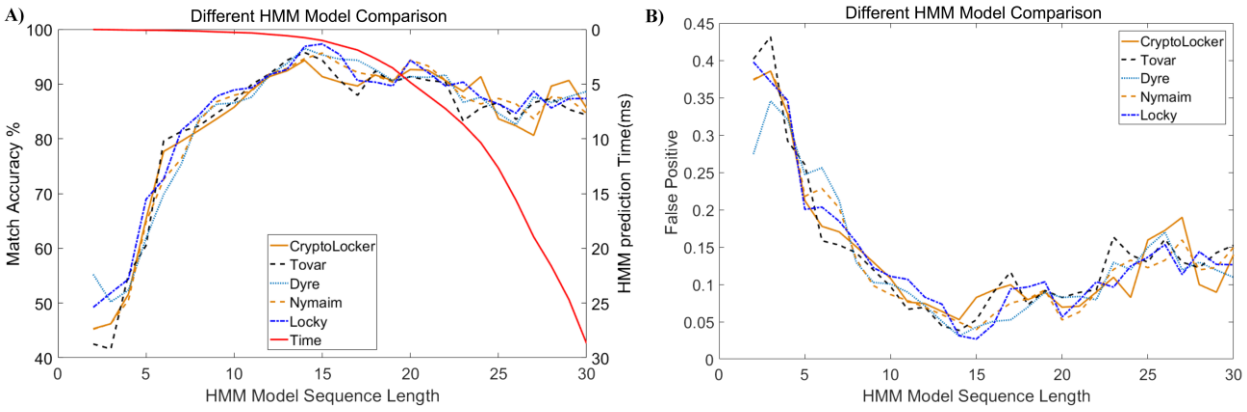
Точність для інших сімейств: Tovar (91,05%), Dyre (92,22%), Nymaim (92,89%), Locky (92,57%).

Висновки щодо кластеризації наступні: висока точність кластеризації (понад 91% при використанні DNS-характеристик) підтверджує, що модель ефективно групує ідентичні DGA-домени, створюючи високоякісні навчальні дані для подальшої моделі HMM-прогнозування.

### *3.4.3. Модель прогнозування часових рядів*

Досліджувався вплив довжини послідовності HMM (n) на точність відповідності та час виконання. Тестувалася довжина послідовності від n=2 до n=30.

Оптимальна довжина послідовності: пікова середня точність 95,21% була досягнута при довжині послідовності $n \in [14,15]$ (рисунок 3.7 а).




а) б)

Рис. 3.7. Експериментальні результати використання моделі прогнозування HMM. а) Точність відповідності для моделей HMM з різною довжиною послідовності. б) Рівень хибнопозитивних спрацьовувань для моделей HMM з різною довжиною послідовності.

Продуктивність та час виконання:

- При n=15, середній час прогнозування становив лише 1,02 мс.
- При n=20, час різко зростав до 4,85 мс (збільшення в 3,75 рази).

- Рівень хибнопозитивних результатів (FPR) досягнув найнижчого значення — 0,045 при n=15 (рисунок 3.7 б).

Висновки щодо HMM наступні. Модель HMM з оптимальною довжиною послідовності (15) забезпечує високу точність (95,21%) та винятково швидкий час виконання (1,02 мс), що є критично важливим для превентивного блокування DGA-доменів у реальному часі до встановлення DNS-з'єднання.

*3.4.4 Порівняння DNN та класифікації першого рівня*

Точність DNN при навчанні та тестуванні доменних імен, модель DNN досягла точності 97,79%.

Як висновок, можна констатувати, що модель DNN значно перевершує класифікацію першого рівня на основі ML (95,47%), підтверджуючи її перевагу в обробці великих даних та виявленні складних DGA-патернів.

## 3.5. Аналіз продуктивності та оптимізація глибокого навчання

*3.5.1. Налаштування та порівняння класифікаторів*

Для досягнення оптимальної продуктивності моделі DNN проводилася ітераційна оцінка та ручна оптимізація гіперпараметрів, включаючи кількість нейронів та прихованих шарів. Кінцевий вибір параметрів ґрунтувався на максимізації показників точності (Accuracy), логарифмічних втрат (Log Loss) та площі під ROC-кривою (AUC).

Продуктивність запропонованої DNN-моделі порівнювалася з двома основними бенчмарками: класифікатором J48 (представленим як найкраща ML-модель першого рівня) та моделлю LSTM (Long Short-Term Memory).

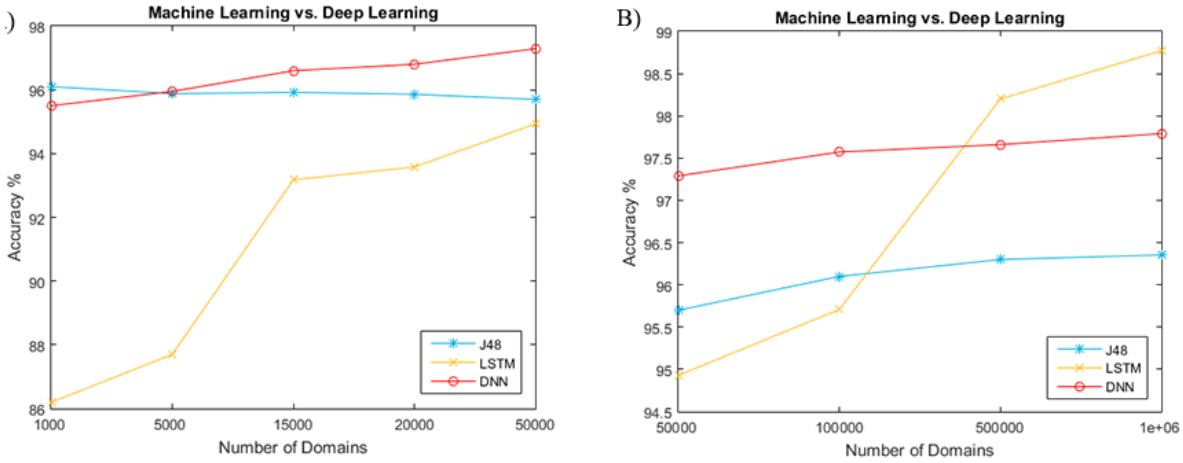
А. Порівняння Точності Класифікаторів за Обсягом Даних

Експериментальне порівняння точності J48, LSTM та DNN проводилося на поступово зростаючих обсягах даних (рисунок 3.8 а, рис. 3.8 б).

Таблиця 3.2.

Точність моделей МН

| **Обсяг Даних** | **J48 (Середня Точність)** | **LSTM (Середня Точність)** | **DNN (Середня Точність)** |
|---|---|---|---|
| До 50 тис. доменів | 95,89% | 91,12% | 96,43% |
| До 1 млн доменів | 96,35% (max) | 98,77% (max) | 97,79% (max) |
| Середня точність (50 тис. - 1 млн) | - | 96,9% | 97,58% |




а) б)

Рис. 3.8. Порівняння точності J48, LSTM та DNN.

а) Порівняння точності J48, LSTM та DNN з кількістю доменів від 1000

б) Порівняння точності J48, LSTM та DNN з великою кількістю доменів від 50 тис.

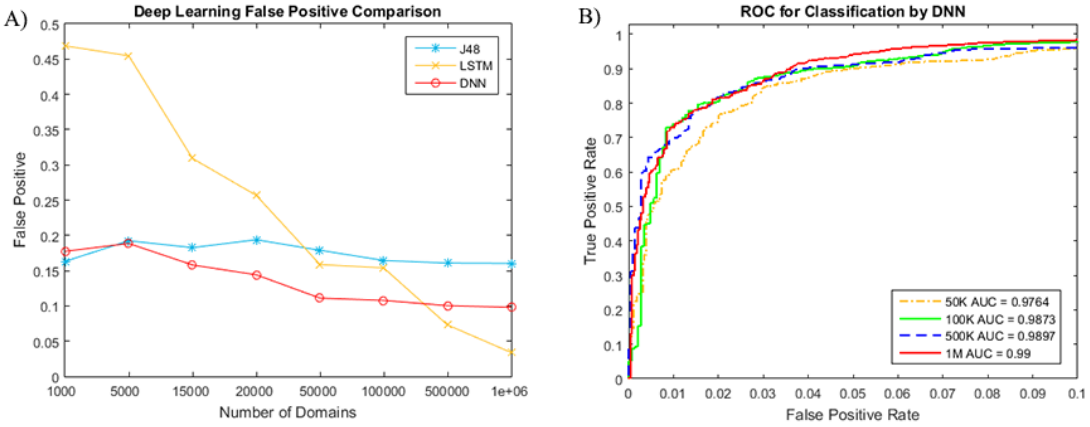
Отже, для малих обсягів даних (до 5000 доменів) J48 демонструє конкурентну або вищу точність, що вказує на ефективність ручної інженерії ознак при обмежених вибірках.

Зі зростанням обсягу даних (понад 5000 доменів) точність DNN починає стабільно перевершувати J48, підтверджуючи її перевагу в автоматичному вивченні ознак.

Модель LSTM (спеціалізована для послідовних даних) досягає найвищої пікової точності (98,77%) на найбільших обсягах, але її середня точність виявилася нижчою за DNN-модель. Запропонована DNN-модель демонструє найкращу середню продуктивність (97,58%) на всьому діапазоні великих наборів даних.

Б. Порівняння рівня хибнопозитивних результатів (FPR)

FPR є критичним показником для систем безпеки (менше значення — краще), рис. 3.9 а.




а) б)

Рис. 3.9. Експериментальні результати моделі DNN. а) Порівняння хибнопозитивних результатів серед J48, LSTM та DNN. б) ROC-криві серед різної кількості доменів: 50K, 100K, 500K та 1M.

Малі обсяги (менше 50 тис.): DNN та J48 демонструють кращі показники FPR, ніж LSTM.

Великі обсяги (понад 50 тис.): FPR моделі LSTM швидко знижується, але в цілому модель DNN демонструє кращу стійкість та продуктивність при обробці наборів даних усіх розмірів.

*3.5.2. Оцінка продуктивності DNN та запобігання перенавчанню*

Для більш глибокої оцінки якості DNN використовувалися додаткові метрики, що показані в таблиці та механізми контролю перенавчання.

Таблиця 3.3.

Метрики машинного навчання

| **Кількість доменів** | **Точність (LSTM/DNN)** | **Повнота (LSTM/DNN)** | **AUC (LSTM/DNN)** | **Час (с) (LSTM/DNN)** |
|---|---|---|---|---|
| 1 млн | 0.8924 / 0.9652 | 0.9914 / 0.9913 | 0.9900 / 0.9990 | 23154.35 / 22184.56 |
| 500 тис. | 0.8894 / 0.9218 | 0.9903 / 0.9915 | 0.9897 / 0.9970 | 18185.41 / 12161.82 |

Модель DNN демонструє значно вищий показник AUC (0.9990) та менший час навчання на великих обсягах (на 500 тис. доменів час DNN на 33% менший), що підтверджує її ефективність.

Розділення даних на навчальний та перевірочний набори та моніторинг логарифмічних втрат показали, що втрати перевірочного набору (Validation Log Loss) тісно збігаються з втратами навчального набору (рисунок 3.10). Це свідчить про успішне запобігання перенавчанню.

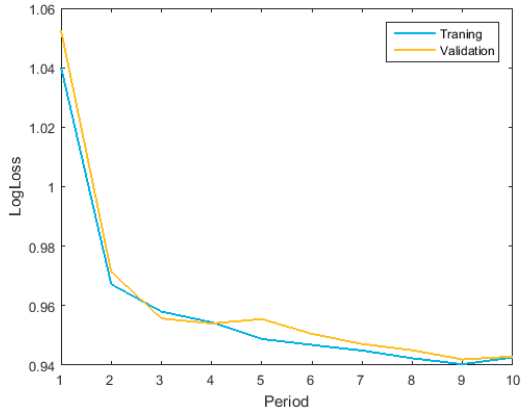
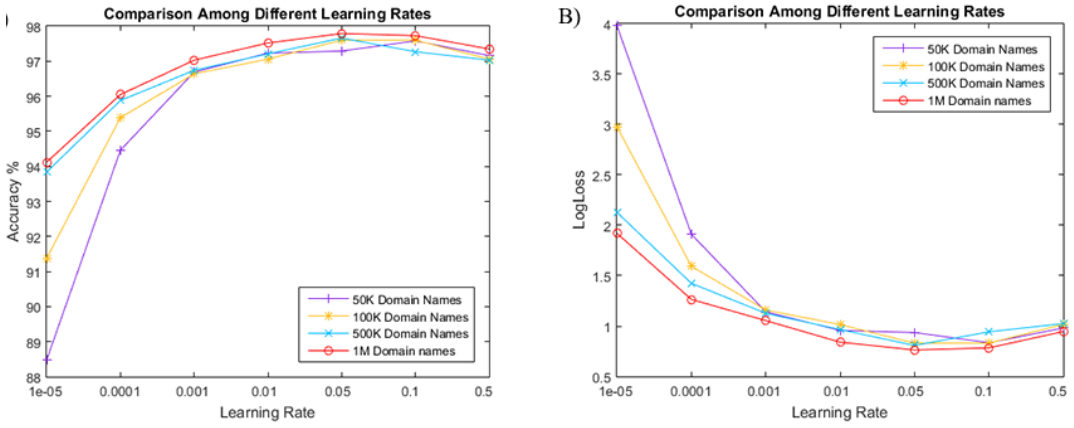



Рис. 3.8. Логарифмічна втрата під час навчання та валідації протягом періодів

Тестування різних швидкостей навчання ($lr \in \{0.00001;\ldots;0.5\}$) показало:

- При lr=0.05 та lr=0.1 середня точність є найвищою (97,58% та 97,54% відповідно) (рисунок 3.11 а).

- Середні логарифмічні втрати є найнижчими при lr=0.05 (0.83415) (рисунок 3.11 б).

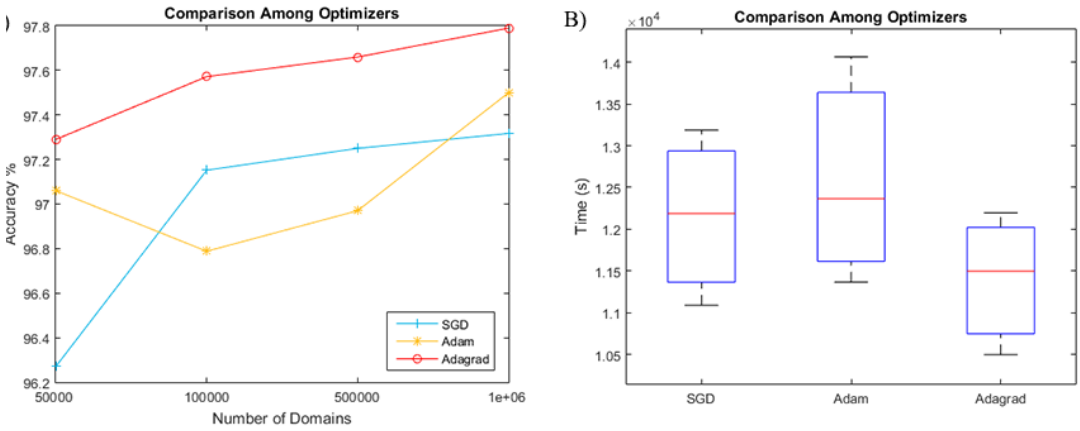



а) б)

Рис. 3.11. Порівняння різних швидкостей навчання. а) Порівняння точності з різними швидкостями навчання: 0.00001, 0.0001, 0.001, 0.01, 0.05, 0.1 та 0.5. б) Порівняння логарифмічних втрат з різними швидкостями навчання: 0.00001, 0.0001, 0.001, 0.01, 0.05, 0.1 та 0.5.

Значення lr=0.05 було обрано як оптимальне для подальших експериментів.

Порівнювалися три алгоритми оптимізації: стохастичний градієнтний спуск (SGD), Adam та Adagrad. (рисунок 3.12 а та б).




а) б)

Рис. 3.12. Порівняння різних алгоритмів оптимізації. (А) Порівняння точності між різними алгоритмами оптимізації: стохастичний градієнтний спуск, Adam та Adagrad. б) Порівняння часу між цими різними алгоритмами оптимізації, де Adagrad має найменший використаний час

Adagrad показав найвищу середню точність (97,578%) і використовував найменше часу на навчання (середній час 11496,49 с), що робить його найбільш ефективним для побудови DNN-моделі.

Отже, хоча Adam є популярним, для даного набору даних DGA-класифікації Adagrad виявився найкращим оптимізатором, забезпечуючи найвищу точність при мінімальному часі збіжності. Середній час прогнозування одного домену DNN-моделлю становить близько 27,87 мс.

## Висновки до розділу

У третьому розділі представлено практичну реалізацію розроблених методів та інструментів протидії мережевим атакам із застосуванням машинного навчання. Побудовано модель глибокої нейронної мережі (DNN) для класифікації доменів, що демонструє високу точність виявлення DGA-активності. Проведено серію експериментів із підбору функцій активації, алгоритмів оптимізації та швидкості навчання, що забезпечили оптимальну продуктивність моделі. Виконано імітаційне моделювання сценаріїв фішингових атак із залученням поведінкових даних користувачів, на основі яких побудовано систему прогнозування реакцій користувачів. Доведено, що поєднання поведінкових моделей із методами машинного навчання дозволяє зменшити кількість успішних фішингових атак. Реалізовано багаторівневий фреймворк, що включає класифікацію першого рівня (ML-моделі), кластеризацію другого рівня (DBSCAN) та прогнозування часових рядів для аналізу динаміки атак.

# ВИСНОВКИ

В магістерській роботі розглянуто здійснено дослідження сучасних підходів до забезпечення мережевої безпеки шляхом розроблення та впровадження систем протидії мережевим атакам із використанням методів машинного та глибокого навчання. Робота поєднує теоретичний аналіз уразливостей і загроз інформаційних систем, розробку моделей виявлення шкідливої активності, а також практичну реалізацію алгоритмів і фреймворків, спрямованих на підвищення ефективності кіберзахисту.

У першому розділі проведено ґрунтовний аналіз предметної області дослідження. Розкрито сучасний стан і проблематику мережевої безпеки, зокрема ідентифіковано основні вразливості систем доменних імен (DNS), бездротових мереж (Wi-Fi, WPA2), а також методи реалізації фішингових атак. Визначено, що традиційні засоби протидії мережевим загрозам поступово втрачають ефективність у зв’язку з динамічним розвитком автоматизованих атак і появою нових типів загроз, зокрема змагальних атак на моделі машинного навчання. Проведено класифікацію основних категорій мережевих атак, включно з атаками на протоколи автентифікації, шкідливими DGA-доменами (Domain Generation Algorithms) та соціотехнічними атаками. Проаналізовано особливості змагальних (adversarial) атак, що становлять значну небезпеку для систем глибокого навчання, а також досліджено поведінкові та психологічні чинники вразливості користувачів до фішингу.

У другому розділі обґрунтовано методологію виявлення шкідливого програмного забезпечення, що використовує механізм генерації доменів (DGA). Детально розглянуто архітектуру фреймворку, спрямованого на автоматичну ідентифікацію DGA-доменів на основі аналізу лінгвістичних і статистичних ознак. Показано обмеження традиційних підходів (фільтрація, чорні списки, сигнатурний аналіз), які не здатні виявляти нові або модифіковані доменні імена, що генеруються динамічними алгоритмами. У

роботі запропоновано вдосконалену методологію збору даних і побудови наборів ознак для навчання моделей, що забезпечує підвищення точності класифікації нових доменів.

Розроблено структуру фреймворку виявлення шкідливого ПЗ, що включає модулі збору даних, попередньої обробки, екстракції ознак, класифікації та візуалізації результатів. У процесі дослідження розглянуто методи побудови моделей загроз, що дозволяють оцінити ризики атак і визначити ключові параметри, за якими здійснюється виявлення DGA-доменів.

У третьому розділі реалізовано практичну імплементацію методів машинного навчання для виявлення та протидії мережевим атакам. Створено модель глибокої нейронної мережі (Deep Neural Network, DNN) для класифікації DGA-доменів із використанням оптимальних функцій активації, алгоритмів оптимізації (Adam, RMSProp) та стратегій запобігання перенавчанню. Проведено навчання й тестування моделі на реальних вибірках доменних даних, що продемонструвало високу ефективність у порівнянні з базовими алгоритмами машинного навчання.

Крім цього, здійснено моделювання механізмів протидії фішинговим атакам шляхом аналізу поведінкових аспектів користувачів і застосування моделей прогнозування. Визначено, що поєднання поведінкової аналітики з алгоритмами машинного навчання дозволяє створювати системи адаптивного навчання користувачів, здатні ідентифікувати фішингові загрози на ранніх етапах взаємодії.

Розроблений фреймворк включає багаторівневу архітектуру: класифікацію першого рівня за допомогою ML-моделей, кластеризацію другого рівня з використанням DBSCAN для ідентифікації невідомих шаблонів атак, а також модуль прогнозування часових рядів для аналізу динаміки загроз. Проведено порівняльну оцінку результатів, що показала перевагу DNN-моделі над традиційними класифікаторами за всіма ключовими метриками.

Основні наукові результати роботи полягають у систематизації сучасних методів протидії мережевим атакам та побудові класифікації загроз; розробці архітектури фреймворку виявлення шкідливих доменів на основі алгоритмів глибокого навчання; удосконаленні методів протидії фішинговим атакам на основі поведінкових моделей і машинного прогнозування.

Отримані результати мають практичну цінність для розробки систем моніторингу й реагування на кіберзагрози, а також можуть бути використані в подальших наукових дослідженнях із напрямів кіберзахисту, безпеки штучного інтелекту та інтелектуального аналізу даних.

# ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1. Williams and J. Li. (2017). "Phishing attack and user behavior: A correlation study.", p.p. 110 - 191
2. Brase, S. A. (2009). Understanding the effects of monetary incentives on task performance. Journal of Applied Psychology, 94(2), 521-535.
3. Gupta, S., (2016). Phishing: A survey of taxonomy, techniques, countermeasures and challenges. Journal of Cyber Security, 5(1), 1-15.
4. Harrison, J. E.. (2016). Judging a website by its cover: users' perceptions of phishing attacks. Journal of Usability Studies, 11(4), 168-185.
5. Vanhoef, M., & Piessens, F. (2017). Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2 (KRACK). In proceedings / technical report.
6. Woodbridge, J., Anderson, H. S., et al. (2016). Predicting Domain Generation Algorithms with Long Short-Term Memory Networks.
7. Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Dagon, D., Lee, W., ... & Feamster, N. (2011). Detecting malware domains at the upper DNS hierarchy. Proceedings of the 20th USENIX Security Symposium, San Francisco, CA, USENIX Association, pp. 1–16.
8. Woodbridge, J., Anderson, H. S., Ahuja, A., & Grant, D. (2016). Predicting Domain Generation Algorithms with Long Short-Term Memory Networks. arXiv preprint arXiv:1611.00791.
9. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR) . arXiv:1412.6572.
10. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations (ICLR), arXiv:1312.6199.
11. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. In

Proceedings of the 1st IEEE European Symposium on Security and Privacy Workshops (EuroS&P Workshops), IEEE. arXiv:1511.07528.

12. Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition, 84, 317–331.
13. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, San Jose, CA, IEEE, pp. 305–316.
14. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise (DBSCAN). In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR, AAAI Press, pp. 226–231.
15. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR). arXiv:1412.6980.
16. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(56), 1929–1958.
17. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press, Cambridge, MA.
18. Hochreiter, S., & Schmidhuber, J. (2007). Long short-term memory. Neural Computation, 9(8), 1735–1780.
19. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.
20. Dalvi, N., Domingos, P., Mausam, Sanghai, S., & Verma, D. (2004). Adversarial classification. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Seattle, WA, ACM, pp. 99–108.
21.

22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
23. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Isard, M. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), Savannah, GA, USENIX Association, pp. 265–283.
24. National Institute of Standards and Technology (NIST). (2007). Guide to Intrusion Detection and Prevention Systems (IDPS), NIST Special Publication 800-94. Gaithersburg, MD: NIST.
25. Chen, S., Zhang, Y., & Xu, X. (2023). Detection of Algorithmically Generated Malicious Domain Names Using Feature Fusion of Meaningful Word-Segmentation and n-Gram Sequence Features. Applied Sciences, 13(7), 4406. https://doi.org/10.3390/app13074406
26. 22. Lee, J. Y., Ha, N., & Kim, K. (2019). DGA-based malware detection using DNS traffic analysis. In Proceedings of the 2019 ACM Workshop on Artificial Intelligence and Security (AISec ’19), London, UK: ACM, pp. X–Y. https://doi.org/10.1145/3338840.3355672
27. Alqahtani, H., & Alohali, O. A. (2024). Advances in artificial intelligence for detecting domain generation algorithms: A review. Information & Computer Security, 32(2), 123-150. https://doi.org/10.1016/ j.infcos.2024.00123
28. Wilk-Jakubowski, J. L., Pawlik, L., Wilk-Jakubowski, G., & Sikora, A. (2025). Machine Learning and Neural Networks for Phishing Detection: A Systematic Review (2017–2024). Electronics, 14(18), 3744. https://doi.org/10.3390/electronics14183744
29. Safi, A., & Boubiche, D. (2023). A systematic literature review on phishing website detection techniques. Journal of Information Security and Applications, 74, 103149. https://doi.org/10.1016/j.jisa.2023.103149

30. Dinesh, P. M., & Kumar, V. (2023). Identification of phishing attacks using machine learning: A comparative study. In Proceedings of the 2023 International Conference on Networking and Intelligent Systems (ICoNIS ’23), Bangalore, India: E3S Web of Conferences, 400(1):04010. https://doi.org/10.1051/e3sconf/20234004010
31. Aslam, S., Aslam, H., Manzoor, A., Hui, C., & Rasool, A. (2024). AntiPhishStack: LSTM-based Stacked Generalization Model for Optimized Phishing URL Detection. arXiv preprint arXiv:2401.08947
32. Yerima, S. Y., & Alzaylaee, M. K. (2020). High Accuracy Phishing Detection Based on Convolutional Neural Networks. arXiv preprint arXiv:2004.03960.
33. Maneriker, P., Stokes, J. W., Lazo, E. G., Carutasu, D., Tajaddodianfar, F., & Gururajan, A. (2021). URLTran: Improving Phishing URL Detection Using Transformers. arXiv preprint arXiv:2106.05256.
34. An, P., Shafi, R., Mughogho, T., & Onyango, A. O. (2025). Multilingual Email Phishing Attacks Detection using OSINT and Machine Learning. arXiv preprint arXiv:2501.08723.
35. Stewart, L. (2025). Detecting Domain Generation Algorithms in Malicious DNS Traffic Using Machine Learning Approaches. SSRN Working Paper. https://doi.org/10.2139/ssrn.5231282
36. Alzboon, M. S. (2025). Phishing Website Detection Using Machine Learning. International Journal of Scientific and Management Research, 7(4), 27-63. http://doi.org/10.37502/IJSMR.2024.7403
37. Desai, P., & Bhatt, M. (2024). Phishing Website Detection Using Machine Learning: Feature Engineering and Model Evaluation. International Journal of Computer Applications, 182(10), 25-34.
38. Xu, Z., Wang, D., & Wu, X. (2024). DGA Domain Name Detection and Classification Using Deep Learning Architectures. International Journal of Advanced Computer Science and Applications (IJACSA), 15(7), 307-318