Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Методи аналізу Big Data та їх застосування до розробки інструменту для прогнозування порушень умов використання доменних імен
(тема)

Виконав:
Здобувач другого року навчання
групи ІПЗм-23-4

Андрій УШАКОВ
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник проф. Анатолій РУТКАС
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Ушакову Андрію Михайловичу
(прізвище, ім’я, по батькові)

1. Тема роботи «Методи аналізу Big Data та їх застосування до розробки інструменту для прогнозування порушень умов використання доменних імен»
Затверджена наказом по університету від 15.04.2025 р. № 290 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 23.06.2025
3. Вихідні дані до роботи науково-технічні публікації, дані Інтернет-джерел та друкованих видань щодо методів аналізу великих даних і машинного навчання, дослідження та розробки систем моніторингу доменних імен, статистичні звіти про кіберзагрози, технічна документація для обраних платформ і бібліотек
4. Перелік питань, що потрібно опрацювати в роботі
мета роботи, аналіз предметної галузі і постановка задачі, огляд та аналіз літературних джерел з дослідження, теоретичне дослідження, практичне дослідження

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	16.04.2025	*виконано*
2	Аналіз предметної галузі	17.04.2025 – 22.04.2025	*виконано*
3	Огляд та аналіз літературних, наукових джерел	23.04.2025 – 26.04.2025	*виконано*
4	Постановка задачі	27.04.2025	*виконано*
5	Теоретичне дослідження	18.04.2025 – 01.05.2025	*виконано*
6	Підготовка до апробації результатів дослідження. Публікація матеріалів	02.05.2025 – 10.05.2025	*виконано*
7	Практичне дослідження	11.05.2025 – 25.05.2025	*виконано*
8	Підготовка пояснювальної записки	26.05.2025 – 06.06.2025	*виконано*
9	Підготовка презентації та доповіді	07.06.2025 – 15.06.2025	*виконано*
10	Перевірка на плагіат	16.06.2025	*виконано*
11	Нормоконтроль	17.06.2025	*виконано*
12	Рецензування	18.06.2025	*виконано*
13	Попередній захист	18.06.2025	*виконано*
14	Занесення диплома в електронний архів	19.06.2025	*виконано*
15	Допуск до захисту у зав. кафедри	20.06.2025	*виконано*

Дата видачі завдання 16 квітня 2025 р.

Студент _____ Андрій УШАКОВ
(підпис)

Керівник роботи _____ проф. Анатолій РУТКАС
(підпис) (посада, Власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 96 с., 14 рис., 7 табл., 19 джерел.

АНАЛІЗ ДАНИХ, ДОМЕННІ ІМЕНА, КІБЕРБЕЗПЕКА, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ ЗАГРОЗ, ФІШИНГ, APACHE SPARK, BIG DATA, PYTHON.

Об’єктом дослідження є процес аналізу та моніторингу доменних імен для виявлення порушень умов їх використання.

Метою роботи є розробка прототипу інструменту, що забезпечує прогнозування загроз у використанні доменних імен за допомогою аналізу великих даних і застосування алгоритмів машинного навчання.

Методами розробки та проектування є аналіз проблемної області дослідження, використання архітектурного проєктування програмного забезпечення, проєктування баз даних, алгоритмів класифікації та виявлення аномалій, а також інтеграції різнорідних даних із застосуванням технологій Big Data.

У результаті кваліфікаційної роботи було реалізовано повнофункціональний прототип програмного інструменту, що здійснює автоматизований аналіз доменних імен, виявлення потенційних загроз та визначення ступеня ризику на основі реальних WHOIS та DNS даних. Розроблена система включає повноцінну архітектуру, інтерфейс користувача, модулі обробки даних, а також натреновану модель машинного навчання для класифікації доменів. Інтеграція із веб-інтерфейсом дозволила забезпечити зручний механізм взаємодії користувача із системою.

Отримані результати свідчать про дієвість запропонованого підходу та потенціал для подальшого розширення системи в умовах реального часу та великого навантаження.

DATA ANALYSIS, DOMAIN NAMES, CYBERSECURITY, MACHINE LEARNING, THREAT PREDICTION, PHISHING, APACHE SPARK, BIG DATA, PYTHON.

The object of research is the process of analyzing and monitoring domain names to detect violations of their usage conditions.

The purpose of the work is to develop a prototype of a tool that ensures threat prediction in the use of domain names through the analysis of Big Data and the use of Machine Learning algorithms.

The development and design methods are the analysis of the problem area of the study, software architecture design, database, classification and anomaly detection algorithms design, and the integration of heterogeneous data using Big Data technologies.

As a result of the qualification work, a fully functional prototype of a software tool, that performs automated analysis of domain names, identifies potential threats, and determines the degree of risk based on real WHOIS and DNS data, was implemented. The developed system includes full-fledged architecture, user interface, data processing modules, and a trained machine learning model for domain classification. Integration with the web interface allowed us to provide a convenient mechanism for user interaction with the system.

The results obtained indicate the effectiveness of the proposed approach and the potential for further expansion of the system in real-time and high-load conditions.

Завідувачу кафедри
ПІ
(скорочена назва кафедри)
проф. Кирилу СМЕЛЯКОВУ
(вчене звання, власне ім'я, прізвище)

ЗАЯВА
щодо самостійності виконання кваліфікаційної роботи та можливості її публікації (та/або публікації анотації кваліфікаційної роботи) в електронному архіві відкритого доступу ElAr KhNURE

Я, Ушаков Андрій Михайлович, студент гр. ІПЗм-23-4, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Методи аналізу Big Data та їх застосування до розробки інструменту для прогнозування порушень умов використання доменних імен», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата 19.06.2025 Підпис 

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення

ACID – Atomicity, Consistency, Isolation, Durability

API – Application Programming Interface

AWS – Amazon Web Services

C&C – Command and Control

DGA – Domain Generation Algorithm

DKIM – DomainKeys Identified Mail

DMARC – Domain-based Message Authentication, Reporting & Conformance

DNS – Domain Name System

DNSBL – Domain Name System Blacklist

DoH – DNS-over-HTTPS

HDFS – Hadoop Distributed Filesystem

HTTPS – HyperText Transfer Protocol Secure

IP – Internet Protocol

MFA – multi-factor authentication

ML – Machine Learning

MX – mail exchanger

PCAP – Packet Capture

RDAP – Registration Data Access Protocol

SIEM – Security Information and Event Management

SPA – Single Page Application

SPF – Sender Policy Framework

SQL – Structured Query Language

TTL – Time To Live

UI – User Interface

UML – Unified Modeling Language

USD – United States dollar

UX – User Experience

ВСТУП

Сучасний розвиток інформаційних технологій створює нові можливості для аналізу великих обсягів даних (Big Data) та їх використання у різних сферах, включаючи кібербезпеку та адміністрування інтернет-ресурсів. Однією з важливих проблем є прогнозування порушень умов використання доменних імен, що пов’язано зі зростанням кількості кіберзагроз, таких як шахрайство, фішинг або незаконна діяльність через мережеві ресурси. Наразі існує потреба в розробці ефективних методів виявлення аномалій і ризиків у використанні доменних імен, які базуються на аналізі великих даних. Актуальність цієї теми обумовлена необхідністю забезпечення надійного функціонування інтернету, захисту користувачів та зменшення економічних втрат, спричинених порушеннями умов використання доменів.

Метою цієї роботи є аналіз методів Big Data, які можуть бути застосовані для прогнозування порушень умов використання доменних імен, та розробка теоретичних та практичних підходів до створення відповідного інструменту. У рамках дослідження ставляться такі задачі як вивчення сучасних методів збору, обробки та аналізу великих даних, оцінка їхньої ефективності у вирішенні завдань прогнозування, визначення можливості застосування машинного навчання, статистичних методів і алгоритмів виявлення аномалій у даних про доменні імена, формулювання теоретичних рекомендацій для розробки інструменту прогнозування та практична реалізація функціонального прототипу такої системи.

Об’єктом дослідження є процеси, що призводять до порушення умов використання доменних імен, зокрема використання доменів для нелегальної діяльності. Предметом дослідження виступають методи аналізу великих даних та алгоритми прогнозування, які дозволяють виявляти потенційні загрози і порушення у використанні доменних імен.

У ході роботи використано такі методи: аналіз літературних джерел для вивчення існуючих підходів, обробка великих даних для моделювання поведінки доменів, застосування методів машинного навчання для побудови прогнозних моделей, критичний аналіз результатів, отриманих за допомогою обраних підходів.

Використання цих методів дозволило реалізувати комплексну систему аналізу доменних імен із підтримкою витягу WHOIS та DNS даних, генерації структурних ознак та класифікації доменів за ступенем ризику з використанням моделей машинного навчання.

Результатом дослідження стало не лише формулювання теоретичних основ використання методів Big Data для виявлення порушень у використанні доменних імен, а й створення робочого прототипу програмного інструменту з базовим веб-інтерфейсом, що дозволяє у напівавтоматичному режимі аналізувати доменні імена та виявляти потенційно шкідливі. Отримані результати демонструють ефективність запропонованого підходу для оперативного виявлення кіберзагроз, що дозволяє розглядати систему як основу для подальшого розвитку у реальних умовах.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Огляд предметної галузі

Порушення умов використання доменних імен є актуальною проблемою, яка набуває все більшого значення у сучасному цифровому світі. Домени, як ключові елементи інтернет-інфраструктури, забезпечують доступ до веб-ресурсів, проте часто стають об'єктами зловживань у кіберпросторі. Основні типи порушень включають фішинг, спам, створення шкідливих вебсайтів, розповсюдження зловмисного програмного забезпечення (ПЗ) та шахрайство. Наприклад, фішингові домени створюються для обману користувачів з метою отримання їхніх конфіденційних даних, таких як паролі або фінансова інформація. Спам-домени використовуються для масового надсилання небажаних електронних листів, часто з метою поширення шкідливого ПЗ або або рекламної інформації сумнівного змісту.

Масштаби цієї проблеми значні. Фішингові атаки є однією з найбільш поширених форм кіберзагроз, кількість яких щороку зростає. Кібератаки останніх років показали, наскільки вразливою є глобальна цифрова інфраструктура, що підкреслює необхідність вдосконалення методів їх виявлення [1]. У березні 2024 року обсяг втрат від фішингових атак досяг $71 млн, що на 50% більше порівняно з лютим, а жертвами шахраїв стали понад 77500 користувачів [2].

Такі атаки завдають шкоди репутації компаній, які стають жертвами зловмисників. Крім того, проблема ускладнюється тим, що зловмисники активно адаптуються до нових інструментів моніторингу, швидко створюючи нові домени після блокування попередніх.

Особливістю аналізу порушень умов використання доменів є те, що дані для такого аналізу можуть бути надзвичайно великими та різноманітними. Серед них записи DNS-запитів, інформація з баз даних WHOIS, логи веб-трафіку, дані про IP-адреси, метрики поведінки користувачів на сайтах тощо. Традиційні методи аналізу часто виявляються недостатньо ефективними через обсяги та динаміку таких даних. Сучасні підходи до аналізу великих даних дозволяють автоматизувати процес виявлення загроз та значно підвищують точність прогнозування [3].

Станом на сьогодні багато організацій і компаній використовують різноманітні інструменти для боротьби з порушеннями. Однак ці інструменти, як правило, мають низку обмежень, таких як повільне виявлення загроз, низька точність аналізу або висока вартість впровадження. Крім того, вони нерідко орієнтовані на виявлення відомих типів загроз і неефективні проти нових, ще не ідентифікованих моделей поведінки доменів.

Таким чином, проблема порушення умов використання доменних імен вимагає створення нових підходів, які враховуватимуть масштаби сучасних даних і забезпечуватимуть виявлення загроз у реальному часі. Це створює передумови для використання технологій Big Data, машинного навчання та аналізу поведінкових даних, які можуть стати ефективними інструментами для вирішення цієї проблеми.

1.2 Огляд сучасних підходів до аналізу доменних імен

Захист доменних імен та моніторинг їх використання є важливим завданням для підтримки безпеки у кіберпросторі. Існуючі підходи до аналізу доменів зосереджуються на кількох ключових аспектах: виявленні аномалій, аналізі поведінкових патернів, використанні чорних списків та машинного навчання.

Одним із найпоширеніших методів є використання чорних списків (blacklists), таких як DNSBL (Domain Name System Blacklist) або Spamhaus. Ці системи працюють на основі попередньо зібраної інформації про підозрілі домени та IP-адреси, які використовувалися у шкідливих цілях. Наприклад, система Spamhaus регулярно оновлює базу даних, що дозволяє ідентифікувати спам-домени у реальному часі. Однак, недоліком такого підходу є затримка між виявленням нової загрози та її додаванням до бази.

Другим підходом є аналіз DNS-запитів та поведінкових метрик. Наприклад, аномалії в частоті DNS-запитів до певного домену можуть вказувати на його потенційно шкідливу активність. Сучасні системи, такі як Passive DNS Monitoring, дозволяють виявляти зв'язки між доменами та IP-адресами, які можуть

використовуватися для побудови мереж ботнетів [4]. Крім того, поведінковий аналіз може включати відстеження патернів доступу користувачів до доменів.

Значного поширення набувають підходи, засновані на алгоритмах машинного навчання. Вони дозволяють автоматизувати процес виявлення порушень та адаптуватися до нових, ще не ідентифікованих загроз. Наприклад, класифікаційні моделі можуть використовуватися для розпізнавання зловмисних доменів на основі таких характеристик, як довжина, частота використання символів, або часові характеристики реєстрації [5]. Недоліком є висока обчислювальна складність і потреба у великих обсягах навчальних даних.

Окремої уваги заслуговує інтеграція декількох підходів у комплексні системи аналізу, такі як Threat Intelligence Platforms. Ці платформи поєднують виявлення аномалій, аналіз даних WHOIS, моніторинг DNS та машинне навчання. Вони надають користувачам не лише інструменти для виявлення загроз, а й механізми для їх нейтралізації.

Таким чином, сучасні підходи до аналізу доменних імен постійно вдосконалюються, адаптуючись до нових загроз. Проте кожен із них має свої обмеження, що підкреслює необхідність розробки нових, більш універсальних рішень.

1.3 Роль Big Data в аналізі доменних імен

Сучасний аналіз доменних імен неможливо уявити без використання технологій Big Data. Масштаби даних, які генеруються в цій сфері, зростають з кожним роком. Ці дані включають мільйони DNS-запитів щодня, записи баз даних WHOIS, дані веб-трафіку, IP-адреси, метрики поведінки користувачів та іншу інформацію. Використання технологій Big Data дозволяє обробляти ці дані в реальному часі, виявляти аномалії та прогнозувати потенційні загрози.

Одним із ключових аспектів застосування Big Data є масштабованість. Сучасні системи аналізу, такі як Apache Hadoop та Apache Spark, дозволяють працювати з величезними обсягами даних, забезпечуючи їх швидке оброблення та зберігання. Наприклад, аналіз DNS-запитів у глобальному масштабі потребує

інфраструктури, яка здатна опрацьовувати десятки терабайт даних щодня. Завдяки Big Data це стало можливим.

Іншим важливим аспектом є інтеграція різних типів даних. Для аналізу доменних імен можуть використовуватися дані з різних джерел, таких як WHOIS, DNS, веб-трафік або реєстраційні логи. Технології Big Data забезпечують можливість інтегрувати ці дані в єдину платформу, що дозволяє отримати більш повне уявлення про потенційно шкідливі домени. Наприклад, системи BigQuery від Google часто використовуються для аналізу таких великих і різнорідних масивів даних.

Big Data також відіграє важливу роль у побудові моделей прогнозування. Наприклад, алгоритми машинного навчання можуть використовувати великі набори історичних даних для ідентифікації патернів, характерних для шкідливих доменів. Такі моделі здатні прогнозувати ризики, пов'язані з новими доменами, або ідентифікувати домени, що вже використовуються у ботнет-мережах. Дослідження показують, що використання машинного навчання у поєднанні з Big Data підвищує точність прогнозів до 85–90% [6].

Підхід використання Big Data для оцінки ризиків за допомогою кластеризації та аналізу великих обсягів інформації можна застосувати для аналізу ризиків у використанні доменних імен [7]. Підхід до кластеризації при оцінці ризиків можна адаптувати для групування доменів за схожими характеристиками, що дозволить виявляти потенційно шкідливі домени, а обробка великих обсягів даних може застосовуватися для роботи з великими наборами DNS-логів.

Однак застосування Big Data в аналізі доменних імен має і свої виклики. Це, зокрема, висока обчислювальна складність, потреба у значних ресурсах для зберігання та обробки даних, а також забезпечення конфіденційності та безпеки інформації. Крім того, ефективне використання цих технологій потребує кваліфікованих фахівців і спеціалізованих інструментів.

Таким чином, Big Data стає ключовим елементом у сучасному аналізі доменних імен, відкриваючи нові можливості для виявлення загроз і забезпечення

високого рівня кібербезпеки. Використання цих технологій дозволяє значно підвищити ефективність моніторингу та попередження порушень у цій сфері.

1.4 Тенденції та виклики у сфері безпеки доменів

Сфера безпеки доменних імен постійно змінюється під впливом розвитку технологій та еволюції кіберзагроз. Однією з ключових тенденцій останніх років є зростання кількості доменів, які використовуються для фішингу, спаму та інших зловмисних цілей. За даними звіту Global Phishing Survey 2024, кількість фішингових доменів зросла на 35% за останній рік, а частка нових доменів, зареєстрованих для шахрайських дій, становить близько 12% від загальної кількості реєстрацій [8].

Ще однією тенденцією є автоматизація шкідливих дій. Зловмисники дедалі частіше використовують ботнети для автоматичної реєстрації доменів та їх швидкої зміни у відповідь на блокування. Це явище, відоме як «fast flux», ускладнює моніторинг та виявлення шкідливих доменів. Крім того, розвиток технологій DNS-over-HTTPS (DoH) ставить нові виклики перед фахівцями з кібербезпеки. Хоча DoH покращує конфіденційність користувачів, він також ускладнює виявлення аномальної активності в мережевому трафіку [9].

У сфері кібербезпеки важливим питанням є використання доменів для управління ботнетами. Так звані «command-and-control» (C&C) домени дозволяють зловмисникам керувати великою кількістю заражених пристроїв, залишаючись при цьому невидимими для традиційних систем моніторингу. Згідно з дослідженням SANS Institute, більше 40% ботнетів використовують домени динамічної реєстрації, що дозволяє їм швидко змінювати інфраструктуру та уникати виявлення [10].

Етичні виклики також стають дедалі більш актуальними. Наприклад, збір і аналіз великих обсягів даних про домени може порушувати конфіденційність користувачів, особливо якщо ці дані включають інформацію про реєстрантів або користувацький трафік. Це вимагає збалансованого підходу до підвищення рівня безпеки та захисту приватності.

Серед викликів, з якими стикається галузь, є також обмеженість ресурсів та компетенцій. Впровадження сучасних інструментів аналізу доменів, таких як технології машинного навчання, вимагає значних фінансових витрат і залучення висококваліфікованих спеціалістів.

Таким чином, сфера безпеки доменних імен стикається як із новими можливостями, так і з викликами. Ефективне вирішення цих проблем потребує інтеграції сучасних технологій, міждисциплінарного підходу та врахування етичних аспектів.

1.5 Оцінка попередніх рішень

Протягом останнього десятиліття було розроблено чимало рішень для моніторингу та аналізу доменних імен, спрямованих на підтримку безпеки у кіберпросторі. Найпоширенішими є використання чорних списків, систем моніторингу DNS та поведінкових аналізаторів. Однак, попри їхню ефективність у певних умовах, ці підходи мають значні обмеження, що відкривають простір для вдосконалення.

Наприклад, чорні списки, такі як DNSBL, є широко впроваджуваними у багатьох системах кібербезпеки. Вони дозволяють оперативно блокувати відомі шкідливі домени, однак не здатні ефективно реагувати на нові, раніше невідомі загрози. Крім того, ці списки залежать від регулярного оновлення, що створює затримки у виявленні загроз. Подібні системи також не враховують складні патерни поведінки зловмисників, такі як «fast flux», коли домени швидко змінюють IP-адреси.

Сучасні системи аналізу DNS та поведінкові аналізатори стали інноваційним кроком у напрямку розширення можливостей виявлення загроз. Вони використовуються у хмарних сервісах кібербезпеки, таких як Cisco Umbrella, і дозволяють виявляти потенційно небезпечні домени на основі аномальної активності у запитах. Однак такі системи часто потребують значних обчислювальних ресурсів, а також не завжди ефективні для виявлення нових методів атак, що еволюціонують.

Інтеграція машинного навчання у моніторинг доменів стала черговим важливим кроком у розвитку галузі. Наприклад, рішення на основі класифікаційних моделей, таких як Random Forest або градієнтний бустинг, продемонстрували високий рівень точності у виявленні шкідливих доменів. Проте їх впровадження вимагає великих обсягів навчальних даних, а також значної оптимізації алгоритмів для зменшення хибнопозитивних результатів [5].

Практична цінність цього дослідження полягає в подоланні низки ключових проблем, які залишаються актуальними в межах сучасних підходів. Зокрема, впровадження інструменту прогнозування, здатного працювати з великими обсягами даних у реальному часі, знизить час реакції на загрози. Інноваційність підходу полягає у поєднанні технологій Big Data, поведінкового аналізу та моделей машинного навчання для побудови універсального інструменту, здатного адаптуватися до нових кіберзагроз. Це дозволить не лише підвищити точність моніторингу, але й знизити ризики помилкового блокування доменів.

Таким чином, аналіз попередніх рішень свідчить про необхідність розробки більш ефективних, масштабованих та адаптивних інструментів для аналізу доменних імен. Вирішення цих задач сприятиме зміцненню кібербезпеки та захисту глобальної інтернет-інфраструктури.

2 ОГЛЯД ТА АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

Літературний огляд є важливим етапом наукового дослідження, оскільки він дозволяє систематизувати існуючі знання, проаналізувати ключові підходи та визначити прогалини у вивченні обраної теми. Для обраної теми дослідження літературний огляд є основою для розуміння сучасних тенденцій, методологій та обмежень у сфері кібербезпеки.

Для виконання огляду та аналізу було відібрано джерела, що відповідають кільком критеріям. По-перше, актуальність: джерела не старші за 10 років, що гарантує врахування сучасних викликів у галузі. По-друге, авторитетність: джерела включають наукові статті з рецензованих журналів, монографії провідних авторів і звіти міжнародних організацій, таких як ISO або APWG. По-третє, об'єктивність і достовірність: не розглядаються матеріали із сумнівних або неперевірених джерел, перевага надається публікаціям із чітко визначеними методами та результатами досліджень.

Огляд джерел структурований за трьома основними напрямами, що відповідають ключовим аспектам теми дослідження. Перший напрям охоплює загальні аспекти аналізу доменних імен та сучасні виклики у цій сфері. Другий присвячений ролі технологій Big Data у вирішенні проблем моніторингу та аналізу доменних імен. Третій напрям досліджує використання методів машинного навчання для прогнозування порушень у кіберпросторі. Такий підхід дозволяє забезпечити всебічний аналіз існуючих підходів, виділити ключові тенденції та визначити сфери, які потребують подальшого розвитку.

Цей огляд допоможе побудувати основу для розробки ефективного інструменту прогнозування, який поєднає інноваційні підходи аналізу Big Data з методами машинного навчання.

2.1 Огляд основних джерел

Для виконання дослідження було відібрано низку авторитетних джерел, що висвітлюють ключові аспекти аналізу доменних імен, застосування технологій

Big Data та методів машинного навчання у сфері кібербезпеки. Джерела згруповані за тематичними напрямами, які відповідають структурі дослідження.

2.1.1 Загальні аспекти аналізу доменних імен та кібербезпеки

Доменні імена є важливим елементом інтернет-інфраструктури, однак вони часто використовуються зловмисниками для шахрайських дій. Аналіз доменів є ключовою складовою кібербезпеки, яка дозволяє виявляти шкідливі активності, зокрема фішинг, спам, ботнети та інші кіберзагрози.

Ключовим джерелом у цій категорії є робота Kim T. H. та Reeves D. "A survey of domain name system vulnerabilities and attacks", яка пропонує глибокий аналіз фундаментальних аспектів безпеки доменів та інфраструктури DNS [11]. Автори акцентують увагу на важливості поєднання різних методів аналізу для виявлення аномалій у доменах.

Звіти Anti-Phishing Working Group (APWG), зокрема "Global Phishing Survey 2024", містять актуальну статистику та аналіз загроз, пов'язаних із використанням доменних імен для фішингу та інших шахрайських дій [8].

Ці джерела забезпечують розуміння поточного стану безпеки доменних імен, виявляючи ключові виклики, такі як швидкість адаптації зловмисників, масштабність атак та недостатня інтеграція сучасних технологій у боротьбі зі шкідливими доменами.

2.1.2 Методи аналізу великих даних (Big Data)

Технології Big Data стали невід’ємною частиною сучасних підходів до обробки великих обсягів інформації. У контексті аналізу доменних імен ці технології забезпечують швидке збирання, зберігання та обробку даних, необхідних для виявлення загроз та прогнозування порушень.

Одним із ключових джерел, що охоплює теоретичні та практичні аспекти Big Data, є книга White T. "Hadoop: The Definitive Guide" [12]. У ній детально описано принципи роботи з розподіленими обчисленнями, які є основою для аналізу великих обсягів даних, а також розглядаються можливості використання

технологій Big Data для обробки великих обсягів даних у реальному часі. Hadoop дозволяє масштабувати інфраструктуру обробки даних і використовувати її для аналізу DNS-запитів та поведінкових метрик у реальному часі.

Ще одним важливим джерелом є книга "Big Data: Principles and Paradigms" під редакцією Buyya R., яка систематично описує архітектуру, методи обробки та алгоритми аналізу великих даних [13]. Цей підхід є критично важливим для інтеграції різнорідних даних, таких як записи DNS, метрики поведінки користувачів та бази WHOIS.

Практичний аспект аналізу великих даних розглядається у книзі Vandana P. Janeja "Data Analytics for Cybersecurity" [3]. Автор фокусується на застосуванні таких інструментів, як Apache Spark, для розпізнавання патернів, характерних для шкідливих доменів. Також наведено реальні приклади впровадження технологій Big Data у кібербезпеку, що підкреслює їхню практичну значущість.

Методи машинного навчання у поєднанні з Big Data розглядаються у книзі "Data Mining: Practical Machine Learning Tools and Techniques" під редакцією Witten I.H., яка пропонує підходи до створення моделей для класифікації великих наборів даних [14]. Особливу увагу приділено алгоритмам кластеризації та виявлення аномалій, які є ефективними у контексті аналізу доменних імен.

Таким чином, джерела, що охоплюють методи аналізу великих даних, забезпечують розуміння фундаментальних принципів та інструментів, необхідних для дослідження у сфері моніторингу доменів. Вони надають як теоретичні основи, так і практичні рекомендації для ефективного використання технологій Big Data у кібербезпеці.

2.1.3 Використання машинного навчання у прогнозуванні загроз

Машинне навчання (ML) стало одним із найпотужніших інструментів для прогнозування загроз, пов’язаних із доменними іменами. Воно дозволяє створювати моделі, які автоматично розпізнають шкідливі домени на основі аналізу великих обсягів даних.

Робота Thapliyal V. та Thapliyal P. "Machine learning for cybersecurity: threat detection, prevention, and response" детально розглядає використання алгоритмів класифікації, таких як Random Forest, градієнтний бустинг та нейронні мережі для виявлення шкідливих доменів [5]. Автори акцентують увагу на можливості зниження хибнопозитивних результатів завдяки використанню ансамблевих методів.

Практичний підхід до аналізу доменів описано у роботі A. Garg "An evaluation of machine learning methods for domain name classification" [6]. У ній наведено реальні приклади використання методів кластеризації та аномалій для аналізу структури та патернів доменних імен. Ці методи дозволяють виявляти нові загрози, що не входять до чорних списків.

Кожна з тематичних груп джерел надає важливу інформацію для побудови цілісного уявлення про стан досліджуваної проблеми, що є необхідним для обґрунтування нових підходів у цій галузі.

2.2 Аналіз літератури

Аналіз обраних літературних джерел дозволяє виявити ключові теорії, концепції та моделі, що є основою дослідження в галузі кібербезпеки та аналізу доменних імен.

2.2.1 Основні теорії та концепції

Стаття Kim T. H. та Reeves D. "A survey of domain name system vulnerabilities and attacks" представляє концепцію багаторівневого аналізу, яка поєднує моніторинг DNS, аналіз баз WHOIS та поведінкові дані. Автори підкреслюють важливість системного підходу до виявлення аномалій у доменних іменах [11].

Додатково, у роботі Buyya R., Calheiros R. N. та Dastjerdi A. V. "Big Data: Principles and Paradigms" розглядається принцип розподілених обчислень, що є основою для обробки великих обсягів даних у реальному часі. Це створює фундамент для інтеграції даних з різних джерел у контексті аналізу доменних імен [13].

2.2.2 Моделі та методи аналізу

Методи класифікації та виявлення аномалій, описані у книзі Witten I.H., є основою для багатьох сучасних моделей прогнозування загроз. Зокрема, Random Forest демонструє високу точність у класифікації доменів як шкідливих або безпечних [14].

A. Garg у своїй роботі пропонує методи аналізу патернів у структурі доменних імен, використовуючи алгоритми кластеризації. Цей підхід дозволяє виявляти потенційно шкідливі домени, які ще не були зареєстровані у чорних списках [6].

2.2.3 Ефективність існуючих підходів

Попри досягнення, сучасні підходи до аналізу доменних імен мають свої обмеження. Чорні списки, такі як DNSBL, демонструють високу ефективність у виявленні відомих шкідливих доменів, однак вони не здатні реагувати на нові загрози. Це робить їх менш ефективними у випадках швидкої зміни інфраструктури зловмисників, наприклад, при використанні техніки «fast flux» [11].

Моделі машинного навчання дозволяють значно підвищити точність аналізу, однак їх використання залежить від доступу до великих обсягів якісних даних. Крім того, існує ризик хибнопозитивних результатів, які можуть призводити до блокування легітимних доменів [5].

Інтеграція Big Data відкриває нові можливості для аналізу великих обсягів даних у реальному часі. Проте, ці технології потребують значних ресурсів для впровадження та підтримки. Використання розподілених обчислень є критично важливим, але залишається технічно складним для багатьох організацій [13].

Таким чином, існуючі підходи демонструють значний прогрес у моніторингу доменів, проте залишаються проблеми, які потребують подальшого вирішення. Загалом, аналіз літератури вказує на важливість поєднання різних підходів та інструментів для підвищення ефективності моніторингу доменних імен. Інтеграція технологій Big Data, алгоритмів машинного навчання та поведінкових моделей є перспективним напрямком для подальших досліджень.

2.3 Оцінка актуальності та новизни

Аналіз літератури показує, що проблема прогнозування порушень умов використання доменних імен є однією з найактуальніших у сфері кібербезпеки. Зростання кількості кіберзагроз, таких як фішинг, спам, ботнети та інші види зловживань, підкреслює необхідність удосконалення сучасних підходів до моніторингу та аналізу.

Наукова новизна представлених джерел виявляється у розробці та впровадженні інноваційних підходів до аналізу доменів. Наприклад, Thapliyal V. та Thapliyal P. пропонують сучасні алгоритми машинного навчання, які дозволяють підвищити точність прогнозування загроз до 90% [5]. Окрім цього, технології Big Data, описані Buyya R., забезпечують можливість обробки великих обсягів даних у реальному часі, що є ключовим для реагування на нові кіберзагрози [13].

Оцінка актуальності джерел також свідчить про їхню відповідність сучасним викликам у сфері кібербезпеки. Наприклад, багато досліджень фокусуються на інтеграції різних підходів, таких як поведінковий аналіз, алгоритми класифікації та використання розподілених обчислень. Це дозволяє адаптувати наявні рішення до сучасного середовища кіберзагроз, що швидко змінюється.

Попри значний прогрес, дослідження також виявляють існуючі прогалини. Наприклад, обмеження чорних списків у боротьбі з новими загрозами залишають простір для розвитку більш адаптивних та інтегрованих інструментів. Крім того, недостатнє використання поведінкових моделей для аналізу доменів створює можливості для подальших досліджень.

Таким чином, представлені у літературі підходи є не лише актуальними, але й науково інноваційними. Вони закладають основу для розвитку ефективних рішень у сфері моніторингу доменів та прогнозування кіберзагроз, водночас створюючи передумови для подальших досліджень.

2.4 Висновки з огляду

Огляд літератури дозволив систематизувати сучасні підходи та інструменти, які використовуються для аналізу доменних імен і прогнозування кіберзагроз.

Основні теорії та концепції, розглянуті у представлених роботах, підкреслюють важливість інтеграції різних підходів, зокрема моніторингу DNS, аналізу поведінкових патернів та використання технологій Big Data.

Аналіз моделей і методів, таких як класифікаційні алгоритми машинного навчання (Random Forest, градієнтний бустинг) та методи виявлення аномалій, показав їхню високу ефективність у прогнозуванні загроз, проте існують виклики, пов’язані з потребою у великих обсягах навчальних даних та обчислювальних ресурсах, що потребує подальшого вдосконалення інструментів і алгоритмів.

Оцінка актуальності літератури свідчить про те, що проблема аналізу доменних імен залишається важливою у сучасному цифровому світі. Розвиток методів машинного навчання та Big Data відкриває нові можливості для моніторингу та прогнозування загроз. Водночас існуючі прогалини, такі як недостатня інтеграція поведінкових моделей або обмеження чорних списків, створюють передумови для нових досліджень.

На основі проведеного огляду можна зробити висновок, що ефективне прогнозування порушень умов використання доменних імен потребує розробки адаптивного інструменту, здатного інтегрувати різнорідні дані та використовувати сучасні технології обробки великих обсягів інформації. Подальші дослідження повинні бути спрямовані на створення універсальних рішень, які дозволять оперативно реагувати на нові кіберзагрози.

3 ПОСТАНОВКА ЗАДАЧІ

Актуальність аналізу та прогнозування порушень умов використання доменних імен зумовлена зростанням кількості кіберзагроз у сучасному цифровому світі. Шкідливі доменні імена використовуються для фішингу, розповсюдження шкідливого програмного забезпечення та управління ботнетами. Це створює значні ризики як для користувачів, так і для організацій, які зазнають фінансових та репутаційних втрат.

Кваліфікаційна робота має на меті розробку ефективного інструменту прогнозування, який дозволить виявляти порушення умов використання доменних імен на основі аналізу великих обсягів даних. Цей інструмент має забезпечувати виявлення загроз у реальному часі, бути масштабованим і адаптивним до нових типів атак.

Для досягнення цієї мети необхідно інтегрувати сучасні технології Big Data та методи машинного навчання. Це дозволить обробляти різнорідні дані, такі як DNS-запити, записи WHOIS та поведінкові метрики, створюючи багатовимірну модель для аналізу потенційних загроз. Розробка такого інструменту сприятиме підвищенню ефективності моніторингу доменів, зменшенню ризиків кіберзагроз і зміцненню загальної кібербезпеки.

3.1 Формулювання задачі

Метою дослідження є розробка ефективного інструменту для прогнозування порушень умов використання доменних імен, який буде базуватися на аналізі великих обсягів даних та використанні сучасних алгоритмів машинного навчання. Для досягнення цієї мети необхідно вирішити низку конкретних підзадач:

- вибір методів аналізу даних: ідентифікувати оптимальні методи для обробки великих обсягів даних, включаючи алгоритми класифікації, виявлення аномалій та поведінковий аналіз;
- розробка концептуальної моделі інструменту: побудувати модель, яка інтегрує дані з різних джерел (DNS-запити, бази WHOIS, поведінкові метрики) та дозволяє проводити аналіз у реальному часі;

– аналіз існуючих рішень: провести порівняння сучасних підходів до моніторингу доменів, визначити їхні сильні сторони та обмеження;
– вибір програмних рішень: визначити платформи, інструменти та алгоритми, які будуть використані для реалізації інструменту;
– розробка прототипу: реалізувати початкову версію інструменту для тестування його ефективності та точності;
– оцінка ефективності запропонованих методів: провести тестування інструменту на реальних наборах даних, порівняти результати з існуючими рішеннями та визначити можливості для вдосконалення.

Кожна з цих підзадач є важливим етапом у розробці інструменту, що дозволить забезпечити його практичність та ефективність у вирішенні задач моніторингу доменів.

3.2 Обґрунтування вибору методів та засобів

Для досягнення поставленої мети необхідно обрати методи та засоби, які забезпечать ефективність аналізу великих обсягів даних, точність прогнозування та адаптивність інструменту до нових загроз.

3.2.1 Методи дослідження

Обрані методи базуються на поєднанні технологій Big Data та алгоритмів машинного навчання, зокрема:

– класифікація даних: алгоритми, такі як Random Forest та градієнтний бустинг, використовуватимуться для визначення, чи є домен шкідливим, ці методи демонструють високу точність при роботі з великими наборами даних;
– виявлення аномалій: алгоритми, такі як Isolation Forest та методи кластеризації, дозволяють виявляти підозрілу активність у DNS-запитах, навіть якщо домен ще не зареєстрований у чорних списках;

– поведінковий аналіз: метрики поведінки, наприклад, частота запитів до домену або часові патерни, будуть інтегровані у модель для кращого прогнозування.

3.2.2 Інструменти для обробки великих даних

У рамках дослідження передбачається розробка інструменту для прогнозування порушень умов використання доменних імен. Для реалізації цього інструменту необхідно обрати найбільш відповідний інструмент або бібліотеку Big Data, яка забезпечить ефективну обробку великих обсягів даних.

Зважаючи на важливість правильного вибору такого інструменту, було вирішено розв’язати багатокритеріальну задачу для обрання найбільш відповідної платформи.

Для множини альтернатив сформульованої задачі було обрано такі варіанти інструментів та бібліотек Big Data:

- Hadoop: це масштабована та надійна платформа для обробки великих обсягів даних, яка підтримує розподілену обробку даних через використання моделі MapReduce, вона широко застосовується для роботи з великими наборами даних, але може мати обмеження в продуктивності для задач реального часу через пакетну обробку;
- Apache Spark: це високопродуктивна платформа для обробки даних, яка підтримує як пакетну, так і потокову обробку, забезпечує значно вищу швидкість роботи, ніж Hadoop, за рахунок використання пам'яті для обробки даних (in-memory processing), а також підтримує інтеграцію з машинним навчанням, SQL та графовими обчисленнями;
- Apache Flink: орієнтований на потокову обробку даних у реальному часі, забезпечує високий рівень масштабованості та має розвинуту систему управління станом для задач з великими обсягами даних, він часто використовується для додатків, де потрібна мінімальна затримка;
- Apache Storm: це платформа, спеціалізована на потоковій обробці даних у реальному часі, перевагою якої є низька затримка і простота

налаштування, але вона може поступатися Flink у складних завданнях зі станом або великими обсягами даних;

- Presto: це високошвидкісний SQL-двигун для інтерактивного аналізу даних, який підтримує обробку даних із різних джерел, таких як HDFS, AWS S3, тощо, він добре підходить для аналітичних запитів, але має обмеження у випадках потокової обробки.

Крім того, було визначено такі критерії для задачі:

- продуктивність обробки даних: відображає швидкість, з якою платформа може обробляти великі обсяги даних; цей критерій є ключовим, оскільки обробка великих масивів даних з мінімальними затримками необхідна для прогнозування порушень у реальному часі;
- масштабованість: визначає здатність системи ефективно працювати при збільшенні обсягу даних або кількості обчислювальних вузлів; оскільки аналіз великих даних вимагає гнучкості для обробки зростаючих обсягів, цей критерій є обов'язковим;
- легкість інтеграції: визначає, наскільки просто платформа може бути інтегрована з іншими технологіями, такими як мови програмування, бази даних або бібліотеки машинного навчання; інструмент для прогнозування повинен бути легко інтегрованим у загальну інфраструктуру проєкту;
- спільнота підтримки та документація: оцінюється наявність активної спільноти користувачів, які можуть надавати допомогу, а також якість документації, що важливо для зниження ризику технічних проблем і забезпечення швидкого розв’язання питань під час впровадження платформи;
- вартість використання: враховує прямі (ліцензійні) та непрямі витрати (вимоги до апаратного забезпечення, споживання ресурсів); оптимізація витрат важлива для забезпечення економічної ефективності розробки інструменту.

Ці критерії було обрано на основі їхньої релевантності до задачі прогнозування порушень. Вони охоплюють як технічні, так і економічні аспекти

вибору платформи Big Data. Критерії продуктивності та масштабованості дозволяють оцінити, наскільки платформа відповідає технічним вимогам. Легкість інтеграції забезпечує зручність реалізації системи. Спільнота підтримки знижує ризики у процесі впровадження, а вартість використання допомагає оптимізувати бюджет.

Для кожного критерію було визначено відповідний тип шкали, який найкраще описує його природу. Це дозволяє застосувати кількісний або якісний підхід для оцінки альтернатив (див. табл. 3.1).

Таблиця 3.1 – Опис та аналіз шкал за кожним з обраних критеріїв (таблиця виконана самостійно)

Критерій	Тип шкали	Приклади значень	Пояснення
Продуктивність обробки даних	Шкала інтервалів	- 50 млн рядків/сек. - 100 млн рядків/сек. - 150 млн рядків/сек.	Продуктивність вимірюється у швидкості обробки даних, наприклад, у мільйонах рядків за секунду. Значення є числовими, і різниця між ними має зміст.
Масштабованість	Порядкова	- Низька (ефективна для обробки до 1 ТБ даних). - Середня (ефективна для обробки до 10 ТБ даних). - Висока (ефективна для обробки понад 10 ТБ даних).	Масштабованість оцінюється якісно, наприклад, «низька», «середня», «висока». Порядок між рівнями існує, але різницю між ними важко кількісно оцінити.
Легкість інтеграції	Порядкова	- Легка (мінімальні вимоги до інтеграції). - Помірна (потребує певного рівня адаптації). - Складна (потребує значної адаптації та налаштувань).	Інтеграція оцінюється якісно за рівнем складності. Наприклад, «легка», «помірна», «складна». Ці рівні мають порядок, але не кількісну різницю.

Кінець таблиці 3.1

Критерій	Тип шкали	Приклади значень	Пояснення
Спільнота підтримки та документація	Номінальна	- Низька активність (мінімальна кількість документації). - Середня активність (обмежена кількість матеріалів). - Висока активність спільноти (велика кількість форумів, активна спільнота).	Оцінюється кількість доступної документації та рівень активності спільноти користувачів. Значення не мають порядку, вони просто ідентифікують якість підтримки.
Вартість використання	Шкала відношень	- 0 USD (відкрите програмне забезпечення). - 10 000 USD/рік. - 50 000 USD/рік.	Вимірюється у грошових одиницях (доларах, євро тощо). Значення є кількісними, і наявний абсолютний нуль (безкоштовне використання).

Векторний опис альтернатив представлений у вигляді таблиці (див. табл. 3.2), що дозволяє представити всі альтернативи у формалізованому вигляді для подальшого аналізу та використання методів багатокритеріального вибору.

Таблиця 3.2 – Векторний опис альтернатив (таблиця виконана самостійно)

	Продуктивність (млн рядків/сек)	Масштабо-ваність	Легкість інтеграції	Спільнота підтримки	Вартість (USD/рік)
Hadoop	50	Висока	Складна	Висока	10 000
Spark	150	Висока	Помірна	Висока	20 000
Flink	100	Висока	Помірна	Середня	15 000
Storm	80	Середня	Легка	Середня	10 000
Presto	70	Середня	Легка	Середня	5 000

Продуктивність оцінюється кількісно у мільйонах рядків, які платформа здатна обробити за секунду.

Масштабованість оцінюється якісно (низька, середня, висока).

Легкість інтеграції оцінюється якісно (легка, помірна, складна).

Спільнота підтримки оцінюється якісно (низька, середня, висока).

Вартість оцінюється кількісно у доларах США на рік.

Для перетворення якісних шкал у кількісні використовується метод ранжування або бальна оцінка, що дозволяє присвоїти числові значення якісним категоріям. Для нашого прикладу необхідно перетворити такі шкали:

а) масштабованість: якісні рівні («низька», «середня», «висока») перетворюються у числові значення:
   1) низька = 1;
   2) середня = 2;
   3) висока = 3;

б) легкість інтеграції: якісні рівні («легка», «помірна», «складна») перетворюються у числові значення:
   1) легка = 1;
   2) помірна = 2;
   3) складна = 3;

в) спільнота підтримки та документація: якісні рівні («низька», «середня», «висока») перетворюються у числові значення:
   1) низька = 1;
   2) середня = 2;
   3) висока = 3.

Таким чином, критерії «масштабованість», «легкість інтеграції» та «спільнота підтримки» були переведені у кількісні шкали для можливості обчислення загальної корисності альтернатив (див. табл. 3.3).

Таблиця 3.3 – Векторний опис альтернатив за кількісними шкалами (таблиця виконана самостійно)

	Продуктивність (млн рядків/сек)	Масштабо-ваність	Легкість інтеграції	Спільнота підтримки	Вартість (USD/рік)
Hadoop	50	3	3	3	10 000
Spark	150	3	2	3	20 000

Кінець таблиці 3.3

	Продуктивність (млн рядків/сек)	Масштабо-ваність	Легкість інтеграції	Спільнота підтримки	Вартість (USD/рік)
Flink	100	3	2	2	15 000
Storm	80	2	1	2	10 000
Presto	70	2	1	2	5 000

Перетворення векторного опису до принципу оптимальності «за максимумом» полягає в тому, щоб змінити напрямок всіх критеріїв, які орієнтовані «на мінімум» (наприклад, вартість або складність інтеграції), так, щоб вони були співставні з критеріями, орієнтованими «на максимум». Це дозволяє уніфікувати шкали і спростити обчислення (див. табл. 3.4).

Таблиця 3.4 – Векторний опис альтернатив після перетворення (таблиця виконана самостійно)

	Продуктивність (млн рядків/сек)	Масштабо-ваність	Легкість інтеграції	Спільнота підтримки	Вартість (USD/рік)
Hadoop	50	3	0.0	3	0.67
Spark	150	3	0.5	3	0.0
Flink	100	3	0.5	2	0.33
Storm	80	2	1.0	2	0.67
Presto	70	2	1.0	2	1.0

Після приведення всіх значень до єдиної шкали, у межах від 0 до 1, отримуємо нормований векторний опис альтернатив (див. табл. 3.5).

Таблиця 3.5 – Нормований векторний опис альтернатив (таблиця виконана самостійно)

	Продуктивність (млн рядків/сек)	Масштабо-ваність	Легкість інтеграції	Спільнота підтримки	Вартість (USD/рік)
Hadoop	0.0	1.0	0.0	1.0	0.67
Spark	1.0	1.0	0.5	1.0	0.0
Flink	0.5	1.0	0.5	0.0	0.33
Storm	0.3	0.0	1.0	0.0	0.67
Presto	0.2	0.0	1.0	0.0	1.0

Присвоїмо кожному критерію ранг на основі його важливості та обчислимо вагові коефіцієнти:

- продуктивність – 0,33;
- масштабованість – 0,27;
- легкість інтеграції – 0,2;
- спільнота підтримки – 0,1;
- вартість – 0,1.

Розрахуємо значення корисності для кожної альтернативи шляхом підсумовування добутків нормованих значень на вагові коефіцієнти за формулою 3.1:

$$Z_i = \sum_{j=1}^{n} a_{ij} \cdot \beta_j \tag{3.1}$$

де Z_i – загальна корисність альтернативи i;

a_{ij} – нормоване значення критерію j для альтернативи i;

β_j – ваговий коефіцієнт критерію j, який відображає його відносну важливість;

n – кількість критеріїв.

Отримані значення занесемо до таблиці розрахунку корисності альтернатив (див. табл. 3.6).

Таблиця 3.6 – Розрахунки корисності альтернатив (таблиця виконана самостійно)

	Продуктив-ність (млн рядків/сек)	Масшта-бованість	Легкість інтеграції	Спільнота підтримки	Вартість (USD/рік)	**Корисність**
Hadoop	0,00	1,00	0,00	1,00	0,67	**0,44**
Spark	1,00	1,00	0,50	1,00	0,00	**0,80**
Flink	0,50	1,00	0,50	0,00	0,33	**0,57**
Storm	0,30	0,00	1,00	0,00	0,67	**0,37**
Presto	0,20	0,00	1,00	0,00	1,00	**0,37**
Ваги	0,33	0,27	0,20	0,10	0,10	

З таблиці вище можемо побачити, що найкращою альтернативою є Apache Spark, оскільки вона має найвищу корисність – 0.8. Ця альтернатива має найвищі оцінки за ключовими критеріями, такими як «продуктивність», «масштабованість» та «підтримка спільноти». Водночас, це рішення є найдорожчим серед представлених.

Apache Flink посідає друге місце, демонструючи гарний баланс між продуктивністю, масштабованістю та вартістю.

Hadoop займає середню позицію, виграючи за критеріями «масштабованість» та «спільнота підтримки», але програючи за продуктивністю та легкістю інтеграції.

Storm і Presto мають найнижчі значення корисності, через слабкі показники за кількома ключовими критеріями («продуктивність», «масштабованість», «спільнота підтримки»), проте виграють за легкістю інтеграції та ціною.

Отже, альтернатива Spark є оптимальним вибором для вирішення задачі з урахуванням вагових коефіцієнтів та нормованих значень критеріїв. Її коефіцієнт корисності значно перевищує значення інших альтернатив.

3.2.3 Бібліотеки та платформи машинного навчання

Бібліотеки TensorFlow та PyTorch дозволяють створювати складні моделі машинного навчання, включаючи глибокі нейронні мережі, для прогнозування загроз.

Бібліотека Scikit-learn використовуватиметься для реалізації класичних алгоритмів класифікації та кластеризації.

3.2.4 Причини вибору методів та інструментів

Обрані методи та засоби обґрунтовані їхньою ефективністю у вирішенні задач кібербезпеки. Вони забезпечують основу для розробки універсального інструменту прогнозування:

– масштабованість: обрані технології дозволяють працювати з великими обсягами даних, зберігаючи високу продуктивність;

– точність: алгоритми машинного навчання, що застосовуються, забезпечують високу точність класифікації;
– гнучкість: інструменти, такі як TensorFlow, дозволяють адаптувати моделі до нових загроз завдяки можливості навчання на нових наборах даних.

3.3 Необхідні ресурси

Для виконання проєкту з розробки інструменту прогнозування порушень умов використання доменних імен потрібно забезпечити доступ до певних ресурсів. Вони охоплюють як технічні, так і інформаційні складові, які є ключовими для реалізації поставлених задач:

а) обчислювальні ресурси:
   1) потужні сервери або хмарні платформи для обробки великих обсягів даних та тренування моделей машинного навчання;
   2) локальні обчислювальні ресурси для тестування прототипів;

б) програмне забезпечення:
   1) платформи Big Data;
   2) бібліотеки машинного навчання;
   3) інструменти візуалізації даних, такі як Tableau або Power BI, для створення звітів і графічного представлення результатів аналізу;

в) дані:
   1) набори DNS-запитів для аналізу патернів доступу до доменів;
   2) інформація з баз даних WHOIS для вивчення характеристик доменів (дата реєстрації, IP-адреси, контактні дані реєстрантів);
   3) набори поведінкових метрик, наприклад, частота запитів до певних доменів або географічна локалізація запитів;

г) література та документація:
   1) наукові статті, книги та технічна документація для формування теоретичної бази дослідження;
   2) керівництва користувача та документація для обраних платформ і бібліотек.

3.4 Вимоги до інструменту прогнозування

Розробка інструменту для прогнозування порушень умов використання доменних імен потребує чіткого визначення вимог до його функціональності, ефективності та адаптивності. Такий інструмент повинен відповідати сучасним викликам кібербезпеки, забезпечувати високу точність прогнозів та здатність працювати з великими обсягами даних у реальному часі.

Основною вимогою до інструменту є його здатність інтегрувати різноманітні типи даних. Для якісного прогнозування необхідно враховувати дані з баз WHOIS, DNS-запити, інформацію про трафік і поведінку користувачів. Це дозволить створити багатовимірну модель для виявлення потенційно шкідливих доменів. Зокрема, дані з баз WHOIS можуть містити інформацію про реєстрацію доменів, а аналіз DNS дозволяє виявляти аномалії в запитах.

Ще однією ключовою вимогою є висока точність прогнозів. Інструмент має забезпечувати точність не нижче 90% при класифікації доменів як шкідливих або безпечних. Для досягнення цього можна застосовувати методи машинного навчання, такі як дерева рішень, глибоке навчання або градієнтний бустинг. Також важливим є мінімізація кількості хибнопозитивних результатів, щоб уникнути помилкових блокувань доменів.

Швидкодія інструменту також є критично важливою. Для виявлення загроз у реальному часі система повинна забезпечувати обробку великих обсягів даних із затримкою не більше кількох секунд. Використання розподілених обчислень на базі Hadoop або Spark може значно прискорити цей процес.

Масштабованість є ще однією важливою характеристикою. Інструмент має легко адаптуватися до зростання обсягу даних та збільшення кількості запитів. Це особливо важливо для аналізу в глобальних масштабах, коли щодня обробляються мільйони DNS-запитів.

Додатково необхідно враховувати аспекти безпеки та конфіденційності. Інструмент повинен відповідати вимогам загальносвітових стандартів кібербезпеки, таких як ISO/IEC 27001, та забезпечувати захист даних користувачів від несанкціонованого доступу.

Таким чином, інструмент прогнозування повинен поєднувати потужну функціональність, точність та швидкість роботи з можливістю масштабування й гарантування високого рівня безпеки. Лише за умови дотримання цих вимог можна створити ефективний інструмент для моніторингу та попередження порушень у сфері використання доменних імен.

3.5 Очікувані результати

Реалізація задач дослідження передбачає отримання низки конкретних результатів, які зроблять внесок у розвиток інструментів аналізу та прогнозування порушень умов використання доменних імен.

По-перше, це розробка моделі, яка інтегрує дані з різних джерел (DNS-запити, бази WHOIS, поведінкові метрики) та використовує сучасні алгоритми машинного навчання для аналізу загроз. Модель буде базуватися на технологіях Big Data, що забезпечить її здатність працювати з великими обсягами даних.

По-друге, проведення тестування запропонованих алгоритмів та моделей на наборах даних, що дозволить оцінити їхню точність, швидкодію та адаптивність до нових типів загроз. Очікується, що точність класифікації шкідливих доменів досягне не менше 90%.

На основі отриманих результатів буде сформульовано рекомендації для подальшого вдосконалення інструментів моніторингу доменних імен. Це може включати розширення функціоналу моделей, оптимізацію алгоритмів або інтеграцію з іншими системами кібербезпеки.

Створений прототип інструменту може бути адаптований для використання у реальних умовах, наприклад, у компаніях, що спеціалізуються на кібербезпеці, або реєстраторами доменних імен для моніторингу активності доменів.

Розроблені підходи та результати дослідження можуть слугувати основою для подальших наукових досліджень у сфері кібербезпеки, аналізу великих даних та машинного навчання. Таким чином, результати дослідження сприятимуть підвищенню ефективності моніторингу доменних імен, що є важливим кроком у боротьбі з кіберзагрозами та захисті інтернет-інфраструктури.

4 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

Цей розділ описує методи, технології, підходи та алгоритми, які будуть використовуватися для розробки інструменту прогнозування порушень умов використання доменних імен. Зокрема, приділяється увага архітектурі програмного забезпечення, проектуванню структури зберігання даних, використовуваним алгоритмам та методам аналізу, а також дизайну системи.

Описані елементи відіграють ключову роль у створенні ефективного, масштабованого та адаптивного інструменту. Вибір підходів базується на їхній здатності працювати з великими обсягами даних, інтегрувати інформацію з різних джерел та забезпечувати високий рівень точності виявлення потенційних загроз.

Метою розділу є детальний опис концептуальної та технічної бази інструменту, який буде реалізовано в межах проєкту. Це включає обґрунтування обраних рішень, деталі їх реалізації та приклади використання.

4.1 Архітектура та проектування ПЗ

Розробка інструменту прогнозування порушень умов використання доменних імен потребує створення чіткої архітектури, яка забезпечить ефективну взаємодію між компонентами системи. Основними компонентами архітектури є модуль збору даних, модуль аналізу, база даних та інтерфейс користувача.

4.1.1 Загальна структура архітектури

Архітектура системи передбачає використання розподіленої моделі обробки даних. Основні компоненти:

- модуль збору даних: відповідає за отримання даних із різних джерел, таких як DNS-запити, бази WHOIS та поведінкові метрики, для цього використовуються API-інтеграції, наприклад, Google Public DNS та WHOIS API;
- модуль аналізу: виконує основну обробку даних та застосовує алгоритми машинного навчання для класифікації доменів і виявлення аномалій;

– база даних: використовується для зберігання сирих даних, результатів аналізу та моделей машинного навчання;
– інтерфейс користувача: дашборд для візуалізації результатів аналізу, надання рекомендацій та інтерактивної взаємодії з користувачем.

4.1.2 Структура зберігання даних

Зберігання даних реалізовано на основі комбінованого підходу:
– SQL-база даних (PostgreSQL) для зберігання структурованих даних, таких як записи про домени, результати класифікації;
– NoSQL-база даних (MongoDB) для зберігання великих обсягів логів DNS-запитів та поведінкових метрик, що забезпечує швидкий доступ до напівструктурованих даних;
– резервне копіювання даних у хмарному середовищі для забезпечення їхньої безпеки.

4.1.3 Візуалізація

Для візуалізації архітектури системи розроблено кілька діаграм.

Use Case діаграма моделює основні сценарії взаємодії користувача з системою, наприклад, аналіз шкідливих доменів (див. рис. 4.1).

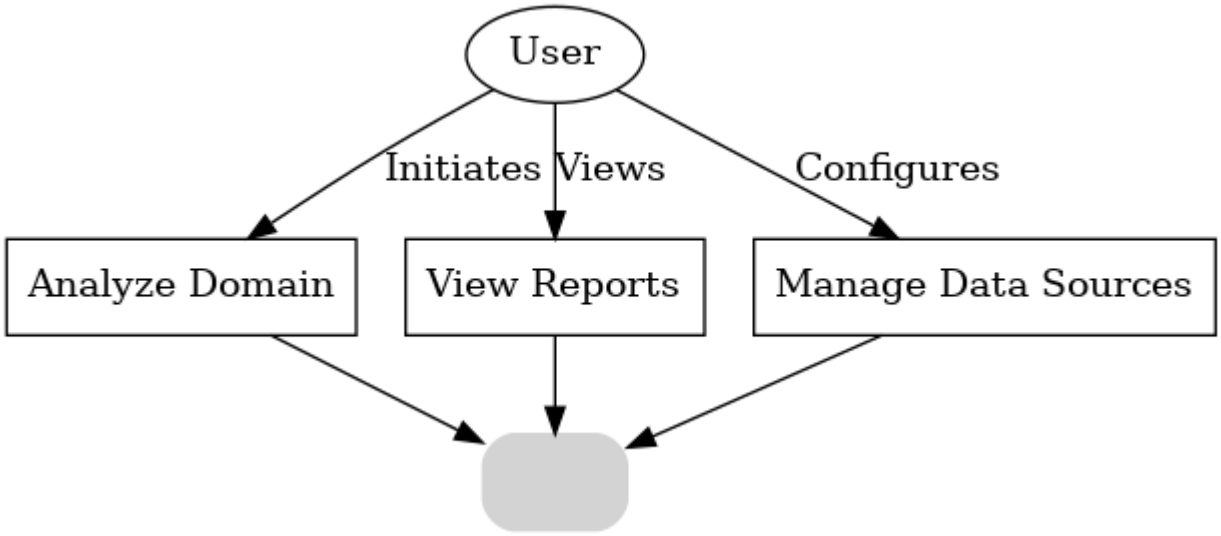

Рисунок 4.1 – Use Case діаграма (рисунок створено самостійно)

UML-діаграма компонентів відображає зв’язки між основними модулями системи (див. рис. 4.2).

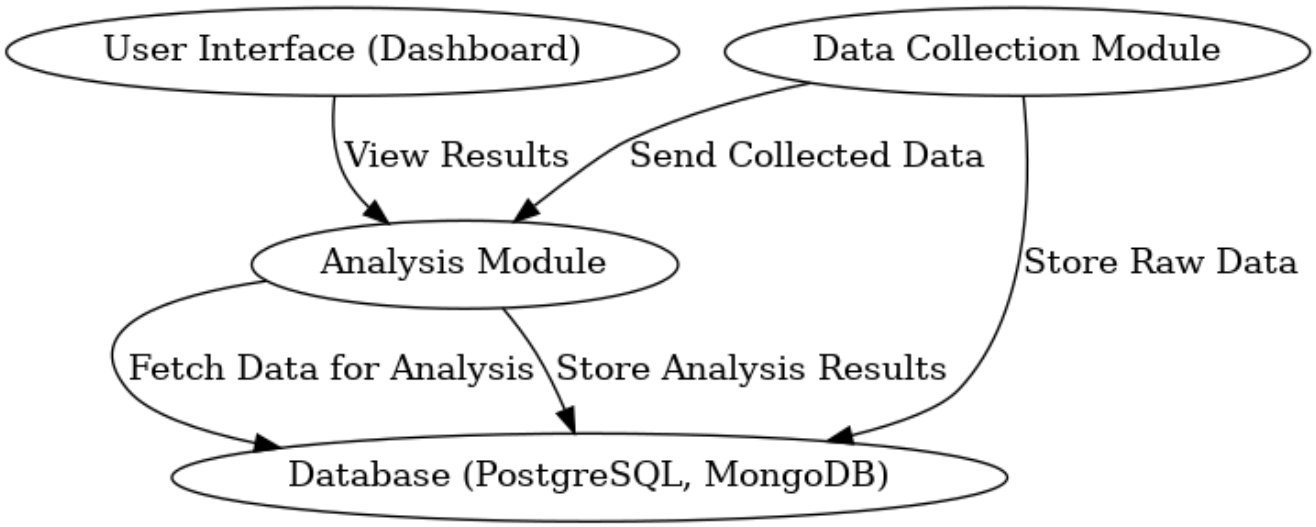

Рисунок 4.2 – UML-діаграма (рисунок створено самостійно)

Схема бази даних включає таблиці для структурованих даних (наприклад, домени, результати аналізу) та колекції для логів DNS-запитів (див. рис. 4.3).

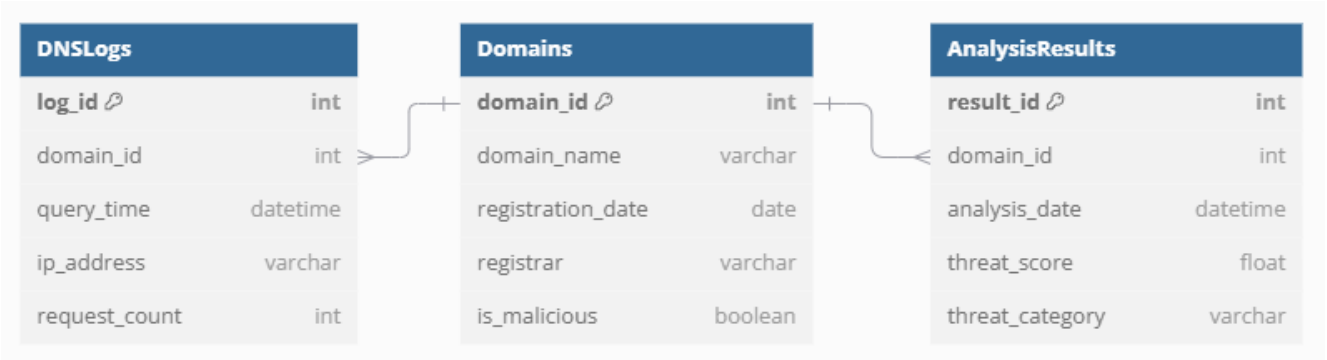

Рисунок 4.3 – Схема бази даних (рисунок створено самостійно)

Архітектура програмного забезпечення побудована таким чином, щоб забезпечити гнучкість, масштабованість та високу продуктивність. Вона дозволяє легко інтегрувати нові джерела даних та адаптувати алгоритми до змін у кіберзагрозах.

4.2 Проектування структури зберігання даних

Зберігання даних є важливим аспектом розробки інструменту прогнозування, оскільки саме на основі даних виконується аналіз і приймаються рішення. Для забезпечення ефективності системи було обрано гібридну модель зберігання даних, яка поєднує реляційні та NoSQL бази даних.

4.2.1 Вибір технологій

PostgreSQL використовуватиметься для зберігання структурованих даних, таких як записи про домени (ID, ім’я, реєстратор, статус) та результати аналізу. PostgreSQL забезпечує ACID-транзакції та підтримує складні запити, що важливо для точності аналізу [15].

MongoDB призначена для зберігання напівструктурованих і неструктурованих даних, таких як логи файлів DNS-запитів [16]. Ця база даних забезпечує високу швидкість обробки великих обсягів даних та гнучкість у роботі з JSON-подібними документами.

Ця структура забезпечує гнучкість, швидкодію та надійність системи зберігання даних, що є критично важливим для аналізу великих обсягів інформації.

4.2.2 Схема бази даних

Схема бази даних побудована таким чином, щоб забезпечити зручний доступ до даних та їх інтеграцію для аналізу:

– таблиця Domains містить основну інформацію про домени;
– таблиця DNSLogs зберігає записи про DNS-запити, включаючи час, IP-адресу та кількість запитів;
– таблиця AnalysisResults зберігає результати аналізу, зокрема оцінку загрози та категорію загрози.

Схему бази даних було представлено в роботі раніше (див. рис. 4.3).

4.2.3 Резервування та масштабованість

Для забезпечення надійності зберігання даних необхідно реалізувати резервне копіювання бази даних у хмарі (наприклад, Amazon S3 або Google Cloud Storage).

Масштабованість забезпечується використанням кластерів бази даних PostgreSQL та шардингу у MongoDB.

4.2.4 Інтеграція даних

Інтеграція різних типів даних здійснюється через ETL-процеси (Extract, Transform, Load), які дозволяють об'єднувати дані з кількох джерел. Наприклад, дані WHOIS можуть бути оброблені для оновлення інформації про домени, а DNS-логи аналізуються для виявлення підозрілих патернів у реальному часі.

4.3 Алгоритми та методи

Розробка інструменту прогнозування порушень умов використання доменних імен вимагає використання сучасних алгоритмів машинного навчання та методів аналізу даних, які забезпечують високу точність і масштабованість. У цьому підрозділі наведено приклади найцікавіших алгоритмів, їхній математичний опис та обґрунтування наукової новизни.

4.3.1 Алгоритми класифікації

Random Forest є ансамблевим методом класифікації, який використовує множину дерев рішень. Кожне дерево створюється на випадковій підмножині даних, а результат класифікації визначається голосуванням дерев. Формула для передбачення класу виглядає так (формула 4.1):

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \dots, T_n(x)\} \quad (4.1)$$

де $T_i(x)$ – передбачення і-го дерева.

Цей алгоритм демонструє високу стійкість до перенавчання та добре працює з великими обсягами даних.

Градієнтний бустинг – це метод, який поєднує послідовно побудовані слабкі моделі (дерева рішень). Кожне нове дерево навчається на помилках попередніх моделей, мінімізуючи функцію втрат (формула 4.2):

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x) \tag{4.2}$$

де $F_m(x)$ – модель на m-ій ітерації;

η – швидкість навчання;

$h_m(x)$ – слабкий учасник.

4.3.2 Методи виявлення аномалій

Isolation Forest використовує ізолювання точок даних шляхом створення випадкових поділів у просторах ознак. Шкідливі домени можуть бути ідентифіковані за допомогою коротких шляхів ізоляції (формула 4.3):

$$AnomalyScore = 2^{\frac{-E(h(x))}{c(n)}} \tag{4.3}$$

де $E\big(h(x)\big)$ – середня довжина шляху до ізоляції точки x;

$c(n)$ – нормалізація.

Кластеризація k-means ділить дані на k кластерів, мінімізуючи відстань між точками в кластері та його центроїдом (формула 4.4):

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{4.4}$$

де μ_i – центроїд і-го кластера;

C_i – набір точок у кластері.

4.3.3 Інтеграція Big Data

MapReduce – метод розподіленої обробки даних, що дозволяє розділити завдання на два основні етапи: Map (перетворення даних) і Reduce (агрегація результатів). Використовується для обробки великих наборів даних у системах Hadoop і Spark.

Spark MLlib – набір алгоритмів машинного навчання у Spark, який забезпечує паралельну обробку великих даних, включаючи класифікацію, регресію та кластеризацію. Spark MLlib підтримує реалізацію таких алгоритмів, як градієнтний бустинг і Random Forest.

4.4 UI/UX дизайн системи

Дизайн користувацького інтерфейсу (UI) та користувацького досвіду (UX) відіграє ключову роль у створенні інструменту, який буде зрозумілим і зручним для використання. Основні принципи дизайну орієнтовані на забезпечення швидкого доступу до аналітичних результатів, інтерактивності та інтуїтивності інтерфейсу.

4.4.1 Основні функціональні можливості інтерфейсу

Інтерфейс інструменту реалізує такі ключові функції:

а) дашборд аналітики: головний екран, який відображає результати аналізу доменів у вигляді графіків, таблиць та карт, наприклад:
   1) список підозрілих доменів із оцінками загроз;
   2) візуалізація патернів запитів (часові графіки, географічне розташування тощо);

б) функція пошуку: користувач може шукати інформацію за конкретним доменом, отримуючи деталі про його реєстрацію, історію запитів та результати аналізу;

в) налаштування параметрів аналізу: інтерфейс дозволяє змінювати налаштування алгоритмів, наприклад, поріг загрози для класифікації доменів.

4.4.2 Принципи UX-дизайну

UX-дизайн побудований на основі наступних принципів:

– інтуїтивність: мінімум навчання для користувача завдяки зрозумілій структурі інтерфейсу та логічному розташуванню елементів;
– інтерактивність: можливість взаємодії з даними, деталізація інформації за кліком, налаштування графіків та вибір метрик;
– доступність: оптимізація інтерфейсу для різних пристроїв (десктопи, планшети, смартфони) через адаптивний дизайн.

4.4.3 Дизайн-схеми

Для розробки інтерфейсу було створено декілька дизайн-схем.

Головний дашборд – основний екран, що містить аналітичні віджети (див. рис. 4.4), такі як:

– лінійний графік запитів до підозрілих доменів;
– таблиця з характеристиками доменів;
– карта географічного розташування запитів.

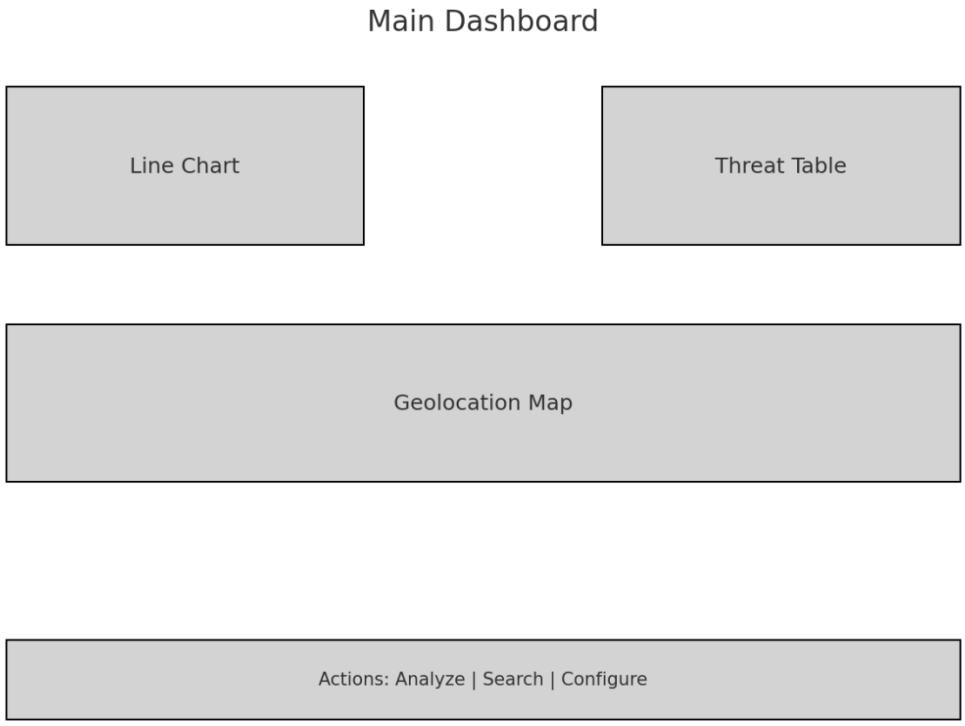

Рисунок 4.4 – Макет головного дашборду (рисунок створено самостійно)

Екран деталізації – вікно, яке відкривається при виборі конкретного домену, що містить більш детальну інформацію, таку як реєстрація, IP-адреси, оцінка загрози тощо (див. рис. 4.5).

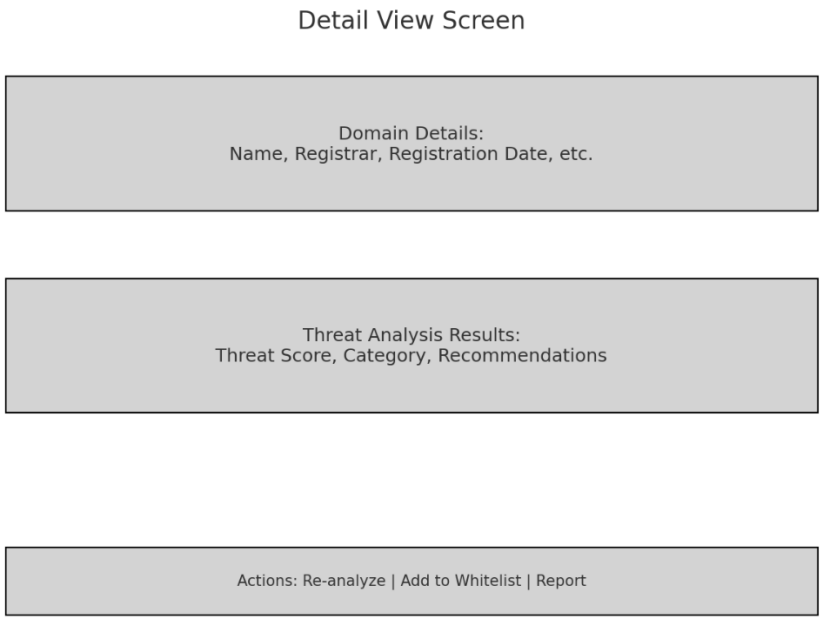

Рисунок 4.5 – Макет екрану деталізації (рисунок створено самостійно)

Екран налаштувань – простий інтерфейс для зміни алгоритмів, параметрів порогів та інтеграції нових джерел даних (див. рис. 4.6).

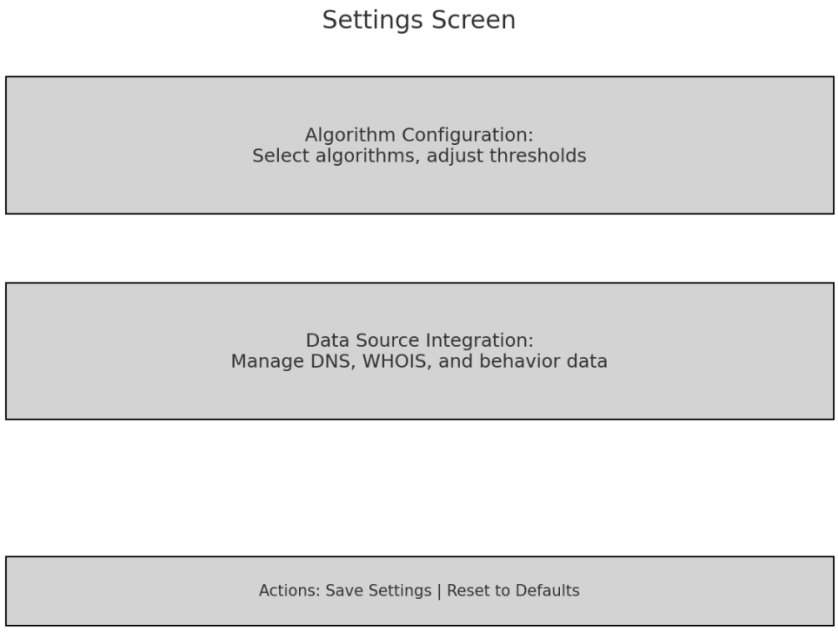

Рисунок 4.6 – Макет екрану налаштувань (рисунок створено самостійно)

4.4.4 Використання інструментів

Для реалізації UI/UX можливо використати такі інструменти:

– Figma: для створення прототипів і мокапів інтерфейсу;
– React.js: для розробки інтерактивного веб-інтерфейсу;
– Chart.js: для візуалізації даних у вигляді графіків і діаграм.

4.5 Інші елементи, важливі для реалізації проєкту

Для забезпечення повноцінної реалізації інструменту прогнозування порушень умов використання доменних імен слід врахувати кілька додаткових аспектів, які мають вплив на функціональність, безпеку та масштабованість системи.

4.5.1 Інтеграція з існуючими системами

Інструмент повинен бути здатним інтегруватися з іншими системами кібербезпеки та моніторингу:

– інтеграція API: система повинна мати підтримку для підключення до API зовнішніх джерел даних, таких як Google Public DNS або сторонні бази WHOIS;
– сумісність: інструмент має забезпечувати сумісність із популярними SIEM-платформами (наприклад, Splunk, IBM QRadar) для обміну даними та автоматизації процесів аналізу.

4.5.2 Масштабованість

Масштабованість системи є критично важливим аспектом, враховуючи можливе збільшення обсягу даних:

– горизонтальне масштабування: використання контейнерних технологій (Docker, Kubernetes) для розподілу обчислювальних завдань між кількома вузлами;
– масштабовані сховища: хмарні рішення від Amazon або Google забезпечують надійне зберігання та доступ до великих обсягів даних.

4.5.3 Безпека

Безпека є одним із ключових аспектів розробки:

- шифрування даних: дані, що зберігаються у базах, та під час передавання повинні бути зашифровані за допомогою сучасних протоколів, таких як AES-256 та TLS 1.3;
- автентифікація та авторизація: використання багатофакторної автентифікації (MFA) та ролей для контролю доступу до системи;
- логування подій: ведення журналу подій для відстеження дій користувачів та аудиту.

4.5.4 Моніторинг і підтримка

Система повинна забезпечувати стабільну роботу навіть під час високого навантаження:

- моніторинг стану системи: використання рішень, таких як Prometheus та Grafana, для моніторингу продуктивності компонентів;
- резервування: регулярне резервне копіювання баз даних та критичних компонентів системи.

4.5.5 Документація

Для забезпечення легкості впровадження і використання системи слід створити повну технічну документацію:

- керівництво користувача з описом функцій інтерфейсу;
- інструкції для розгортання системи в локальному чи хмарному середовищі;
- опис API для інтеграції з іншими системами.

4.6 Висновки з теоретичного дослідження

У межах цього розділу було розглянуто ключові аспекти розробки інструменту для прогнозування порушень умов використання доменних імен.

Проведений аналіз дозволив визначити основні елементи, які забезпечують ефективність, масштабованість та безпеку системи.

Було запропоновано модульну архітектуру, яка забезпечує гнучкість системи та зручність інтеграції з іншими інструментами. Діаграми компонентів, бази даних і сценаріїв використання дають чітке уявлення про структуру інструменту.

Використання гібридної моделі зберігання даних, що поєднує PostgreSQL для структурованих даних та MongoDB для напівструктурованих даних, забезпечує високу продуктивність та масштабованість системи.

Запропоновані алгоритми класифікації (Random Forest, градієнтний бустинг) та методи виявлення аномалій (Isolation Forest, кластеризація) довели свою ефективність у контексті прогнозування загроз. Технології Big Data, такі як Spark MLlib, гарантують можливість роботи з великими обсягами даних.

Продуманий дизайн інтерфейсу дозволяє користувачам легко взаємодіяти із системою, швидко отримувати аналітичні дані та налаштовувати параметри аналізу.

Особлива увага приділяється інтеграції, безпеці та масштабованості системи. Використання сучасних технологій, таких як Docker та Kubernetes, забезпечує стабільну роботу навіть під час значного навантаження.

Запропоновані рішення відповідають меті дослідження, забезпечуючи всі необхідні інструменти для аналізу та прогнозування порушень у використанні доменних імен. Розглянуті технології та методи не лише вирішують поточні виклики, але й створюють основу для подальшого вдосконалення системи.

Розроблена концепція може бути використана для створення прототипу інструменту, який після тестування та доопрацювання стане частиною реальної інфраструктури кібербезпеки. Подальші дослідження мають бути спрямовані на оптимізацію обчислювальних ресурсів, інтеграцію нових джерел даних та вдосконалення алгоритмів.

5 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ

5.1 Вимоги до системи

Розроблювана система має на меті аналіз та виявлення потенційних кіберзагроз на основі аналізу доменних імен, DNS-даних та поведінкових характеристик. Основними користувачами системи є аналітики з кібербезпеки, дослідники загроз, а також системи автоматичного моніторингу.

5.1.1 Функціональні вимоги

Розроблювана система повинна:

- приймати великі обсяги DNS-логів або списків доменних імен у реальному часі або у вигляді файлів;
- виконувати попередню обробку, нормалізацію та очищення даних;
- використовувати алгоритми машинного навчання для класифікації або оцінки потенційної загрози доменного імені;
- забезпечити аналітичні можливості для перегляду, фільтрації та візуалізації результатів класифікації;
- надавати інтерфейс для взаємодії користувача з результатами (через веб-інтерфейс або API);
- зберігати історію запитів, рішень і результатів аналізу.

5.1.2 Нефункціональні вимоги

Серед нефункціональних вимог варто виділити такі:

- система має бути здатна працювати з великими обсягами даних (Big Data), використовуючи масштабовані інструменти, зокрема Apache Spark;
- обробка даних має бути побудована модульно, з можливістю масштабування окремих компонентів;
- усі обробки та класифікації мають бути виконані в межах прийнятного часу (мінімальні затримки при роботі з потоковими даними);
- UI має бути інтуїтивно зрозумілим і адаптованим під аналітичну роботу.

5.1.3 Вхідні дані

Система має приймати такі вхідні дані:

– списки доменних імен;
– DNS-запити (формат PCAP або лог-файли);
– WHOIS-інформація, дані про IP-адреси;
– інші метадані (час, TTL, кількість запитів тощо).

5.1.4 Вихідні дані:

Вихідними даними роботи програми мають бути:

– оцінка загроз за кожним доменом (наприклад, «шкідливий», «підозрілий», «безпечний»);
– ймовірність належності до зловмисної активності;
– інтерактивні графіки, таблиці, аналітичні звіти;
– можливість експорту результатів у CSV або JSON.

5.2 Вибір технологій та середовища розробки

Вибір технологій для реалізації практичної частини кваліфікаційної роботи був зумовлений специфікою поставленої задачі, необхідністю обробки великих обсягів даних, а також потребою у зручному, масштабованому та гнучкому середовищі розробки. З огляду на це було обрано стек технологій, який дозволяє ефективно реалізувати функціонал аналізу DNS-записів та доменів з використанням методів Big Data.

Основним інструментом для обробки даних виступає Apache Spark – високопродуктивна платформа для розподіленої обробки даних, яка забезпечує гнучкість при роботі з різними джерелами даних, підтримку мов програмування Python, Scala та Java, а також наявність бібліотек для машинного навчання та SQL-аналітики.

Для реалізації логіки обробки даних обрано мову програмування Python, яка має широку екосистему бібліотек для аналізу даних (наприклад, pandas, numpy, scikit-learn, matplotlib), а також підтримується в Apache Spark через модуль PySpark.

Система зберігання даних реалізується на основі Apache Parquet – колонкового формату зберігання, оптимізованого для аналітичних задач і сумісного з Apache Spark. У разі потреби масштабування може бути використане сховище типу HDFS або хмарні аналоги.

Таким чином, обрані інструменти дозволяють забезпечити ефективну реалізацію системи аналізу доменних імен з урахуванням особливостей великих даних, вимог до масштабованості та зручності візуалізації результатів.

5.3 Архітектура та структура системи

Архітектура прототипу системи (див. рис. 5.1) побудована на мікросервісному підході з акцентом на обробку великих обсягів даних. Основні компоненти взаємодіють через стандартизовані API та черги повідомлень. Обробка даних виконується за допомогою Apache Spark, що забезпечує горизонтальне масштабування, а UI реалізовано як окремий веб-застосунок з аналітичним інтерфейсом.

Основні компоненти архітектури:

а) компонент збору та завантаження даних: приймає вхідні файли DNS-логів або списки доменів, виконує валідацію та запис у сховище (наприклад, HDFS або об’єктне сховище типу MinIO або Amazon S3);

б) сховище даних: вхідні та оброблені дані зберігаються у двох формах:
   1) сире сховище для необроблених даних;
   2) очищене сховище у вигляді структурованих таблиць (наприклад, Parquet), з якими працює Apache Spark;

в) обробник даних (Data Processing Engine): цей компонент побудований на Apache Spark з використанням PySpark та MLlib, здійснює:
   1) нормалізацію DNS-даних;
   2) агрегацію та обчислення характеристик (наприклад, ентропія доменного імені, кількість запитів);
   3) виявлення підозрілих шаблонів з використанням машинного навчання;

г) API-сервер: забезпечує доступ до результатів аналізу, приймає запити з інтерфейсу користувача, а також може бути використаний для інтеграції з зовнішніми сервісами;

д) фронтенд (веб-інтерфейс): побудований з використанням сучасного фреймворку React, він надає можливість переглядати загальний стан системи, фільтрувати результати, переглядати деталізацію по кожному домену, а також експортувати дані;

е) компонент логування та моніторингу: всі сервіси логують свою діяльність у централізовану систему (наприклад, ELK Stack або Prometheus та Grafana).

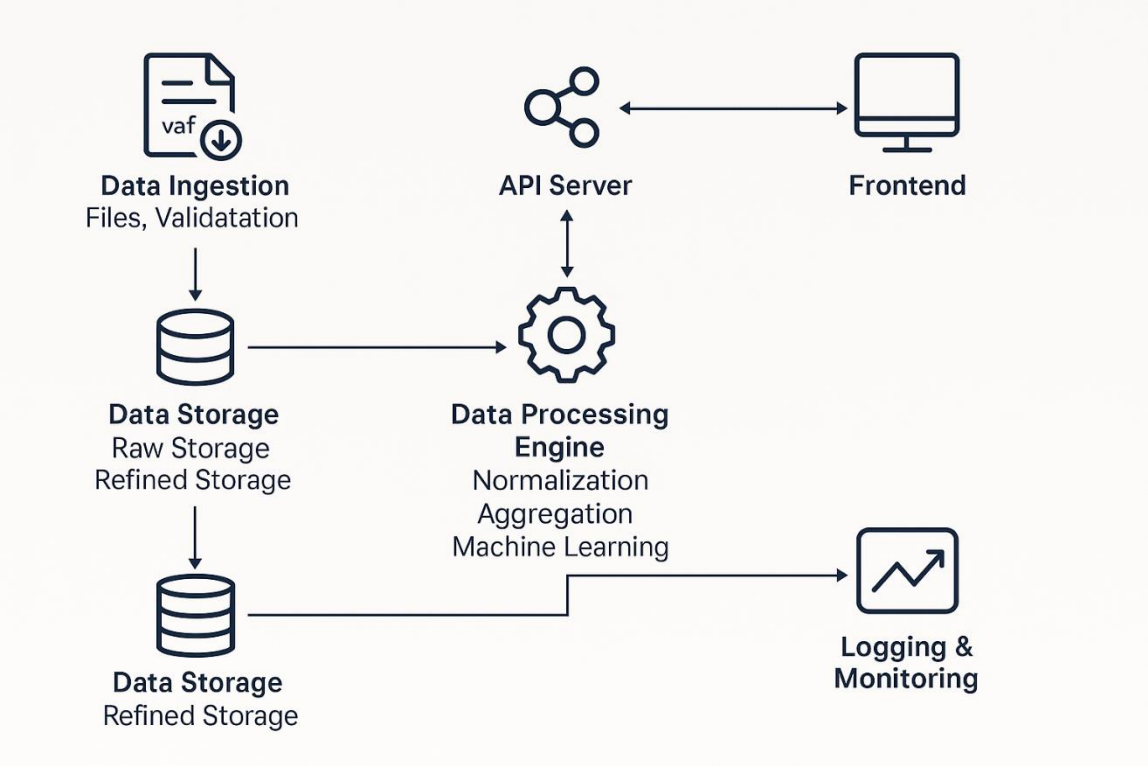

Рисунок 5.1 – Схема архітектури системи (рисунок створено самостійно)

Стиль архітектури включає:

– мікросервіси для гнучкості та масштабування;
– асинхронну обробку даних через черги (Kafka або RabbitMQ);
– REST API для взаємодії між компонентами;
– використання контейнеризації (Docker) для ізоляції сервісів і можливості розгортання в хмарі або локально.

5.4 Алгоритми та методи обробки і класифікації доменних імен

На цьому етапі проєкту реалізовано підсистему інтелектуального аналізу та класифікації доменних імен на основі методів машинного навчання та інструментів обробки великих даних. Основною задачею є виявлення потенційно шкідливих або підозрілих доменів шляхом їхнього аналізу та класифікації на основі сукупності ознак.

Дані для аналізу надходять із DNS-логів, WHOIS-реєстрів, а також публічних списків відомих шкідливих доменів. Вони мають різну структуру, значні обсяги та надходять у режимі наближеному до реального часу, що обумовлює потребу в використанні технологій Big Data. Для цього у системі використовується фреймворк Apache Spark, який забезпечує паралельну обробку великих масивів даних та інтеграцію з модулями машинного навчання.

На основі попередньо зібраних даних формується вхідна вибірка, яка проходить декілька етапів обробки. На першому етапі здійснюється очищення та нормалізація даних, включаючи видалення дублікатів, перетворення доменів до уніфікованого формату та фільтрацію за ключовими параметрами. Наступним кроком є генерація ознак, зокрема таких, як довжина доменного імені, частота появи в логах, наявність або відсутність WHOIS-інформації, частка цифр, ентропія символів, географія IP-адрес, історія реєстрації тощо.

Одержані вектори ознак подаються на вхід моделі класифікації, яка була навчена за допомогою Apache Spark MLlib. Серед протестованих алгоритмів – логістична регресія, дерева рішень, випадкові ліси та градієнтні бустингові моделі. Вибір фінального класифікатора здійснено на основі оцінки якості за метриками точності, повноти та F1-мірою.

Навчання здійснюється на попередньо розмічених вибірках, які включають як безпечні, так і шкідливі доменні імена. Балансування класів досягається за рахунок підбору рівнозначної кількості зразків для кожної категорії, а також шляхом генерації додаткових доменів на основі шаблонів, характерних для шкідливих генераторів доменних імен (DGA).

Результатом застосування навченої моделі є класифікація нових доменних імен на підставі їхніх ознак із подальшою передачею результатів до модуля візуалізації або до підсистеми автоматизованого оповіщення (див. рис. 5.2). Завдяки використанню Spark-платформи обробка та класифікація можуть виконуватись у розподіленому середовищі, що забезпечує масштабованість і придатність рішення до реального використання.

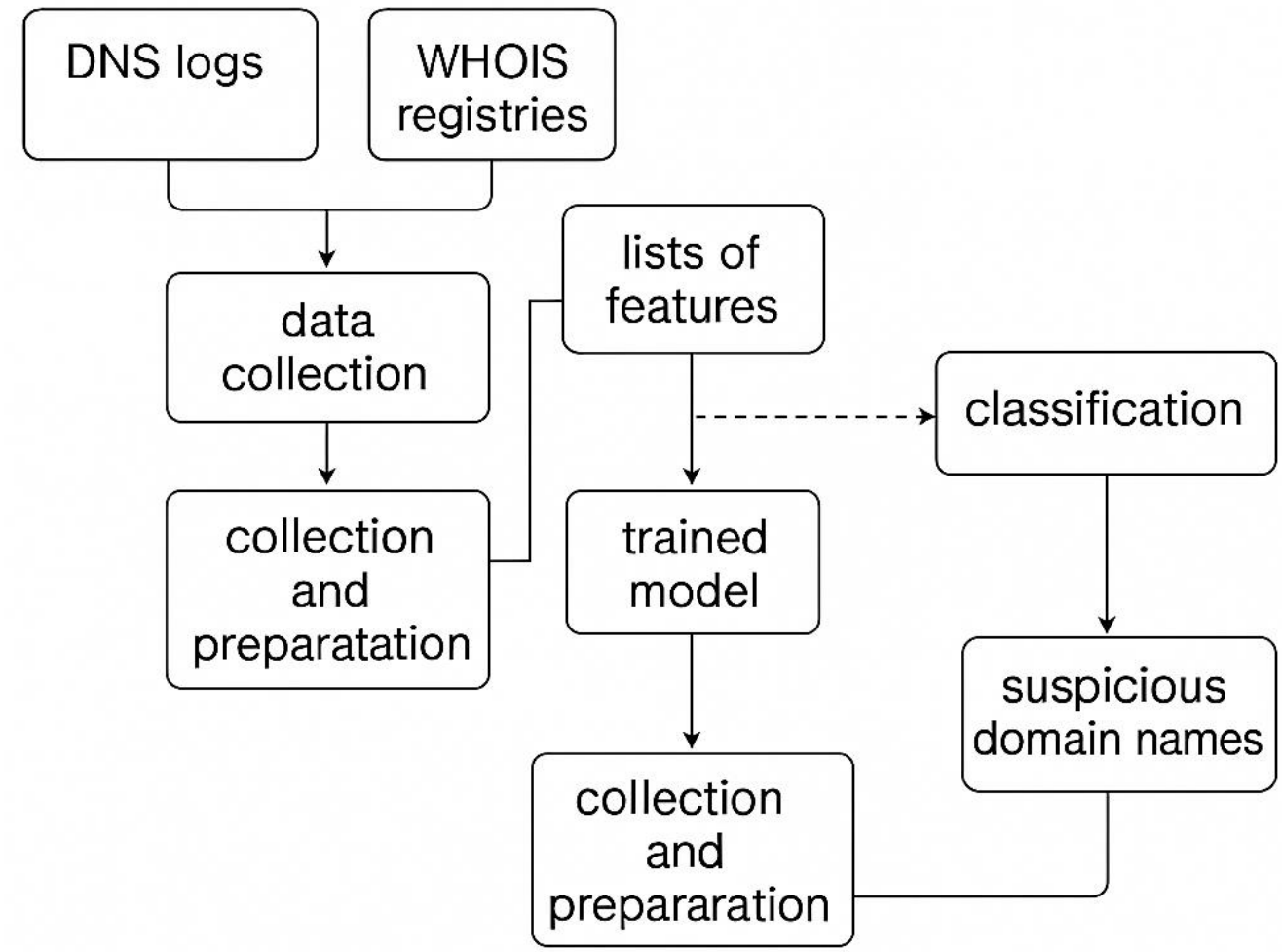

Рисунок 5.2 – Схема процесу класифікації (рисунок створено самостійно)

Таким чином, застосування алгоритмів машинного навчання у зв’язці з технологіями обробки великих даних дозволяє ефективно аналізувати потоки DNS-запитів, виявляти аномальні або потенційно небезпечні доменні імена та здійснювати їх класифікацію з високим рівнем точності.

5.5. Навчання моделі машинного навчання

З метою автоматизованого виявлення шкідливих доменних імен було застосовано модель машинного навчання на основі алгоритму Random Forest – ансамблевого методу, який створює сукупність дерев рішень та здійснює голосування для ухвалення остаточного рішення. Для навчання використовувався набір даних «Benign and malicious domains based on DNS logs», що містить дані про 90000 доменних імен, розподілених порівну на 50% безпечних та 50% шкідливих [17].

Попередньо дані були очищені, нормалізовані та закодовані: рядкові значення конвертовано у числові за допомогою LabelEncoder, логічні значення приведено до 0 та 1. Метод для попередньої обробки датасету:

```
df = pd.read_csv('BenignAndMaliciousDataset.csv')
# Перевірка наявності пропущених значень
missing_values = df.isnull().sum()
# Аналіз унікальних значень у стовпці "Class"
class_distribution = df['Class'].value_counts()
# Перетворення 'null' строк у NaN для аналізу
df.replace("null", np.nan, inplace=True)
# Спроба перевести всі булеві значення до 0/1
df = df.applymap(lambda x: 1 if str(x).strip().lower() == 'true' else
(0 if str(x).strip().lower() == 'false' else x))
# Конвертація числових значень
df = df.apply(pd.to_numeric, errors='ignore')
# Копія датафрейму
df_encoded = df.copy()
# Кодування всіх об'єктових колонок
label_encoders = {}
for col in df_encoded.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df_encoded[col].astype(str))
    label_encoders[col] = le
# Повторний поділ
X = df_encoded.drop(columns=['Class'])
y = df_encoded['Class']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
# Модель
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
# Збереження моделі
model_filename = "model.pkl"
joblib.dump(rf_classifier, model_filename)
```

Після обробки датасет містив 32 ознаки, що формували вхідний вектор для моделі. Розподіл на тренувальну та тестову вибірки здійснювався у пропорції 80:20 зі збереженням балансу класів.

Після навчання модель досягла точності на тестовій вибірці на рівні 99,9% (див. рис. 5.3), демонструючи високі значення precision, recall та F1-score (див. табл. 5.1). Така якість є задовільною для задач кібербезпеки, де важливо не лише виявити потенційно небезпечні домени, а й мінімізувати хибнопозитивні спрацьовування.

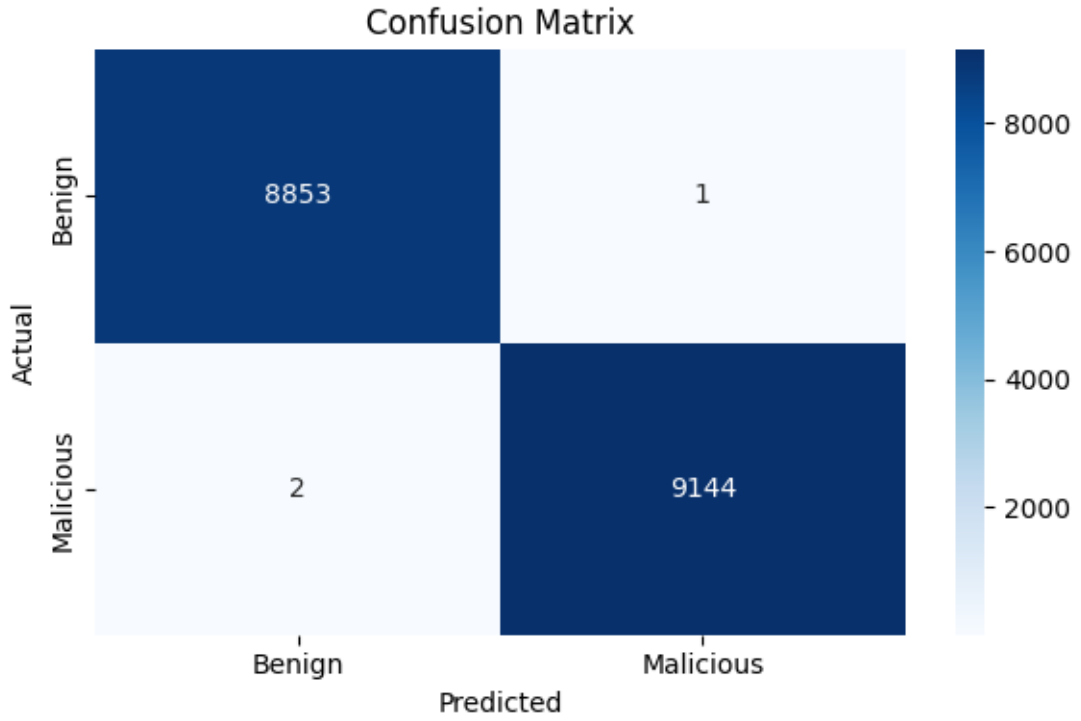

Рисунок 5.3 – Результат роботи отриманої моделі на тестовій вибірці (рисунок створено самостійно)

Таблиця 5.1 – Показники отриманої моделі (таблиця виконана самостійно)

	precision	recall	f1-score	support
0	0.999774	0.999887	0.999831	8854.000000
1	0.999891	0.999781	0.999836	9146.000000
accuracy	0.999833	0.999833	0.999833	0.999833
macro avg	0.999832	0.999834	0.999833	18000.000000
weighted avg	0.999833	0.999833	0.999833	18000.000000

Для зберігання моделі використовувався формат .pkl, що дозволяє легко інтегрувати її в розроблений прототип. Під час подальшої роботи система автоматично формує вектор ознак для нового домену відповідно до структури навчального датасету, завдяки чому забезпечується коректність і узгодженість передбачень.

5.6 Реалізація основних алгоритмів

У рамках розробленої системи реалізовано декілька ключових алгоритмів, які забезпечують обробку введеного доменного імені, витяг релевантних ознак та їх подальший аналіз за допомогою моделі машинного навчання. Основна мета реалізації – забезпечити достатній рівень автоматизації та точності для ефективного визначення потенційної шкідливості домену.

5.6.1 Витяг ознак з домену

Для кожного введеного доменного імені програма здійснює серію запитів до зовнішніх джерел:

- WHOIS-інформація: перевірка наявності домену, дати створення та закінчення реєстрації;
- DNS-записи: перевірка наявності записів A, MX, TXT, витяг SPF, DKIM, DMARC, що свідчать про налаштованість домену для легітимної діяльності;
- структурний аналіз: розрахунок кількості символів, довжини, кількості цифр, специфічних символів, а також співвідношення голосних та приголосних у доменному імені.

У разі неможливості зібрати реальні дані (наприклад, у випадку неіснуючого домену або відсутності DNS-відповіді), система все одно проводить оцінку, базуючись лише на структурних характеристиках, що гарантує універсальність підходу.

Основна логіка реалізована за допомогою методів extract_name_features, fetch_whois_info та fetch_dns_info:

```
    def extract_name_features(domain):
        hyphen_count = domain.count('-')
        dot_count = domain.count('.')
        domain_length = len(domain)

        vowels = 'aeiou'
        consonants = 'bcdfghjklmnpqrstvwxyz'
        letters = re.sub(r'[^a-zA-Z]', '', domain.lower())

        vowel_count = sum(1 for c in letters if c in vowels)
        consonant_count = sum(1 for c in letters if c in consonants)
        numeric_count = sum(1 for c in domain if c.isdigit())
        special_char_count = len(re.findall(r'[^a-zA-Z0-9.-]', domain))

        total_chars = len(domain)
        vowel_ratio = vowel_count / total_chars
        consonant_ratio = consonant_count / total_chars
        numeric_ratio = numeric_count / total_chars
        special_char_ratio = special_char_count / total_chars

        return {
            'HyphenCount': hyphen_count,
            'DotCount': dot_count,
            'DomainLength': domain_length,
            'VowelRatio': round(vowel_ratio, 3),
            'ConsoantRatio': round(consonant_ratio, 3),
            'NumericRatio': round(numeric_ratio, 3),
            'SpecialCharRatio': round(special_char_ratio, 3)
        }


    def fetch_whois_info(domain):
        try:
            w = whois.whois(domain)
            creation_date = w.creation_date
            if isinstance(creation_date, list):
                creation_date = creation_date[0]
            if creation_date:
                age_days = (datetime.utcnow() - creation_date).days
                return {'HasWhois': 1, 'CreationDate': age_days}
            # Some registrars hide creation date but still return a
domain name
            if w.domain_name:
                return {'HasWhois': 1, 'CreationDate': 0}
        except Exception:
            pass

        # Fallback to HTTP API if direct WHOIS lookup failed
        try:
            resp = requests.get(f'https://api.whois.vu/?q={domain}',
timeout=10)
            if resp.status_code == 200:
                data = resp.json()
                available = str(data.get('available', '')).lower()
                if available in ('yes', 'true', '1'):
                    return {'HasWhois': 0, 'CreationDate': 0}
```

```
                created_ts = data.get('created') or
data.get('created_date')
                if created_ts:
                    try:
                        ts = int(created_ts)
                        creation_date = datetime.utcfromtimestamp(ts)
                        age_days = (datetime.utcnow() -
creation_date).days
                        return {'HasWhois': 1, 'CreationDate': age_days}
                    except Exception:
                        pass

                whois_text = data.get('whois', '')
                match = re.search(r"Creation Date:\s*(\d{4}-\d{2}-
\d{2})", whois_text)
                if match:
                    try:
                        creation_date = datetime.strptime(match.group(1),
"%Y-%m-%d")
                        age_days = (datetime.utcnow() -
creation_date).days
                        return {'HasWhois': 1, 'CreationDate': age_days}
                    except Exception:
                        pass
                return {'HasWhois': 1, 'CreationDate': 0}
            except Exception:
                pass

            # Additional fallback using RDAP protocol
            try:
                resp = requests.get(f'https://rdap.org/domain/{domain}',
timeout=5)
                if resp.status_code == 404:
                    return {'HasWhois': 0, 'CreationDate': 0}
                if resp.status_code == 200:
                    data = resp.json()
                    events = data.get('events', [])
                    for event in events:
                        if event.get('eventAction') in ('registration',
'registered'):
                            date_str = event.get('eventDate')
                            if date_str:
                                try:
                                    creation_date =
datetime.strptime(date_str[:10], '%Y-%m-%d')
                                    age_days = (datetime.utcnow() -
creation_date).days
                                    return {'HasWhois': 1, 'CreationDate':
age_days}
                                except Exception:
                                    break
                return {'HasWhois': 1, 'CreationDate': 0}
            except Exception:
                pass

            # Last resort: if DNS responds, assume the domain exists
            try:
                dns.resolver.resolve(domain, 'NS')
```

```
            return {'HasWhois': 1, 'CreationDate': 0}
        except Exception:
            pass

        # If all methods fail, treat as not having WHOIS info
        return {'HasWhois': 0, 'CreationDate': 0}


    def fetch_dns_info(domain):
        result = {'HasMX': 0, 'HasSPFInfo': 0, 'HasDkimInfo': 0,
'MXDnsResponse': 0, 'TXTDnsResponse': 0}
        try:
            answers = dns.resolver.resolve(domain, 'MX')
            if answers:
                result['HasMX'] = 1
                result['MXDnsResponse'] = 1
        except:
            pass

        try:
            answers = dns.resolver.resolve(domain, 'TXT')
            if answers:
                result['TXTDnsResponse'] = 1
                for rdata in answers:
                    txt = str(rdata.strings[0], 'utf-8') if
hasattr(rdata, 'strings') else str(rdata)
                    if "spf" in txt.lower():
                        result['HasSPFInfo'] = 1
                    if "dkim" in txt.lower():
                        result['HasDkimInfo'] = 1
        except:
            pass

        return result
```

Відсутні або неможливі до отримання значення замінюються на нулі або умовні значення (наприклад, якщо WHOIS не знайдено).

Кількісні ознаки (довжина, співвідношення символів тощо) обчислюються програмно без зовнішніх запитів.

Таким чином, реалізована логіка дозволяє проводити комплексну оцінку доменів у автоматичному режимі з гнучким підходом до неповних або обмежених даних.

5.6.2 Попередня обробка та узгодження ознак

Після збору ознак дані нормалізуються до узгодженого з навчальним набором формату за допомогою методу align_features_to_model:

```
    def align_features_to_model(current_features: dict) -> pd.DataFrame:
        aligned = {}
        for col in expected_features:
            if col in current_features:
                aligned[col] = current_features[col]
            else:
                aligned[col] = 0 if isinstance(current_features.get(col,
0), (int, float)) else ''
        return pd.DataFrame([aligned])
```

Таким чином створюється датафрейм з такою самою структурою, як і в оригінальному датасеті для тренування.

5.6.3 Прогнозування та інтерпретація результату

За допомогою моделі RandomForestClassifier, натренованій на основі реального датасету, що містить мітки класу (benign або malicious), після визначення необхідних для її роботи даних, відбувається класифікація домену як шкідливого або безпечного методом analyze_domain:

```
    def analyze_domain(domain, model):
        name_features = extract_name_features(domain)
        whois_features = fetch_whois_info(domain)
        dns_features = fetch_dns_info(domain)

        all_features = {**whois_features, **dns_features,
**name_features}
        df_input = align_features_to_model(all_features)

        prediction = model.predict(df_input)[0]
        confidence = model.predict_proba(df_input)[0][prediction]

        class_str = "Malicious" if prediction == 1 else "Benign"
        print(f"Domain: {domain}")
        print(f"Status: {class_str}")
        print(f"Confidence: {confidence:.2f}")
        print(f"Details:")
        for k, v in all_features.items():
            print(f"  {k}: {v}")

        return {
            'domain': domain,
            'class': int(prediction),
            'confidence': float(round(confidence, 2)),
            'details': all_features
        }
```

Програма виконує прогноз на основі отриманих ознак і повертає бінарний результат (0 або 1), що відносить домен до безпечних або шкідливих, рівень впевненості в цьому результаті, а також пояснення основних причин класифікації у вигляді списку логічних висновків.

5.7 Візуалізація та інтерфейс користувача

Візуалізація та інтерфейс користувача відіграють важливу роль, оскільки саме візуальний інтерфейс є ключовим способом комунікації між користувачем і системою. З огляду на те, що система орієнтована на обробку та аналіз великих обсягів даних, одним із пріоритетів при розробці стало створення інтуїтивно зрозумілого, адаптивного та функціонального UI/UX, що забезпечує ефективну взаємодію з аналітикою результатів.

Інтерфейс було реалізовано як односторінковий веб-додаток (SPA) з використанням фреймворку React. Цей вибір зумовлений його популярністю, швидкістю роботи, активною спільнотою та сумісністю з сучасними бібліотеками для візуалізації, такими як Recharts та D3.js. Для стилізації інтерфейсу було використано Tailwind CSS, що дозволяє швидко та гнучко будувати адаптивний дизайн.

Архітектурно візуальний інтерфейс побудовано як окремий фронтенд-шар, який взаємодіє з бекенд-модулем, реалізованим на основі FastAPI. Користувач вводить домен у веб-форму, після чого дані передаються на сервер, де виконуються всі етапи аналізу, включаючи запити WHOIS/DNS, побудову вектору ознак та прогнозування класу.

5.7.1 Головна сторінка

На головній сторінці (див. рис. 5.4) відображається форма для пошуку доменного імені, а також зведена інформація про поточні загрози, а саме: кількість доменів, що були перевірені, кількість виявлених активних загроз, поточний рівень ризику та зміну тренду за відсотком шкідливих доменів.

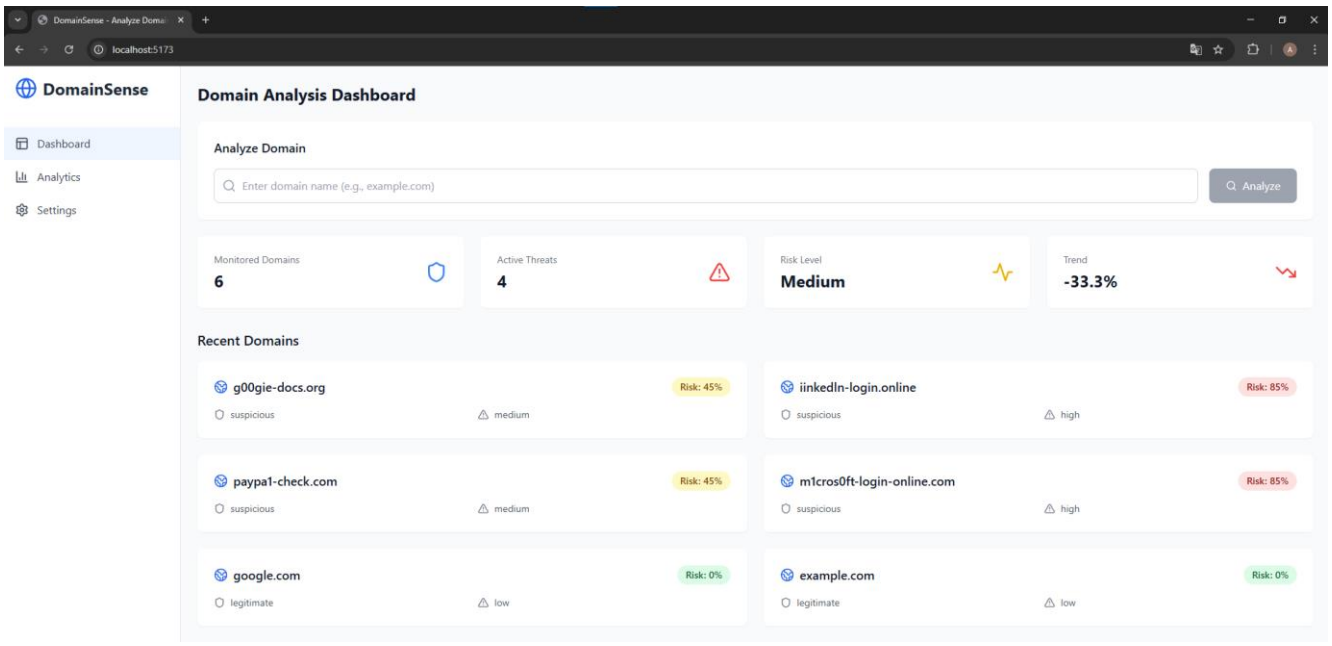

Рисунок 5.4 – Головна сторінка (рисунок створено самостійно)

Під зведеною інформацією наведений список нещодавно проаналізованих доменів, кожен з яких містить статус та ризик-індикатор. При натисканні на картку домена відбувається перехід до сторінки з детальною інформацією.

5.7.2 Сторінка деталізації доменного імені

Ця сторінка (див. рис. 5.5, 5.6) дозволяє переглянути детальну інформацію щодо окремого домена:

– registration details: дата створення та закінчення реєстрації, реєстратор;
– location information: країна розташування за IP;
– security status: прогноз моделі (malicious або benign), поточний статус (suspicious або legitimate), рівень загрози, дата останнього сканування;
– technical details: імена DNS-серверів, IP-адреси;
– DNS records: A, MX, TXT-записи;
– risk factors: перелік виявлених проблем і факторів ризику або повідомлення про те, що їх немає.

Ця сторінка особливо важлива для аналітиків кіберзагроз, адже дозволяє швидко переглянути не лише підсумковий висновок, але й причини його формування, а також основну технічну інформацію щодо конкретного домену.

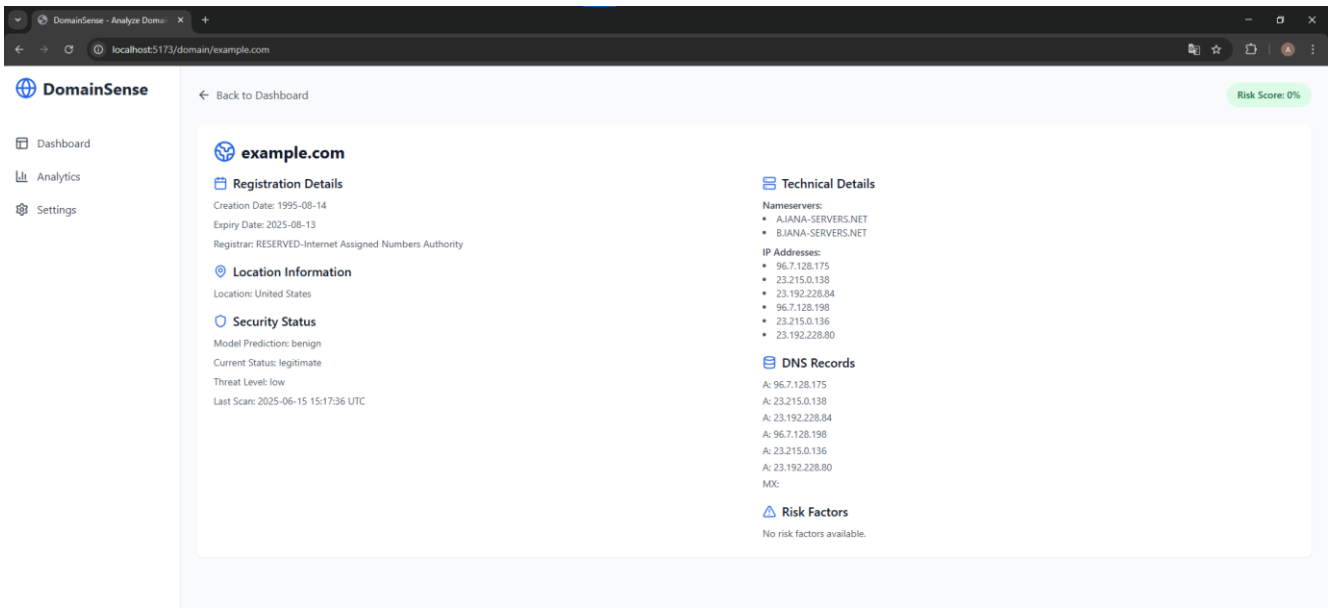

Рисунок 5.5 – Сторінка деталізації для безпечного домена (рисунок створено самостійно)

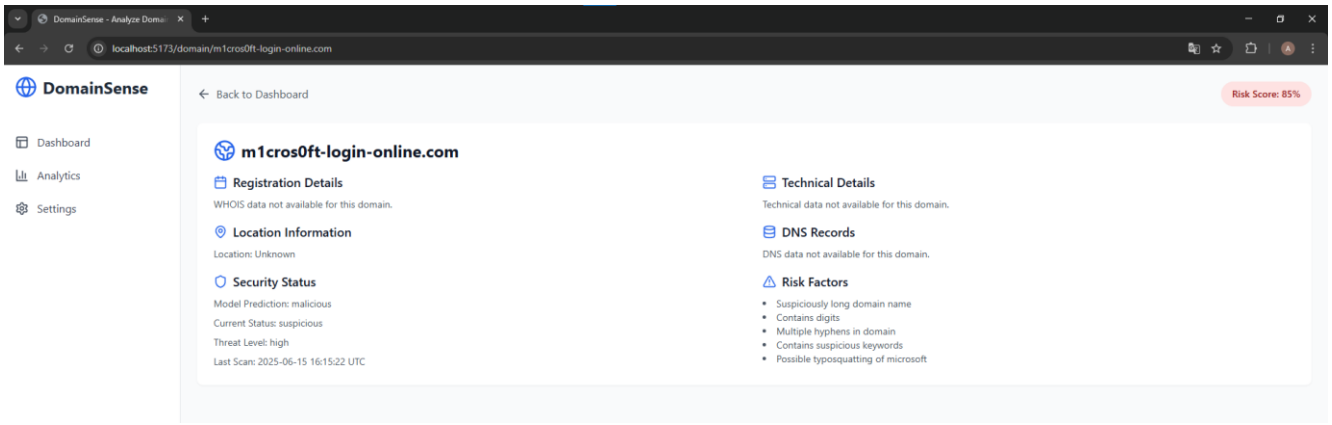

Рисунок 5.6 – Сторінка деталізації для зловмисного домена (рисунок створено самостійно)

5.7.3 Сторінка аналітики

Аналітика (див. рис. 5.7) включає інтерактивні графіки, що дозволяють аналізувати динаміку підозрілих доменів у часі, порівнювати категорії загроз, а також формувати звіти. Розділ аналітики призначено для візуалізації:

– threat distribution: розподіл за класами (benign/malicious);
– trend analysis: відстеження змін у динаміці доменної активності;
– category distribution: у перспективі може відображати типи загроз або групи доменів (наприклад, фішинг, спам, тощо).

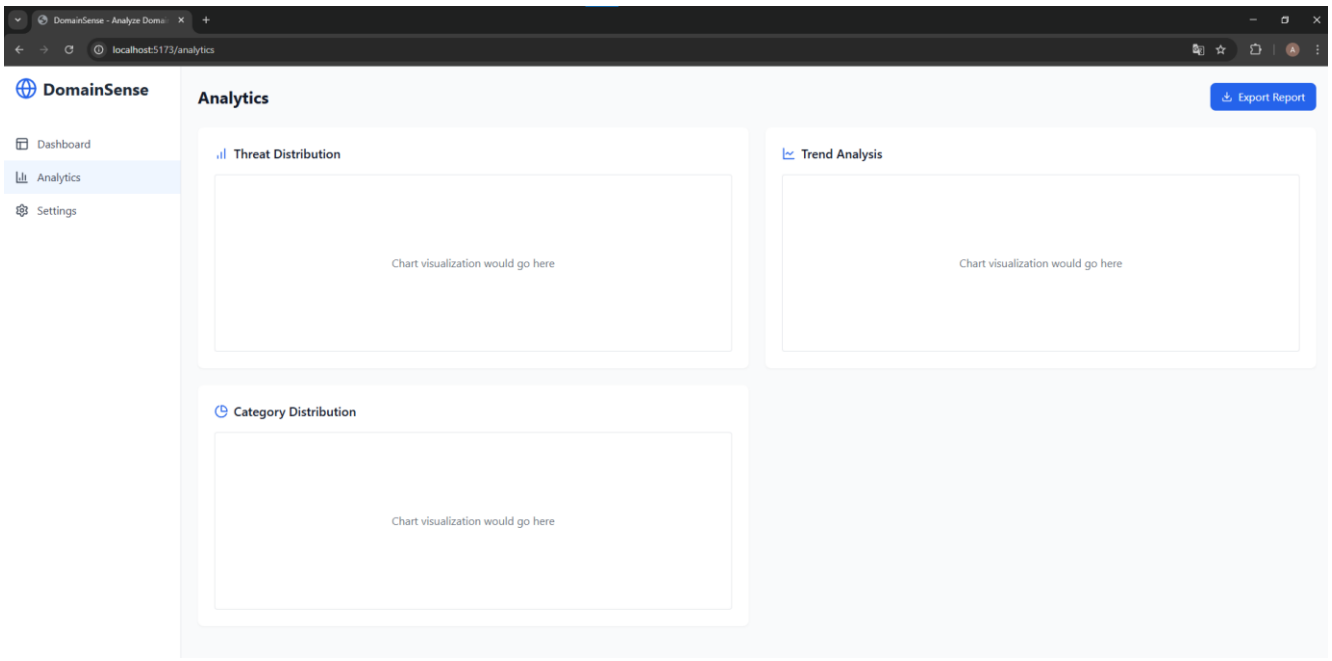

Рисунок 5.7 – Сторінка аналітики (рисунок створено самостійно)

5.7.4 Сторінка налаштувань

Ця сторінка (див. рис. 5.8) дозволяє налаштувати мову, часовий пояс, включити email-сповіщення для доменів із високим рівнем ризику, а також визначити поріг ризику, при перевищенні якого домени вважаються підозрілими.

Таким чином, інтерфейс користувача не лише підтримує зручну взаємодію з даними, але й сприяє ефективному виявленню загроз і прийняттю рішень на основі візуального аналізу.

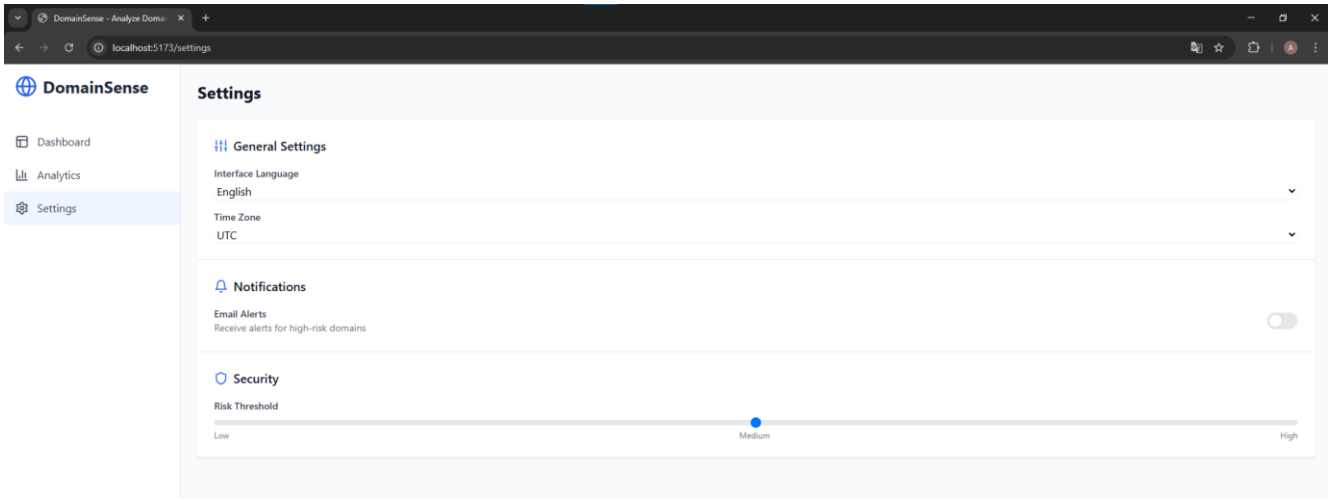

Рисунок 5.8 – Сторінка налаштувань (рисунок створено самостійно)

5.8 Висновки з практичного дослідження

Під час практичного дослідження було детально охарактеризовано ключові аспекти розробки та реалізації системи для аналізу доменних імен та виявлення кіберзагроз. Оцінка вимог до системи показала важливість чіткого визначення функціональних та нефункціональних характеристик, що забезпечують ефективність, масштабованість та зручність у використанні для кінцевих користувачів, таких як аналітики з кібербезпеки.

У результаті було створено повнофункціональний прототип системи, що виконує автоматизований аналіз доменних імен на предмет потенційної шкідливості. Основною перевагою реалізованого рішення є здатність обробляти доменні імена в режимі реального часу та надавати детальний аналіз на основі як структурних характеристик домена, так і DNS/WHOIS-атрибутів.

Програма є універсальною і здатна працювати не лише зі штучно згенерованими даними, а й з реальними доменами, що робить її придатною до розширення у вигляді веб-сервісу або інструменту для кібербезпекових команд. Обрана архітектура з мікросервісами забезпечила гнучкість та масштабованість системи, а також можливість інтеграції з іншими сервісами через API.

Також було реалізовано інтерфейс користувача, який відіграє важливу роль у забезпеченні зручної взаємодії з системою. Завдяки використанню сучасних фреймворків та бібліотек, візуалізація результатів стала доступною та зрозумілою для кінцевих користувачів. Інтерфейс забезпечує зручний перегляд даних, фільтрацію, порівняння результатів та експорт інформації для подальшого аналізу.

Таким чином, реалізований інтерфейс забезпечує простоту використання та можливість оперативного отримання результатів, що є важливим фактором для систем, що працюють у сфері кібербезпеки та моніторингу доменів.

Загалом, практичне дослідження продемонструвало успішне застосування технологій Big Data та машинного навчання для вирішення актуальних задач у галузі кібербезпеки. Система забезпечує не лише ефективний аналіз доменних імен, але і створює основу для подальшого розвитку та вдосконалення, включаючи інтеграцію нових джерел даних та алгоритмів для покращення точності прогнозів.

Представлений інтерфейс поєднує простоту використання із достатньою глибиною технічної інформації. Такий підхід дозволяє застосовувати систему як в автоматизованих системах моніторингу, так і в ручному режимі для оперативного аналізу.

Повний програмний код реалізованої системи, специфікація програмного продукту, а також відеоролик з демонстрацією роботи програми доступні у відкритому доступі на репозиторії GitHub [18].

За результатами теоретичного та практичного досліджень була виконана апробація результатів роботи у вигляді участі у Міжнародній науково-практичній конференції «Сучасні інформаційні технології та системи штучного інтелекту MIT@AIS-2025» [19].

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проведено дослідження, спрямоване на розробку інструменту прогнозування порушень умов використання доменних імен. Було досліджено сучасні тенденції та проблеми у сфері аналізу доменних імен, виявлено основні виклики та обмеження існуючих підходів, а також підкреслено важливість інтеграції технологій Big Data та машинного навчання для розв’язання поставлених задач.

Проведено систематизований огляд наукової літератури та джерел, пов’язаних із аналізом доменів, застосуванням машинного навчання і технологій великих даних, що дозволило обґрунтувати вибір методів і технологій для розробки інструменту.

Визначено основну мету дослідження, яка полягала у створенні ефективного та масштабованого інструменту, що інтегрує різнорідні дані для прогнозування порушень у використанні доменних імен. Розроблено перелік підзадач, включаючи вибір методів аналізу даних, проектування структури зберігання та створення інтерфейсу.

У результаті роботи було реалізовано повноцінний прототип програмного засобу для виявлення потенційно небезпечних або зловмисних доменних імен з використанням технологій Big Data та методів машинного навчання. Була виконана глибока аналітична обробка датасету, що містить реальні приклади як легітимних, так і зловмисних доменів. Проведено попередню обробку, перетворення даних та генерацію ознак, включаючи структурні характеристики, DNS-записи, наявність WHOIS-інформації, реєстраційні атрибути та інші технічні індикатори. Для моделювання застосовано класифікатор Random Forest, який показав високу точність при розпізнаванні потенційно зловмисних доменів. Модель була натренована, збережена та інтегрована в загальну систему.

У системі реалізовано механізми збору даних про домен у реальному часі: витяг WHOIS-даних, DNS-записів (A, MX, TXT), а також обробка самого доменного імені для визначення підозрілих патернів. У разі відсутності технічної інформації аналіз виконується на основі доступних структурних ознак.

Користувацький інтерфейс, розроблений для взаємодії з системою, дозволяє візуалізувати результати у зручному форматі. Це сприяє прозорості прийняття рішень та можливості подальшої перевірки вручну.

Реалізований прототип підтвердив можливість ефективного поєднання великих даних, алгоритмів машинного навчання та автоматизованої обробки інформації для виявлення порушень у використанні доменних імен. Система є придатною для подальшого розширення, як за рахунок підключення додаткових джерел, так і через удосконалення моделі шляхом впровадження глибокого навчання та методів оптимального управління у сценаріях конфліктної поведінки в мережі.

Таким чином, результати роботи відповідають поставленим задачам, а розроблений підхід сприяє зміцненню кібербезпеки та підвищенню ефективності моніторингу доменних імен.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Cybersecurity in the digital age: developing robust strategies to protect against evolving global digital threats and cyber attacks / T. Sendjaja et al. *International journal of science and society*. 2024. Vol. 6, no. 1. P. 1008–1019. URL: https://doi.org/10.54783/ijsoc.v6i1.1098.

2. Scam Sniffer Report: *$71 Million Stolen Due To Phishing In March.* URL: https://drops.scamsniffer.io/71-million-stolen-due-to-phishing-in-march/ (дата звернення: 18.04.2025).

3. Janeja, Vandana P. *Data Analytics for Cybersecurity*. Cambridge: Cambridge University Press, 2022. URL: https://doi.org/10.1017/9781108231954.

4. Detecting internet abuse by analyzing passive DNS traffic: a survey of implemented systems / S. Torabi et al. *IEEE communications surveys & tutorials*. 2018. Vol. 20, no. 4. P. 3389–3415. URL: https://doi.org/10.1109/comst.2018.2849614.

5. Thapliyal V., Thapliyal P. Machine learning for cybersecurity: threat detection, prevention, and response. *Darpan international research analysis*. 2024. Vol. 12, no. 1. P. 1–7. URL: https://doi.org/10.36676/dira.v12.i1.01.

6. An evaluation of machine learning methods for domain name classification / A. Garg et al. *2020 IEEE international conference on big data (big data)*, Atlanta, GA, USA, 10–13 December 2020. 2020.
URL: https://doi.org/10.1109/bigdata50022.2020.9377787.

7. Лазар М. Технології big data у аналізі ризиків страхової компанії / М. Лазар, В. Кобзєв // Інформаційні системи та технології : матеріали статей 7-ї Міжнародної науково-технічної конференції, Коблеве-Харків, 10-15 вересня 2018 р. – Харків : ХНУРЕ, 2018. – С. 364–367.

8. Anti-Phishing Working Group (APWG). *Global Phishing Survey 2024*. URL: https://apwg.org/ (дата звернення: 21.04.2025).

9. Bumanglag K., Kettani H. On the impact of DNS over HTTPS paradigm on cyber systems. *2020 3rd international conference on information and computer technologies (ICICT)*, San Jose, CA, USA, 9–12 March 2020. 2020. URL: https://doi.org/10.1109/icict50521.2020.00085.

10. SANS Institute. *Botnet Threats and Countermeasures*. URL: https://www.sans.org/white-papers/ (дата звернення: 22.04.2025).

11. Kim T. H., Reeves D. A survey of domain name system vulnerabilities and attacks. *Journal of surveillance, security and safety*. 2020. URL: https://doi.org/10.20517/jsss.2020.14.

12. White T. Hadoop: the definitive guide. O'Reilly, 2015. 727 с.

13. Buyya R., Calheiros R. N., Dastjerdi A. V. Big data: principles and paradigms. Elsevier Science & Technology Books, 2016. 494 с.

14. I. H. Witten. Data mining: practical machine learning tools and techniques / I. H. Witten та ін. Elsevier Science & Technology Books, 2016. 654 с.

15. Mazurova O., Naboka A., Shirokopetleva M. Research of acid transaction implementation methods for distributed databases using replication technology. *Innovative technologies and scientific solutions for industries*. 2021. No. 2 (16). P. 19–31. URL: https://doi.org/10.30837/itssi.2021.16.019.

16. Mazurova O., Syvolovskyi I., Syvolovska O. Nosql database logic design methods for mongodb and neo4j. *Innovative technologies and scientific solutions for industries*. 2022. No. 2(20). P. 52–63. URL: https://doi.org/10.30837/itssi.2022.20.052.

17. Mendeley Data. *Benign and malicious domains based on DNS logs*. URL: https://data.mendeley.com/datasets/623sshkdrz/5 (дата звернення: 12.05.2025).

18. GitHub – 2025_M_PI_IPZ-23-4_Ushakov_A_M. GitHub. URL: https://github.com/Andrii-Ushakov/2025_M_PI_IPZ-23-4_Ushakov_A_M (дата звернення: 15.06.2025).

19. Andrii Ushakov and Anatolii Rutkas. Leveraging Big Data Technologies for Domain Name Security, у *Modern Information Technologies and Artificial Intelligence Systems: Proceedings of the 1st International Scientific and Practical Conference. Part 1*, Kharkiv–Yaremche, May 19–22, 2025, ed. Yu.O. Romanenkov et al. – Kharkiv: NURE, 2025. – P. 202–205.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

7. Лазар М. Технології big data у аналізі ризиків страхової компанії / М. Лазар, В. Кобзєв // Інформаційні системи та технології : матеріали статей 7-ї Міжнародної науково-технічної конференції, Коблеве-Харків, 10-15 вересня 2018 р. – Харків : ХНУРЕ, 2018. – С. 364–367.

15. Mazurova O., Naboka A., Shirokopetleva M. Research of acid transaction implementation methods for distributed databases using replication technology. *Innovative technologies and scientific solutions for industries*. 2021. No. 2 (16). P. 19–31. URL: https://doi.org/10.30837/itssi.2021.16.019.

16. Mazurova O., Syvolovskyi I., Syvolovska O. Nosql database logic design methods for mongodb and neo4j. *Innovative technologies and scientific solutions for industries*. 2022. No. 2(20). P. 52–63. URL: https://doi.org/10.30837/itssi.2022.20.052.

19. Andrii Ushakov and Anatolii Rutkas. Leveraging Big Data Technologies for Domain Name Security, y *Modern Information Technologies and Artificial Intelligence Systems: Proceedings of the 1st International Scientific and Practical Conference. Part 1*, Kharkiv–Yaremche, May 19–22, 2025, ed. Yu.O. Romanenkov et al. – Kharkiv: NURE, 2025. – P. 202–205.