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)

Дослідження методів розпізнавання зловмисного зашифрованого трафіку для захисту хмарних систем від DDoS атак. Використання глибинного навчання з підкріпленням
(тема)

Виконав:
здобувач 2 року навчання
групи ІПЗм-23-4
Андрій ВЕЛИКОРОДНІЙ
(власне ім'я, прізвище)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва освітньої програми)

Керівник доц. Наталя КРАВЕЦЬ
(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
(код і повна назва)

Тип програми освітньо-наукова програма
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Великородньому Андрію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів розпізнавання зловмисного зашифрованого трафіку для захисту хмарних систем від DDoS атак. Використання глибинного навчання з підкріпленням».
   затверджена наказом університету від 15 квітня 2025 р. № 290 Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 24 червня 2025 р.
3. Вихідні дані до роботи «*науково-технічні публікації, матеріали конференцій, відкриті інтернет-джерела щодо методів аналізу зашифрованого трафіку, моделювання DDoS-атак, алгоритмів глибинного навчання з підкріпленням, публічні датасети зашифрованого мережевого трафіку».*

Перелік питань, що потрібно опрацювати у роботі «*мета роботи, аналіз предметної галузі та постановка задачі, огляд сучасних методів виявлення зашифрованого DDoS-трафіку, аналіз можливостей застосування Reinforcement Learning, порівняння з класичними підходами, підбір релевантних датасетів, розробка теоретичної архітектури системи, створення та навчання RL-моделі, побудова MVP-прототипу та оцінка ефективності запропонованого рішення.».*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	16.04. 25	*виконано*
2	Аналіз предметної галузі	21.04.25 – 25.04.25	*виконано*
3	Огляд й аналіз літературних, наукових джерел	25.04.25 – 29.04.25	*виконано*
4	Теоретичне дослідження	30.04.25 – 15.05.25	*виконано*
5	Архітектура та проектування системи	15.05.25 – 30.05.25	*виконано*
6	Опис експерементальних досліджень	01.06.25 – 06.06.25	*виконано*
7	Підготовка до апробації результатів дослідження. Публікація матеріалів	06.06.25 – 08.06.25	*виконано*
8	Підготовка пояснювальної записки	08.06.25 – 16.06.25	*виконано*
9	Підготовка презентації та доповіді	16.06.25 – 17.06.25	*виконано*
10	Перевірка на плагіат	16.06.25	*виконано*
11	Нормоконтроль	19.06.25	*виконано*
12	Рецензування	19.06.25 – 21.06.25	*виконано*
13	Попередній захист	23.06.25	*виконано*
14	Занесення диплома в електронний архів	23.06.25	*виконано*
15	Допуск до захисту у зав. кафедри	23.06.25	*виконано*

Дата видачі завдання 15 квітня 2025р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Наталя КРАВЕЦЬ.
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ / ABSTRACT

Робота містить: 93 с., 20 рис., 1 таблиця, 12 джерел, 4 додатки.

ГЛИБИННЕ НАВЧАННЯ, DDoS-АТАКИ, МЕТАДАНІ, МЕРЕЖЕВА БЕЗПЕКА, НАВЧАННЯ З ПІДКРІПЛЕННЯМ, ЗАШИФРОВАНИЙ ТРАФІК, CYBERSECURITY, DETECTION SYSTEM, REINFORCEMENT LEARNING, RL-AGENT.

Об’єктом дослідження є процес виявлення зловмисного зашифрованого трафіку, що використовується для здійснення DDoS-атак на хмарні та локальні інфраструктури.

Метою роботи є дослідження можливості застосування глибинного навчання з підкріпленням (Reinforcement Learning) для побудови системи автоматичного розпізнавання DDoS-атак у зашифрованому трафіку на основі аналізу лише метаданих, без розшифрування вмісту мережевих пакетів.

Методами дослідження є аналіз літературних джерел, порівняльний аналіз класичних та сучасних методів виявлення DDoS у TLS/HTTPS-трафіку, моделювання процесу виявлення через глибокі Q-мережі (DQN, DDQN).

У роботі запропоновано архітектуру агентної системи на основі Deep Q-Learning, що навчається в умовах симульованого середовища, де кожне рішення оцінюється функцією винагороди. Агент працює на рівні метаданих (кількість байт, інтервали між пакетами, частота сесій тощо) та здатен виявляти як високочастотні, так і повільні DDoS-атаки в зашифрованих потоках.

Реалізовано прототип MVP, який може функціонувати як локально, так і у хмарному середовищі (AWS). Навчання та тестування підтвердили доцільність використання RL у задачах кіберзахисту: модель демонструє адаптивність до нових атак і високу точність (>98%) при низькому рівні хибних спрацювань. Підкреслено переваги RL-методів над класичними підходами в умовах динамічних загроз.

Практичне значення роботи полягає у створенні концепції та реалізації адаптивного RL-агента, здатного автономно виявляти DDoS-атаки в зашифрованому трафіку без потреби в глибокій інспекції. Результати можуть бути

використані в системах захисту хмарних сервісів та інтегровані у SIEM-рішення або мережеву периферію.

CYBERSECURITY, DETECTION SYSTEM, ENCRYPTED TRAFFIC, DDoS ATTACKS, DEEP LEARNING, METADATA, NETWORK SECURITY, REINFORCEMENT LEARNING, RL-AGENT, TRAFFIC CLASSIFICATION.

The object of research is the process of detecting malicious encrypted traffic used to conduct DDoS attacks targeting cloud-based and local infrastructures.

The aim of this work is to explore the feasibility of using reinforcement learning-based deep neural models for the automatic recognition of DDoS attacks in encrypted traffic using only metadata, without decrypting the packet payload.

The applied methods include literature review, comparative analysis of classical and modern approaches to encrypted DDoS detection, simulation of detection environments through deep Q-networks (DQN, DDQN).

This research proposes a detection system architecture based on Deep Q-Learning, where the agent learns through simulated interactions and is rewarded based on the accuracy of decisions. The model operates on metadata (e.g., packet size, inter-arrival times, session frequency) and can detect both high-volume and low-rate DDoS attacks within encrypted flows.

An MVP prototype was implemented with deployment capability both locally and in the cloud (AWS). Training and testing confirmed the practicality of the RL approach for cybersecurity tasks: the model demonstrates adaptability to evolving attack patterns and achieves high accuracy (>98%) with a low false-positive rate. The study highlights the benefits of reinforcement learning over classical detection in dynamic threat environments.

The practical significance lies in developing an adaptive RL-agent capable of autonomously detecting DDoS attacks in encrypted traffic without deep packet inspection. The solution can be integrated into cloud service protection systems or modern SIEM architecture.

Заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу ElArKhNURE.

Завідувачу кафедри
ПІ_____
(скорочена назва кафедри)
проф. Кирилу СМЕЛЯКОВУ_
(вчене звання, сласне 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, _____Великородній Андрій Володомирович_____
(прізвище, ім'я, по батькові)

здобувач вищої освіти на другому (магістерському) рівні вищої освіти академічної групи ___ІПЗм-23-4_____
кафедра_____ програмної інженерії _____,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему _Дослідження методів розпізнавання зловмисного зашифрованого трафіку для захисту хмарних систем від DDoS атак. Використання глибинного навчання з підкріпленням_____
(назва роботи)

що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в репозиторії "ElArKhNURE". Погоджуюся з авторським договором, відповідно до Положення про репозиторій ХНУРЕ "ElArKhNURE". Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з вимогами академічної доброчесності, згідно з якими виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата 22.06.2025 Підпис 

ЗМІСТ

Вступ 10
1 Аналіз предметної галузі 12
1.1 Огляд предметної галузі 12
1.2 Огляд сучасних підходів до виявлення DDoS-атак 13
1.3 Використання Reinforcement learning у виявленні DDoS-атак 15
1.4 Тенденції та виклики у сфері виявлення DDoS-атак у зашифрованому трафіку 17
1.5 Постановка задачі 19
2 Огляд й аналіз літературних, наукових джерел 21
2.1 Огляд основних джерел 21
2.1.1 Загальні аспекти розпізнавання DDoS-атак у зашифрованому трафіку 22
2.1.2 Методи розпізнавання DDoS-атак по зашифрованому трафіку 23
2.1.3 Використання машинного навчання у прогнозуванні загроз 25
2.2 Аналіз літератури 26
2.2.1 Основні теорії та концепції 27
2.2.2 Моделі та методи аналізу 28
2.2.3 Ефективність існуючих підходів 29
2.3 Оцінка актуальності та новизни 30
2.4 Висновки з огляду 30
3 Теоретичне дослідження 32
3.1 Архітектура та проектування ПЗ 32
3.1.1 Загальна структура архітектури 32
3.1.2 Архітектурний стиль 34
3.1.3 Візуалізація 34
3.2 Проектування структури зберігання даних 36
3.2.1 Вибір технологій 37
3.2.2 Схема бази даних 38
3.2.3 Резервування та масштабованість 39

3.2.4 Інтеграція даних 39
3.3 Алгоритми та методи 40
3.4 Інші елементи, важливі для реалізації проєкту 42
3.4.1 Інтеграція з існуючими системами 42
3.4.2 Масштабованість 43
3.4.3 Безпека 43
3.4.4 Моніторинг і підтримка 44
3.4.5 Документація 44
3.5 Висновки з теоретичного дослідження 45
4 Архітектура та проектування системи 47
4.1 Вимоги до системи 47
4.1.1 Функціональні вимоги 47
4.1.2 Нефункціональні вимоги 48
4.1.3 Вхідні дані 48
4.1.4 Вихідні дані: 49
4.2 Вибір технологій та середовища розробки 50
4.3 Архітектура та структура системи 51
4.4 Алгоритми та методи виявлення DDoS-атак 53
4.5 Візуалізація та інтерфейс користувача 55
4.5.1 Головна сторінка 56
4.5.2 Екран деталізації потоку 58
4.6 Висновки з практичного дослідження 59
5 Опис експерементальних дослІджень 61
5.1 Умови експерименту 61
5.2 Евристичний метод – SYN Ratio 61
5.3 Евристичний метод – IAT Regularity 64
5.4 Евристичний метод – Unidirectionality 67
5.5 Модель Double DQN 69
5.6 Аналіз Результатів 72
Висновки 76

Перелік джерел посилання 77

Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії 79

ДОДАТОК А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ ... 80

ДОДАТОК Б Слайди презентації 82

ДОДАТОК В Апробація результатів роботи 90

ДОДАТОК Г Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008: 2015..... 94

ВСТУП

У сучасних умовах цифрової трансформації дедалі більше інформаційних систем переходять до хмарної архітектури, що забезпечує масштабованість, гнучкість та високу доступність. Водночас це робить хмарні інфраструктури більш уразливими до різноманітних кіберзагроз, серед яких одним з найбільш небезпечних типів залишаються DDoS-атаки (Distributed Denial of Service). Такі атаки здатні порушити доступ до сервісів, спричинити значні фінансові збитки та підірвати довіру до провайдера хмарних послуг.

Особливої актуальності набуває проблема розпізнавання DDoS-атак у зашифрованому мережевому трафіку. За останні роки обсяг зашифрованого трафіку значно зріс, зокрема завдяки впровадженню протоколів TLS/HTTPS для забезпечення конфіденційності. Проте шифрування водночас ускладнює виявлення шкідливої активності, оскільки традиційні методи глибокої інспекції пакетів стають непридатними. В таких умовах зростає значення методів аналізу метаданих (тривалість сесій, частота пакетів, розмір, інтервали, тощо) як основного джерела інформації для прийняття рішень щодо безпеки.

Сучасні підходи до захисту мереж часто базуються на методах машинного навчання, однак більшість із них є статичними — після первинного навчання на історичних даних такі моделі слабо адаптуються до нових типів атак. Одним із перспективних напрямів є використання глибинного навчання з підкріпленням (Reinforcement Learning, RL), що дозволяє агенту самонавчатися в процесі взаємодії з мережею та адаптивно змінювати свою поведінку відповідно до змін у середовищі.

Метою роботи є дослідження можливості застосування глибинного навчання з підкріпленням (Reinforcement Learning) для побудови системи автоматичного розпізнавання DDoS-атак у зашифрованому трафіку на основі аналізу лише метаданих, без розшифрування вмісту мережевих пакетів.

Для досягнення поставленої мети в роботі вирішуються наступні завдання:

— проаналізувати сучасні методи виявлення DDoS-атак у зашифрованому трафіку;

— дослідити потенціал застосування reinforcement learning у контексті мережевої безпеки;

— розробити архітектуру системи з включенням RL-агента, орієнтованого на аналіз потокових метаданих;

— реалізувати прототип системи з модулем навчання та компонентами обробки, зберігання й візуалізації даних;

— провести експериментальне тестування ефективності запропонованого підходу;

— оцінити переваги моделі порівняно з класичними методами машинного навчання.

Методологічну основу дослідження становлять методи аналізу наукової літератури, побудова системної архітектури, використання технологій машинного навчання, зокрема глибинного підкріплення (Deep Q-Learning), а також інструменти обробки трафіку, зберігання часових даних і розгортання в мікросервісному середовищі. Практична частина реалізована на базі стеку Python, PyTorch, FastAPI, TimescaleDB, React.js і Docker.

Очікуваними результатами є створення функціонального прототипу системи, здатного виявляти DDoS-атаки у зашифрованому трафіку з використанням RL-агента, а також підтвердження його ефективності в реальному або наближеному до реального середовищі. Результати роботи можуть бути використані для подальшого розвитку систем кіберзахисту в хмарних інфраструктурах або адаптовані для інтеграції з існуючими платформами моніторингу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Огляд предметної галузі

Проблема виявлення та нейтралізації DDoS-атак залишається однією з ключових у сфері кібербезпеки. DDoS-атаки становлять серйозну загрозу для доступності інформаційних ресурсів, порушуючи штатну роботу серверів, веб-додатків і хмарних сервісів. Особливу небезпеку становГлять розподілені атаки, які здійснюються з великої кількості зкомпрометованих пристроїв (ботнетів) і можуть мати різну інтенсивність — від миттєвих перевантажень каналів до так званих "повільних" атак, що імітують легітимний трафік.

Сучасна тенденція до шифрування більшості мережевих з'єднань (через HTTPS, TLS тощо) значно ускладнює задачу виявлення атак. Згідно зі статистикою, понад 90% трафіку в Інтернеті у 2024 році передається у зашифрованому вигляді. Традиційні засоби виявлення загроз, засновані на глибокій інспекції пакетів (Deep Packet Inspection), втрачають ефективність, оскільки не мають доступу до вмісту пакетів без порушення конфіденційності. У таких умовах критично важливим стає аналіз метаданих — непрямої інформації про трафік, такої як частота запитів, розмір пакетів, тривалість сесій, час між запитами, розподіл трафіку по IP-адресах тощо.

Аналіз лише метаданих ускладнює задачу виявлення, особливо якщо DDoS-трафік маскується під легітимний або має низьку інтенсивність. Це зумовлює потребу у використанні інтелектуальних методів обробки даних, зокрема алгоритмів машинного та глибинного навчання. Класичні методи (наприклад, дерева рішень, SVM, K-Means) мають обмежену здатність до адаптації і часто демонструють знижену ефективність при зміні умов або типів атак.

У відповідь на ці виклики дослідницька спільнота звертає все більше уваги на глибинне навчання з підкріпленням (Reinforcement Learning, RL) як підхід, що дозволяє агентам навчатись через взаємодію з мережевим середовищем, отримуючи зворотній зв’язок у вигляді винагороди або штрафу за прийняті рішення. Такий підхід відкриває можливість створення адаптивних систем

виявлення, які не лише розпізнають атаки на основі вже відомих шаблонів, але й здатні навчатися в реальному часі, реагуючи на нові, ще не ідентифіковані загрози.

Крім того, DDoS-атаки в хмарних середовищах мають додаткову специфіку. З одного боку, хмарна інфраструктура забезпечує масштабованість і відмовостійкість, що може ускладнити завдання атакуючого. З іншого боку, складність хмарної архітектури і наявність багатьох точок доступу ускладнюють централізований контроль трафіку. Тому інтелектуальні агенти виявлення, які працюють автономно та мають можливість самонавчання, є особливо актуальними для таких сценаріїв.

Таким чином, предметна галузь дослідження охоплює перетин кількох напрямів: мережеву безпеку, аналіз зашифрованого трафіку, застосування алгоритмів машинного навчання, а також розробку адаптивних систем прийняття рішень у реальному часі. Інноваційність полягає в поєднанні цих підходів для вирішення актуального завдання — виявлення DDoS-атак у зашифрованому трафіку без використання глибокої інспекції пакетів.

1.2 Огляд сучасних підходів до виявлення DDoS-атак

Широке впровадження шифрування у мережеві з’єднання, зокрема використання протоколів TLS/HTTPS, значно ускладнило завдання виявлення DDoS-атак. Традиційні методи глибокої інспекції пакетів, які раніше дозволяли аналізувати вміст переданих даних, більше не мають доступу до застосункового рівня, оскільки шифрування приховує payload. Відтак дослідження і практика останніх років зосереджуються на аналізі поведінкових патернів і статистичних характеристик, що можуть бути отримані із метаданих – тобто непрямих ознак трафіку.

Найпоширенішим підходом у цій сфері є аналіз мережевих потоків. Це дозволяє працювати із такими параметрами, як розмір і кількість пакетів, тривалість з’єднання, інтервали між запитами, напрямок передачі та частота звернень до певних IP-адрес. Подібна інформація може бути зібрана за допомогою таких інструментів, як NetFlow, Zeek або CICFlowMeter. Дослідження

підтверджують, що саме ці характеристики дозволяють розпізнати DDoS-активність навіть без розшифрування вмісту пакетів.

Окремим напрямом є аналіз шаблонів TLS-з’єднань (так званих TLS fingerprints). Параметри початкового рукопотискання, набір криптографічних алгоритмів, часові характеристики handshake можуть слугувати індикаторами підозрілої активності, наприклад при масовій генерації схожих запитів із ботнетів. Хоча ці методи більш застосовні для класифікації типів додатків або виявлення шкідливого ПЗ, вони також використовуються у контексті DDoS для виявлення аномальних сценаріїв взаємодії на рівні шифрованого сеансу.

Класичні системи виявлення на основі аномалій формують модель «нормальної» поведінки користувачів та згодом виявляють відхилення від неї. У зашифрованому середовищі ці системи працюють зі статистичними і часовими ознаками, але їх ефективність часто обмежується високою чутливістю до змін у фоновому трафіку, що призводить до помилкових спрацювань.

Поступово ці традиційні підходи доповнюються або замінюються алгоритмами машинного навчання. Використання моделей супервізованого навчання, таких як SVM або дерева рішень, забезпечує певну ефективність за наявності добре розмічених даних. Проте саме глибинне навчання, зокрема згорткові та рекурентні нейронні мережі, відкриває нові можливості виявлення складних шаблонів, які недоступні класичним моделям. Такі мережі можуть аналізувати часову послідовність подій, виявляючи характерні риси DDoS-атак навіть на основі дуже обмеженої кількості параметрів.

Найбільш перспективним напрямом вважається застосування глибинного навчання з підкріпленням. Цей підхід дозволяє моделі не лише розпізнавати атаки, а й навчатися в процесі роботи за рахунок зворотного зв’язку з мережевим середовищем. Агент отримує винагороду за правильні дії (наприклад, блокування атаки або пропуск легітимного трафіку) і штрафується за помилки. Таким чином, він поступово вдосконалює свою політику прийняття рішень, адаптуючись до нових умов і типів атак. У наукових дослідженнях продемонстровано, що такі

моделі перевершують традиційні за точністю і стійкістю до невідомих загроз, а також здатні до роботи в режимі реального часу.

Крім того, сучасні системи захисту часто поєднують кілька підходів у рамках гібридної архітектури. Наприклад, використання евристичного аналізу для швидкої фільтрації, подальше уточнення рішень машинним навчанням, а також інтеграція із контролерами в SDN-мережах для динамічного блокування шкідливого трафіку. У таких архітектурах агент на основі reinforcement learning може виступати як адаптивний модуль, що координує поведінку всієї системи.

Таким чином, розвиток підходів до виявлення DDoS-атак у зашифрованому трафіку демонструє тенденцію до переходу від ручного аналізу до повністю автоматизованих, інтелектуальних систем. Глибинне навчання, зокрема з підкріпленням, відіграє ключову роль у цій трансформації, забезпечуючи високу точність, адаптивність і масштабованість у складних і динамічних мережевих середовищах.

1.3 Використання Reinforcement learning у виявленні DDoS-атак

Reinforcement learning (навчання з підкріпленням) — це клас методів машинного навчання, у яких агент навчається взаємодіяти з середовищем, приймаючи рішення, отримуючи зворотний зв’язок у вигляді винагороди або штрафу та на основі цього вдосконалюючи свою політику поведінки. У контексті кібербезпеки, зокрема виявлення DDoS-атак, цей підхід відкриває нові можливості для створення адаптивних, самонавчальних систем, здатних ефективно діяти в умовах постійної зміни типів загроз.

Застосування reinforcement learning у виявленні DDoS-атак дозволяє агенту аналізувати потік трафіку (на основі метаданих) та навчатися розпізнавати ознаки аномальної активності через послідовність рішень. Агент може отримувати на вхід статистичні ознаки мережевих потоків, такі як тривалість сесії, кількість пакетів, розмір, частота, інтервали між запитами тощо, і обирати дію: дозволити або заблокувати потік. За правильне рішення він отримує позитивну винагороду, за хибне — негативну. Така модель дозволяє навчатися навіть без явно розмічених

даних, що є важливою перевагою в умовах обмеженого доступу до якісних навчальних вибірок.

Порівняно з класичними супервізованими методами, які мають високу початкову точність на заздалегідь розмічених датасетах, reinforcement learning краще підходить для динамічних середовищ, де типи атак постійно змінюються, а розмітка або сигнатури відсутні. RL-модель здатна продовжувати навчання під час експлуатації, адаптуючись до нових видів атак у режимі реального часу. Це особливо актуально для зашифрованого трафіку, де вміст пакета недоступний, а шаблони атак можуть варіюватися від високочастотного флуду до повільних, добре замаскованих спроб виснаження ресурсів (SlowLoris, SlowHTTP тощо).

У низці досліджень було показано, що моделі типу Deep Q-Network (DQN), Double DQN або Actor-Critic здатні досягати високої точності в розпізнаванні шкідливого трафіку. Наприклад, агент, що навчається на симульованому або історичному трафіку, може сформувати ефективну політику виявлення, яка потім переноситься на реальні умови. У порівнянні з традиційними моделями машинного навчання, такі агенти демонструють вищу стійкість до нових, раніше невідомих атак. Крім того, завдяки можливості роботи з неперервними просторами станів та дій, ці моделі можуть враховувати більше контекстної інформації, приймати рішення не лише на основі одного потоку, а й ураховувати попередні взаємодії в мережі.

Перевагою reinforcement learning також є можливість вбудування у автономні системи прийняття рішень. RL-агент може не тільки виявляти загрозу, а й безпосередньо керувати реакцією системи: обмежувати пропускну здатність, перенаправляти трафік, генерувати правила блокування, або ж координувати дії між компонентами захисту в розподіленому середовищі. Таким чином, RL виступає не лише як інструмент виявлення, а як основа для побудови інтелектуальної системи реагування.

Попри свої переваги, використання reinforcement learning має і певні виклики. Найголовніший з них — це складність побудови та налаштування середовища навчання. Необхідно чітко визначити простір станів, допустимі дії та функцію

винагороди, яка б правильно стимулювала бажану поведінку. Невдалий вибір цих компонентів може призвести до того, що агент буде навчатися неефективно або навіть виробить шкідливу політику (наприклад, надмірне блокування нормального трафіку). Ще одна складність — висока обчислювальна вартість навчання, яка потребує потужних ресурсів або спеціалізованих платформ для моделювання середовища.

Незважаючи на ці обмеження, результати практичних експериментів свідчать про ефективність RL-агентів у задачах виявлення DDoS-атак. У роботах останніх років продемонстровано, що DRL-системи на основі аналізу метаданих можуть досягати точності понад 98–99%, водночас знижуючи кількість хибних спрацювань та забезпечуючи адаптивність до змін трафіку. У поєднанні з іншими технологіями, такими як SDN, хмарні платформи або edge-аналітика, reinforcement learning має потенціал стати ключовим компонентом систем захисту наступного покоління.

1.4 Тенденції та виклики у сфері виявлення DDoS-атак у зашифрованому трафіку

У міру того як більшість інтернет-комунікацій переходить до повністю зашифрованих форматів, зокрема через повсюдне використання протоколів TLS 1.3, виникає нова хвиля викликів у сфері забезпечення мережевої безпеки. Одним із ключових напрямів, де ці зміни мають найбільший вплив, є виявлення та протидія DDoS-атакам. Якщо раніше такі атаки часто можна було виявити за сигнатурами у вмісті пакетів або специфікою застосункового рівня, то тепер ці джерела інформації недоступні, що змушує фокусувати увагу винятково на поведінкових характеристиках трафіку.

Серед ключових тенденцій можна виділити зростання низькоінтенсивних та повільних DDoS-атак, які цілеспрямовано імітують легітимну активність. Такі атаки складно виявити, оскільки вони не створюють аномального навантаження миттєво, а поступово виснажують ресурси цільового сервера. У зашифрованому трафіку подібні атаки часто не відрізняються від звичайної поведінки користувача, що суттєво ускладнює класифікацію.

Іншим важливим трендом є зміщення парадигми виявлення від сигнатур до моделювання поведінки. Використання метаданих як основного джерела інформації про трафік стає стандартом для сучасних систем безпеки. У зв’язку з цим зростає роль алгоритмів машинного навчання, які здатні виявляти приховані закономірності в статистичних та часових характеристиках мережевих потоків. Зокрема, моделі на основі глибинного навчання демонструють високий потенціал завдяки здатності автоматично вилучати інформативні ознаки з потоку даних.

Проте на цьому фоні виникає низка технічних і методологічних викликів. По-перше, недостатня кількість відкритих, якісно розмічених датасетів, що містять зашифрований DDoS-трафік, ускладнює навчання та тестування моделей. Більшість наявних датасетів або зібрані в лабораторних умовах, або не охоплюють сучасних типів атак, що знижує релевантність моделей у реальному середовищі.

По-друге, використання алгоритмів машинного навчання в режимі реального часу супроводжується проблемами масштабування, продуктивності та узгодженості рішень. Моделі, що демонструють високу точність у статичних тестах, можуть виявитися неефективними при роботі в динамічному середовищі із змінним трафіком. Це особливо критично для хмарних середовищ, де навантаження може змінюватись у великих діапазонах протягом коротких інтервалів часу.

Ще один виклик полягає в ризику хибних спрацювань. У разі виявлення DDoS-атаки помилкове блокування легітимного трафіку може призвести до порушення доступності сервісів для реальних користувачів. Тому системи мають не лише бути точними у виявленні загроз, але й забезпечувати високу селективність і пояснюваність прийнятих рішень.

Нарешті, важливим аспектом є адаптивність систем виявлення до нових, ще невідомих атак. У зв’язку з цим дослідницька увага поступово зміщується до технологій reinforcement learning, які дозволяють створювати агентів, здатних самостійно вдосконалювати свою політику реагування у процесі роботи. Це відкриває перспективи побудови самооновлюваних систем, здатних оперативно

підлаштовуватись до змін у характері трафіку, особливо у випадках, коли традиційні методи виявлення втрачають ефективність.

Таким чином, галузь виявлення DDoS-атак у зашифрованому трафіку перебуває на етапі активного переходу до інтелектуальних, поведінкових і адаптивних моделей аналізу. Подальші успіхи в цій сфері залежать від розвитку ефективних моделей навчання, накопичення якісних даних, а також інтеграції безпекових рішень у хмарні екосистеми із урахуванням їх масштабності та динаміки.

1.5 Постановка задачі

З огляду на стрімке зростання обсягів зашифрованого мережевого трафіку та неможливість аналізу вмісту пакетів, виникає потреба в застосуванні Double DQN для виявлення DDoS-атак виключно на основі метаданих. Застарілі сигнатурні методи і класичні алгоритми машинного навчання не дають бажаних результатів у таких умовах, отже варто обрати підхід із глибоким підкріпленим навчанням.

Для впровадження цього рішення слід вирішити такі завдання:

— виокремити цифрові й часові характеристики потоків, які лягатимуть в основу векторних представлень;

— створити емуляційне середовище, що відтворює нормальні та атакувальні сценарії для тренування Double DQN-агента;

— сформулювати функцію винагороди, що збалансує чутливість детектора й рівень помилкових спрацювань;

— спроєктувати й реалізувати архітектуру Double DQN, оптимізовану для обробки цих векторних даних;

— інтегрувати модель до веб-додатку, який відображатиме статистику, дозволятиме коригувати налаштування та демонструватиме результати в реальному часі;

— налагодити збереження детекційних даних у базі часових рядів для подальшого аналізу й звітності;

— розробити процедуру верифікації системи в умовах реального трафіку та підготувати експлуатаційну документацію з рекомендаціями з розгортання.

У результаті створюється повноцінний інструмент на базі Double DQN із веб-інтерфейсом для моніторингу й конфігурації, здатний ефективно виявляти DDoS-активність у зашифрованому трафіку.

2 ОГЛЯД Й АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

Для формування теоретичної основи дослідження, визначення актуальних підходів до виявлення DDoS-атак у зашифрованому трафіку та обґрунтування вибору методу глибинного навчання з підкріпленням, було здійснено огляд наукових публікацій, матеріалів конференцій, технічної документації та публічних дослідницьких звітів за останні роки. Особливу увагу приділено роботам, які фокусуються на обробці метаданих мережевих потоків без доступу до зашифрованого вмісту, а також на застосуванні алгоритмів машинного навчання — зокрема, reinforcement learning — у сфері кібербезпеки.

Аналіз джерел дозволив охопити ключові аспекти тематики: загальні підходи до виявлення DDoS-атак у TLS/HTTPS-трафіку, характеристики доступних датасетів, структуру агентно-орієнтованих систем захисту, типові архітектури RL-моделей, проблематику генералізації моделей на нові сценарії атак, а також приклади практичного використання RL у тестових і реальних середовищах.

Далі наведено аналіз відібраних джерел, що мали найбільшу практичну та методологічну цінність у межах дослідження. Вони стали основою для розробки власної архітектури рішення, формалізації середовища та вибору RL-алгоритму, а також визначення обмежень, на які необхідно зважати при реалізації MVP-прототипу системи виявлення DDoS-атак на основі метаданих.

2.1 Огляд основних джерел

Для глибшого розуміння предметної області та обґрунтування вибору технологічного підходу було здійснено відбір і аналіз ключових наукових та технічних джерел, що безпосередньо стосуються виявлення DDoS-атак у зашифрованому трафіку, аналізу мережевих метаданих та використання алгоритмів reinforcement learning у системах мережевої безпеки. У цьому підрозділі наведено короткий огляд тих праць, які мали найбільший вплив на структуру і зміст дослідження.

2.1.1 Загальні аспекти розпізнавання DDoS-атак у зашифрованому трафіку

У сучасних умовах, коли більшість мережевого трафіку шифрується з метою забезпечення конфіденційності та цілісності переданих даних, виявлення DDoS-атак без доступу до вмісту пакетів стає дедалі складнішим завданням. Більшість традиційних систем виявлення загроз базуються на аналізі відкритого трафіку або специфічних сигнатур, що більше не є релевантним для HTTPS/TLS-з'єднань. Це зумовлює зміну фокусу досліджень у бік поведінкових моделей аналізу трафіку та використання метаданих як основного джерела ознак для класифікації.

Загальні підходи до вирішення цієї задачі представлені у низці оглядових і прикладних досліджень. Наприклад, у роботі Kheddar et al. [1] систематизовано існуючі методи застосування reinforcement learning у задачах мережевої безпеки, включаючи виявлення DDoS-атак у зашифрованому середовищі. Автори акцентують на важливості формалізації середовища навчання та правильному визначенні функції винагороди, що є критично важливим для ефективного функціонування RL-агента.

Оглядова праця Ferriyan et al. [2], що супроводжує публікацію датасету HIKARI-2021, наголошує на ключових труднощах у виявленні атак у зашифрованому трафіку, зокрема через обмежену кількість доступних ознак та необхідність адаптації моделей до реальних сценаріїв. Автори зазначають, що саме агрегація статистичних метрик потоку (тривалість, обсяг, частота, напрямок) є основою для розпізнавання аномальної поведінки, включаючи DDoS-атаки.

Крім того, Vargas-Rosales et al. [3] у своїй роботі вказують на важливість побудови тестових стендів, які дозволяють генерувати контрольований, але реалістичний зашифрований трафік. Їх досвід із SDN-SlowRate-DDoS демонструє, що повільні DDoS-атаки, які імітують звичайні користувацькі запити, вимагають високоточних і чутливих систем виявлення, здатних враховувати навіть незначні відхилення у метаданих.

Загалом, у джерелах чітко простежується консенсус щодо ключових проблем: недоступність payload-даних, обмежена спостережуваність внутрішніх

характеристик сеансів, обмеження щодо точності класичних моделей і зростаюча потреба в адаптивних інтелектуальних підходах, таких як RL. Вивчення загальних аспектів дозволяє сформулювати вимоги до архітектури системи виявлення, обрати релевантні ознаки для аналізу та визначити найбільш ефективні методи побудови моделей у рамках зашифрованого середовища.

2.1.2 Методи розпізнавання DDoS-атак по зашифрованому трафіку

З огляду на неможливість застосування глибокої інспекції пакетів у зашифрованому трафіку, дослідження зосереджуються на використанні альтернативних методів виявлення DDoS-атак, здебільшого пов’язаних із поведінковим аналізом мережевих потоків та обробкою метаданих. Аналіз наукових джерел засвідчив наявність кількох ключових підходів, які були реалізовані на практиці та показали ефективність у виявленні як класичних, так і повільних типів атак.

Один із базових напрямів — класифікація трафіку на основі агрегованих статистичних ознак. У роботі Ferriyan et al. [2], присвяченій датасету HIKARI-2021, обґрунтовується підхід, згідно з яким аналіз параметрів, таких як кількість байт, розмір пакетів, тривалість сесії, інтервали між повідомленнями та напрямок трафіку, дозволяє досягти високої точності класифікації навіть у відсутність доступу до вмісту даних. Ці ознаки виступають основою для формування векторів характеристик потоків, на основі яких будуються моделі розпізнавання.

Іншим ефективним підходом є застосування методів машинного навчання, які оперують лише метаданими. Наприклад, у дослідженні Yang et al. [4] використано гібридну модель, яка поєднує згорткову нейронну мережу (CNN) для вилучення ознак із потоків зашифрованого трафіку та reinforcement learning для прийняття рішень щодо класифікації. Результати свідчать про можливість досягнення точності понад 99% при виявленні DDoS-атаки у TLS-з’єднаннях. Цей підхід демонструє перевагу глибоких моделей над традиційними статистичними методами у складних середовищах із зашумленими даними.

Дослідження Hu et al. [5] зосереджується на моделі Double Deep Q-Network (DDQN), що дозволяє підвищити стійкість до перенавчання та забезпечити стабільність прийняття рішень у змінному середовищі. Автори акцентують увагу на важливості генералізації: модель повинна не тільки демонструвати точність на навчальних даних, а й ефективно працювати з новими потоками, які не входили до тренувального набору. Встановлено, що DDQN перевершує класичні моделі DQN у задачах виявлення ботнет- і DDoS-активності в зашифрованому трафіку.

Ще один важливий напрям — використання моделей аномального виявлення. Вони базуються на побудові профілю нормального трафіку і виявленні відхилень. Проте в зашифрованому середовищі такі моделі часто демонструють підвищену кількість хибнопозитивних спрацювань, особливо у хмарних або мобільних мережах, де поведінка легітимного трафіку є нестабільною. Саме тому останні тенденції вказують на поступову відмову від «чистих» anomaly-based моделей на користь гібридних архітектур, які включають модулі як детекції, так і класифікації з адаптивними механізмами прийняття рішень.

У роботі Vargas-Rosales et al. [3], присвяченій виявленню повільних DDoS-атак у SDN-мережах, використано аналіз темпоральних характеристик потоків. Навіть незначні зміни в інтервалах між запитами або частоті сесій можуть виступати ключовими маркерами атаки. Це підтверджує, що ефективне розпізнавання атак у зашифрованому середовищі можливе завдяки комплексному аналізу часових і просторових ознак, а не через прямий доступ до даних.

Узагальнюючи аналіз джерел, можна зробити висновок, що найбільш ефективними методами розпізнавання DDoS-атак у зашифрованому трафіку є ті, що:

— працюють з багатовимірними метаданими потоків;

— враховують динамічні характеристики трафіку;

— поєднують глибоке навчання з адаптивними агентними підходами, зокрема reinforcement learning;

— підтримують оновлення політики прийняття рішень у режимі онлайн.

— застосування таких методів дозволяє досягти високої точності в складних умовах, коли класичні інструменти є неефективними або непридатними до розгортання у зашифрованих середовищах.

2.1.3 Використання машинного навчання у прогнозуванні загроз

Reinforcement Learning (RL), або навчання з підкріпленням, останніми роками отримало широке визнання в галузі мережевої безпеки завдяки своїй здатності моделювати динамічну поведінку системи в умовах змінного середовища. На відміну від класичних моделей, RL-агент здатний навчатися на основі досвіду взаємодії з мережею, приймаючи рішення в реальному часі та адаптуючись до нових типів атак, зокрема — до DDoS у зашифрованому трафіку.

У роботі Yungaicela-Naula et al. [6] представлено повноцінну систему виявлення повільних DDoS-атак у SDN-середовищі, засновану на deep reinforcement learning. Агент працює без доступу до payload-даних, використовуючи лише метадані потоків. Перевагою цієї архітектури є здатність адаптуватися до змін у профілі трафіку, що дозволяє ефективно виявляти атаки, які імітують легітимну поведінку. Модель досягла точності понад 97% у тестових сценаріях.

Дослідження Hu et al. [5] поглиблює ідею RL у контексті DDoS, впроваджуючи модифікацію класичної Q-Learning моделі у вигляді Double Deep Q-Network (DDQN). Завдяки розділенню оцінки поточної та таргетної політик, ця модель знижує ризик переоцінки дій агента, що призводить до стабільнішого навчання. Автори відзначають, що DDQN-модель краще узагальнює знання на нові сценарії та забезпечує високу точність виявлення ботнет- та DDoS-активності у потоках TLS-трафіку.

Особливо цінним є підхід, запропонований Yang et al. [4], який поєднує CNN та RL у гібридній системі. Глибока згорткова мережа виконує вилучення ознак із потоку метаданих, після чого RL-агент приймає рішення щодо блокування чи пропуску з’єднання. Така багаторівнева структура дозволяє одночасно забезпечити глибоку семантичну обробку даних та адаптивність до змін у поведінці атак.

Модель показала точність понад 99% у виявленні зашифрованих шкідливих потоків.

Джерела також звертають увагу на важливість правильного формулювання середовища навчання. Наприклад, у Kheddar et al. [2] підкреслено, що ефективність RL-моделі значною мірою залежить від визначення простору станів, дій і функції винагороди. Середовище повинно відображати реалістичні сценарії поведінки трафіку, а агент має бути заохочений не лише за точне виявлення атак, але й за зменшення кількості хибних спрацювань.

Ще одним важливим прикладом є робота Vargas-Rosales et al. [3], де RL-агент інтегрований у систему моніторингу SDN, що забезпечує динамічну зміну правил маршрутизації для мінімізації впливу атак. Це демонструє практичну придатність RL-архітектур до автоматизованого управління безпекою в реальному часі, включаючи автономне реагування на загрози.

Усі ці джерела підтверджують, що Reinforcement Learning є не лише теоретично обґрунтованим, але й практично дієвим підходом до виявлення DDoS-атак у зашифрованому трафіку. Його ключові переваги — здатність до самонавчання, адаптація до змін, відсутність потреби в ручній розмітці даних — роблять його особливо перспективним у контексті сучасних хмарних інфраструктур, де трафік постійно змінюється, а загрози еволюціонують.

Разом із тим, ефективне застосування RL вимагає уважного підходу до моделювання, достатньої обчислювальної бази для тренування, а також доступу до репрезентативних даних або механізмів генерації реалістичних сценаріїв. Тому подальші дослідження в цій сфері повинні зосереджуватись на удосконаленні середовищ для тренування агентів, генерації навчального трафіку та побудові гібридних систем із можливістю онлайн-адаптації.

2.2 Аналіз літератури

Після проведення огляду ключових наукових джерел доцільним є критичний аналіз виявлених підходів, методів та рішень з точки зору їхньої ефективності, обмежень та можливостей практичного застосування. Метою цього аналізу є

визначення сильних і слабких сторін існуючих досліджень, а також виявлення науково-технічних прогалин, які можуть бути заповнені в межах власного дослідження. У цьому підрозділі подано систематизовану оцінку літературних джерел, що лягли в основу формування технічного рішення щодо виявлення DDoS-атак у зашифрованому трафіку з використанням reinforcement learning.

2.2.1 Основні теорії та концепції

Аналіз літературних джерел дозволяє виділити низку базових теоретичних підходів, які формують фундамент сучасних рішень у сфері виявлення DDoS-атак у зашифрованому трафіку. Ці концепції визначають архітектуру, методологію побудови систем, а також критерії оцінювання їхньої ефективності.

Перш за все, ключовою концепцією є модель мережевого потоку як джерела поведінкових ознак. Оскільки в умовах шифрування вміст пакетів недоступний, основним джерелом інформації стають метадані потоків: частота, розмір, тривалість, напрямок, інтервали між повідомленнями. Ця модель лежить в основі переважної більшості досліджень, зокрема у роботах Ferriyan et al. [2] (HIKARI-2021) та Vargas-Rosales et al. [3] (SDN-SlowRate-DDoS), де продемонстровано, що поведінкові характеристики можуть бути достатніми для точного розпізнавання шкідливої активності.

Другою фундаментальною ідеєю є побудова профілю нормальної поведінки або baseline model. Цей підхід є основою для методів аномального виявлення, які спрацьовують у разі суттєвих відхилень від очікуваної динаміки трафіку. Хоча anomaly-based методи часто використовуються в ізоляції, вони також слугують допоміжним елементом у гібридних системах із машинним навчанням.

Третім базовим положенням є концепція моделі прийняття рішень як марковського процесу, що використовується в reinforcement learning. У цьому контексті трафік і стан мережі розглядаються як середовище, у якому діє агент, що намагається максимізувати кумулятивну винагороду шляхом послідовного прийняття рішень (класифікація потоку як «норма» або «атака»). Як підтверджено у роботах Hu et al. [5] та Yang et al. [4], формалізація задачі в термінах Markov

Decision Process (MDP) дозволяє ефективно тренувати агента навіть на неповних або незбалансованих даних.

Окрему групу складають концепції глибокого навчання, які забезпечують автоматичне вилучення ознак з метаданих. Архітектури згорткових (CNN) або рекурентних (RNN, LSTM) мереж дозволяють розпізнавати складні патерни в часових послідовностях, що особливо актуально для повільних або прихованих DDoS-атак. У поєднанні з RL-механізмами ці мережі слугують основою для гібридних інтелектуальних агентів.

Таким чином, сучасні наукові дослідження базуються на поєднанні кількох ключових теорій: поведінковий аналіз за метаданими, профілювання нормальної активності, машинне навчання для класифікації та адаптивні агентні моделі на основі підкріплення. Системне використання цих концепцій дозволяє створювати ефективні рішення для виявлення DDoS-атак навіть у складних умовах зашифрованого трафіку.

2.2.2 Моделі та методи аналізу

Літературний аналіз показує, що основними моделями, які застосовуються для виявлення DDoS-атак у зашифрованому трафіку, є алгоритми машинного навчання, глибинні нейронні мережі та моделі з підкріпленням.

Супервізовані методи, такі як Random Forest, SVM, та XGBoost, часто використовуються для класифікації потоків на основі метаданих. Вони забезпечують хорошу початкову точність, але мають обмежену здатність до адаптації. Глибинні моделі (CNN, RNN) дозволяють автоматично виділяти ознаки з потокових даних і враховують часову структуру трафіку.

Особливу увагу в дослідженнях приділено моделям Reinforcement Learning, зокрема Deep Q-Network (DQN) і Double DQN. Вони розглядають задачу виявлення атак як процес прийняття рішень у змінному середовищі. Такі моделі показують високу адаптивність та здатність навчатися без повної розмітки даних.

Методи аналізу в джерелах базуються переважно на метриках точності, повноти, рівня хибних спрацювань, а також стійкості моделі до нових типів

трафіку. Усі ці підходи мають спільну рису — орієнтацію на використання лише метаданих, що є принципово важливим для зашифрованого середовища.

2.2.3 Ефективність існуючих підходів

Проаналізовані наукові джерела свідчать про те, що більшість сучасних підходів до виявлення DDoS-атак у зашифрованому трафіку досягають високих показників ефективності в умовах контрольованого середовища. Зокрема, моделі глибинного навчання та reinforcement learning демонструють точність виявлення, яка перевищує 95–99% на тестових вибірках, як це показано у працях Yang et al. [4], Hu et al. [5] та інших.

Найкращі результати фіксуються при використанні комбінованих рішень, які поєднують вилучення ознак із глибоких нейромереж та адаптивне навчання агентів у змінному середовищі. Такі моделі не лише правильно класифікують відомі шаблони атак, але й здатні виявляти нові патерни поведінки трафіку, що відрізняє їх від класичних статичних систем.

Водночас ефективність багатьох підходів різко знижується при переході від лабораторних умов до реальних мереж. Це зумовлено обмеженою генералізованістю моделей, залежністю від обраного датасету та змінною поведінкою легітимного трафіку [10]. Крім того, в окремих випадках RL-моделі демонструють нестабільність у навчанні через некоректну постановку функції винагороди або надмірну складність середовища.

У підсумку можна стверджувати, що найбільш ефективними виявляються адаптивні, гібридні моделі, які враховують часову динаміку трафіку, працюють виключно на основі метаданих та мають можливість самонавчання. Проте їх практичне застосування потребує додаткової оптимізації, зокрема у частині продуктивності, стійкості до нових атак та мінімізації хибнопозитивних спрацювань.

2.3 Оцінка актуальності та новизни

Аналіз літературних та науково-технічних джерел підтверджує високу актуальність дослідження методів виявлення DDoS-атак у зашифрованому трафіку, зокрема у контексті захисту хмарних систем. Зі зростанням частки TLS/HTTPS-з'єднань у глобальному трафіку традиційні системи виявлення, що покладаються на інспекцію вмісту пакетів, втрачають ефективність. Це обумовлює потребу у побудові нових підходів, здатних працювати виключно з метаданими — такими як частота, обсяг, тривалість і послідовність передач у потоці.

Новизна дослідження полягає у застосуванні reinforcement learning до задачі виявлення DDoS-атак у зашифрованому трафіку, що поєднує адаптивність, самонавчання та можливість роботи у динамічному середовищі без потреби в повністю розмічених даних. На відміну від класичних супервізованих моделей, RL-підхід дозволяє агенту оновлювати свою політику на основі зворотного зв’язку в процесі експлуатації, що особливо важливо в умовах постійної зміни шаблонів атак.

Крім того, актуальність підтверджується обмеженим числом публікацій, присвячених безпосередньо темі RL у контексті зашифрованого трафіку [12]. Більшість існуючих робіт або зосереджені на відкритому трафіку, або не розглядають повністю автоматизовані агентні системи. Отже, запропоноване в роботі дослідження розкриває перспективний напрям, який потребує подальшої практичної розробки та апробації.

Таким чином, поєднання глибинного навчання з підкріпленням, аналізу виключно метаданих та адаптивної архітектури забезпечує як теоретичну новизну підходу, так і прикладну цінність для розвитку сучасних систем захисту від DDoS-атак у хмарних та гібридних мережевих середовищах.

2.4 Висновки з огляду

Огляд і аналіз наукових та технічних джерел показали, що проблема виявлення DDoS-атак у зашифрованому трафіку є актуальною та недостатньо

вирішеною на практиці. Основні труднощі пов’язані з неможливістю використання глибокої інспекції пакетів, високим рівнем шуму в метаданих, динамічністю атак та обмеженістю відкритих датасетів.

Найбільш ефективними серед сучасних підходів виявилися адаптивні методи, побудовані на базі глибинного навчання та reinforcement learning. Вони демонструють високу точність і гнучкість, а також здатність до самонавчання в умовах мінливого середовища. Особливої уваги заслуговують моделі типу DQN та DDQN, які дозволяють формалізувати задачу виявлення атак як процес оптимізації поведінки агента у середовищі з неповною інформацією.

Крім того, було встановлено, що поєднання глибоких нейромереж з механізмами прийняття рішень RL-агента забезпечує найвищу якість виявлення у випадках, коли атаки маскуються під легітимний трафік. Також важливою є здатність таких систем працювати на основі метаданих, що відкриває можливість інтеграції в зашифровані канали без порушення конфіденційності.

Таким чином, результати огляду сформували теоретичне підґрунтя для вибору підходу, який буде реалізовано у рамках дослідження: побудову моделі виявлення DDoS-атак у зашифрованому трафіку на основі аналізу метаданих з використанням методів глибинного навчання з підкріпленням.

3 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

На основі проведеного аналізу літературних джерел, сучасних тенденцій та практик у сфері виявлення DDoS-атак у зашифрованому трафіку сформовано концептуальну основу дослідження. Теоретичне дослідження зосереджено на формалізації задачі виявлення атак як задачі прийняття рішень у динамічному середовищі, яку можна вирішувати за допомогою reinforcement learning. У цьому розділі розглядаються принципи побудови моделі агентної системи, її роль у середовищі аналізу мережевого трафіку, а також визначаються ключові параметри, які можуть бути використані як вхідні ознаки для RL-агента.

Окрему увагу приділено обґрунтуванню вибору типу агента, функції винагороди, простору станів і дій, а також архітектурним особливостям глибоких Q-мереж. Також проаналізовано особливості представлення зашифрованого трафіку у вигляді формалізованого потоку даних для подальшої обробки. Метою цього етапу є створення методичної бази для побудови прототипу системи виявлення DDoS-атак у зашифрованому трафіку, орієнтованої на використання лише метаданих.

3.1 Архітектура та проектування ПЗ

Для реалізації системи виявлення DDoS-атак у зашифрованому трафіку з використанням reinforcement learning необхідно чітко визначити архітектуру програмного забезпечення, його ключові компоненти та логіку взаємодії між ними. У цьому підрозділі розглядається загальна структура програмного рішення, описуються його функціональні модулі, обґрунтовується вибір технологічного стеку та підхід до інтеграції агентної моделі RL у процес обробки мережевого трафіку на основі метаданих.

3.1.1 Загальна структура архітектури

Розроблена система виявлення DDoS-атак у зашифрованому трафіку базується на агентно-орієнтованій архітектурі, яка дозволяє реалізувати адаптивну

поведінку на основі методів глибинного навчання з підкріпленням. Загальна структура побудована за принципом модульності, що забезпечує гнучкість, масштабованість та можливість інтеграції в локальне або хмарне середовище.

На найвищому рівні архітектура включає такі основні компоненти:

— Модуль збору та попередньої обробки трафіку — відповідальний за фільтрацію, агрегацію та трансформацію сирих мережевих потоків у набір метаданих, придатних для аналізу. Збір трафіку може здійснюватися за допомогою інструментів типу Zeek або CICFlowMeter.

— Модуль формалізації середовища — реалізує представлення трафіку як послідовності станів для агента. Він визначає простір станів, набір дій та функцію винагороди, що є основою для навчання моделі reinforcement learning.

— RL-агент — головний інтелектуальний компонент системи, який приймає рішення щодо класифікації потоку (атака/норма) та дій у середовищі. Агент навчається за допомогою алгоритму Deep Q-Learning або його модифікації (наприклад, Double DQN) на основі взаємодії зі сформованим середовищем.

— Модуль прийняття рішень і реакцій — реалізує відповідні дії на основі рішень агента, зокрема генерацію сповіщень, формування правил блокування або адаптацію політик маршрутизації (в разі інтеграції з SDN).

— Модуль візуалізації та моніторингу — забезпечує інтерфейс для аналізу результатів роботи системи, перегляду статистики, журналів, динаміки навчання агента та індикаторів ефективності в реальному часі.

— Архітектура системи підтримує як офлайн-режим навчання агента на попередньо зібраних датасетах, так і онлайн-режим роботи, де агент застосовується до реального потоку даних з можливістю подальшого донавчання.

Таким чином, запропонована архітектура поєднує модулі збору даних, інтелектуального аналізу, автоматичного реагування та інтерактивного моніторингу, що дозволяє реалізувати повноцінну систему виявлення та протидії DDoS-атакам у зашифрованому трафіку з використанням reinforcement learning.

3.1.2 Архітектурний стиль

Система побудована за принципами мікросервісної архітектури, що дозволяє запускати окремі компоненти (модуль збору, модуль аналізу, зберігання, інтерфейс) у вигляді незалежних сервісів. Для комунікації між компонентами може використовуватись REST API або брокер повідомлень.

3.1.3 Візуалізація

Для кращого розуміння структури системи виявлення DDoS-атак у зашифрованому трафіку було розроблено низку діаграм, що ілюструють логіку роботи, взаємозв'язки компонентів та сценарії взаємодії користувача з системою. Ці візуальні моделі відображають ключові функціональні елементи програмного забезпечення, побудованого на основі мікросервісної архітектури.

Use Case діаграма моделює основні сценарії використання системи з боку користувача. Зокрема, користувач може запускати аналіз трафіку, переглядати список підозрілих потоків, здійснювати пошук за IP-адресою, змінювати налаштування алгоритмів та експортувати результати. Ці сценарії відображено на рисунку 3.1.

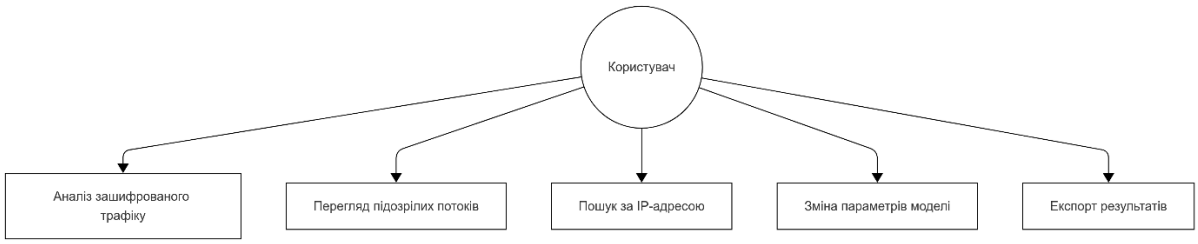


Рисунок 3.1 – Use Case діаграма взаємодії користувача з системою (рисунок виконано самостійно)

UML-діаграма компонентів (рис. 3.2) відображає структуру системи з позиції її основних модулів і способу їх взаємодії. Вона включає:

— зовнішнє джерело трафіку (PCAP або live),

— модуль збору та підготовки даних (FeatureExtractor),

— модуль аналізу (AutoEncoder та RNN),

— сховище результатів (PostgreSQL і InfluxDB),

— інтерфейс користувача з REST API.

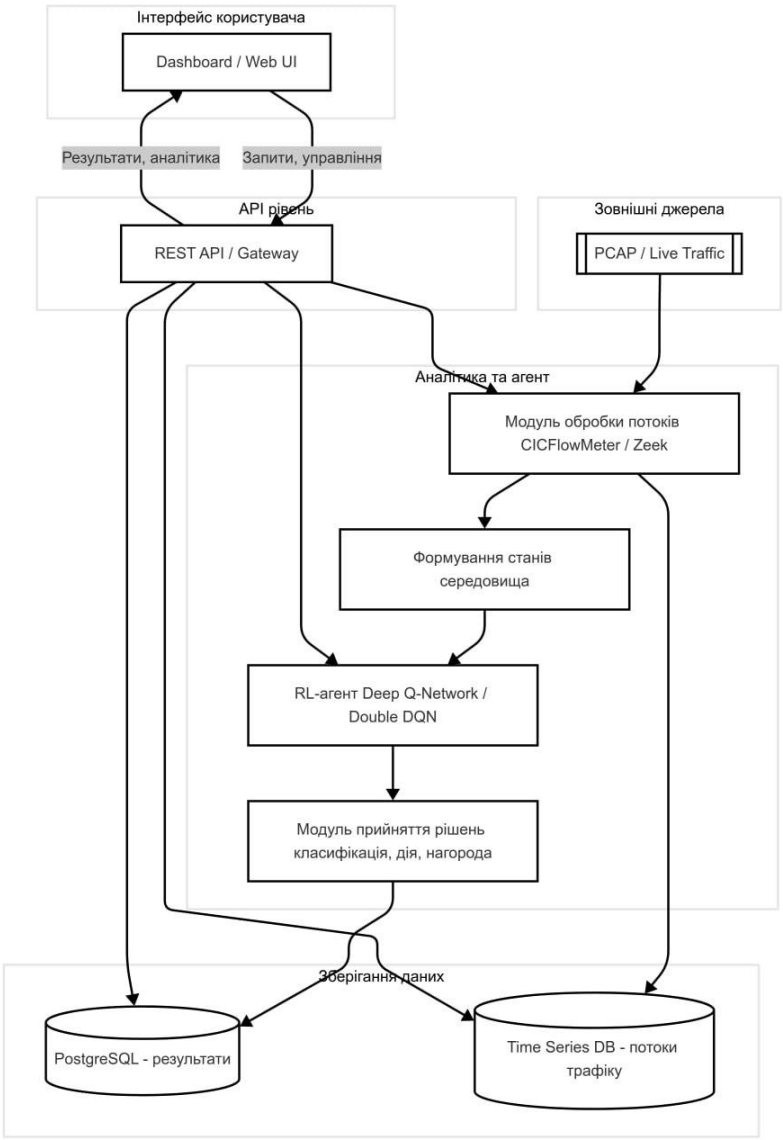


Рисунок 3.2 – UML-діаграма компонентів системи виявлення DDoS-атак (рисунок виконано самостійно)

ER-діаграма бази даних (рис. 3.3) демонструє логічну структуру даних, що використовуються системою. Основними сутностями є:

— Flows – мережеві потоки;

— AnalysisResults – результати обробки;

— Users – зареєстровані користувачі;

— Logs – дії користувачів у системі.

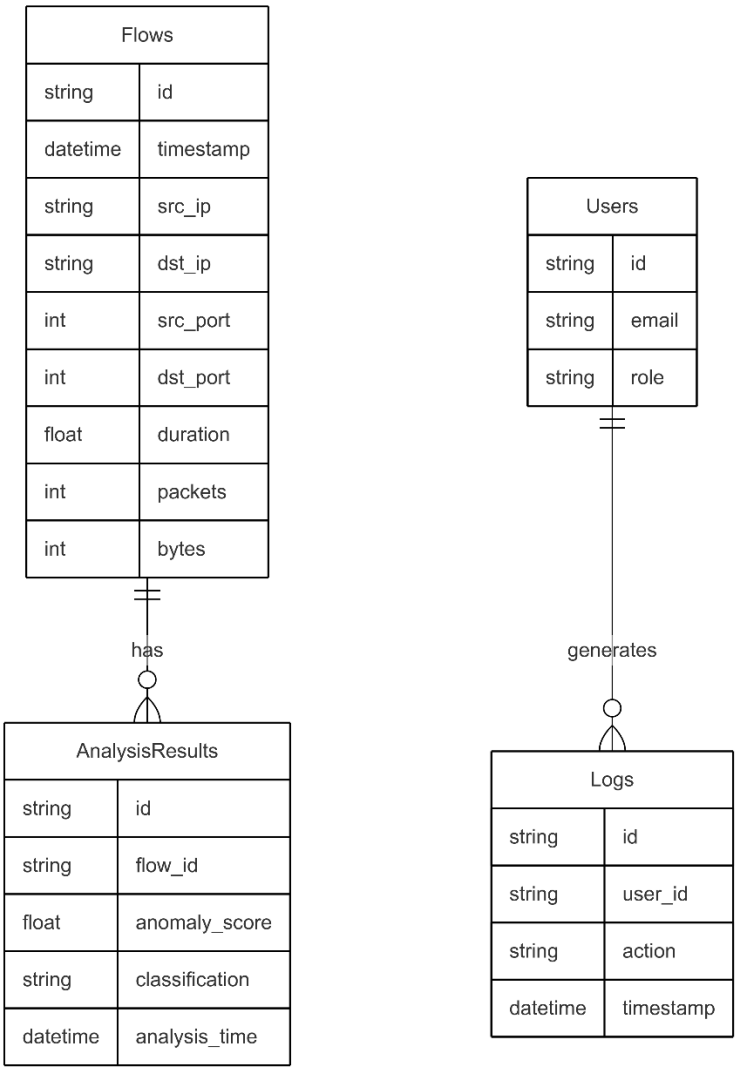


Рисунок 3.3 – Схема бази даних (рисунок виконано самостійно)

Усі візуалізації створено відповідно до логіки, закладеної в архітектурі системи, та забезпечують чітке розуміння її структури, ролей і взаємодій. Це дозволяє легко масштабувати рішення, додавати нові джерела трафіку або моделі аналізу без істотних змін основної логіки.

3.2 Проектування структури зберігання даних

Ефективне функціонування системи виявлення DDoS-атак вимагає належної організації зберігання та обробки мережевих метаданих. У цьому підрозділі розглядається структура зберігання даних, яка забезпечує зручний доступ до поточних та історичних потоків, підтримує процес навчання RL-агента, а також

дозволяє зберігати результати класифікації, стани середовища та статистику взаємодій у процесі роботи системи.

3.2.1 Вибір технологій

Проєктування системи зберігання даних у контексті виявлення DDoS-атак у зашифрованому трафіку передбачає обробку великої кількості потокових метаданих у режимі реального часу або з високою частотою оновлення. [11] Тому вибір технологій зберігання та доступу до даних має базуватись на критеріях масштабованості, продуктивності, гнучкості у структурі даних, а також простоти інтеграції з аналітичними модулями та моделями навчання.

Для зберігання потокових даних було обрано базу даних типу time-series InfluxDB [9]. Такі системи оптимізовані для роботи з часовими рядами і добре підходять для зберігання агрегованих характеристик мережевих потоків: розмірів пакетів, частоти запитів, інтервалів між сеансами тощо. Вони дозволяють ефективно виконувати запити з фільтрацією за часовими вікнами, що важливо при побудові профілів трафіку та аналізі трендів.

Для зберігання додаткових даних, зокрема результатів класифікації, станів агента, інформації про дії та значення функцій винагороди, доцільно використовувати реляційну СУБД, таку як PostgreSQL. Вона забезпечує структуроване зберігання, підтримку складних запитів та є сумісною з більшістю фреймворків для машинного навчання.

Також у системі використовується формат зберігання даних у вигляді CSV або Parquet-файлів для підготовки навчальних вибірок, збереження результатів експериментів та обміну даними між компонентами. Такий підхід є простим у реалізації, забезпечує кросплатформеність і добре підтримується бібліотеками обробки даних (Pandas, PyTorch, TensorFlow).

У разі масштабування системи або переходу до хмарного середовища (наприклад, AWS), можлива інтеграція з S3-сховищами для архівації історичних даних, а також використання Amazon RDS для підтримки PostgreSQL та Amazon Timestream для потокової аналітики.

Таким чином, вибрані технології дозволяють забезпечити надійне, ефективне та масштабоване зберігання даних для потреб як аналітики, так і машинного навчання в рамках реалізації системи на основі reinforcement learning.

3.2.2 Схема бази даних

Для зберігання та обробки мережевого трафіку, результатів аналізу та логів взаємодії з системою розроблено реляційну модель бази даних, яка відповідає вимогам до структурованого зберігання метаданих потоків, результатів класифікації та аудиту дій користувачів.

Схема бази даних включає чотири основні сутності:

— Flows — центральна таблиця, яка містить інформацію про окремі мережеві потоки. Вона зберігає такі поля: унікальний ідентифікатор потоку (id), мітку часу (timestamp), IP-адреси джерела і призначення (src_ip, dst_ip), порти (src_port, dst_port), тривалість сеансу (duration), кількість пакетів (packets) та обсяг трафіку в байтах (bytes). Ці дані є базовим джерелом для подальшої аналітики.

— AnalysisResults — таблиця результатів обробки кожного потоку. Вона пов’язана з таблицею Flows через поле flow_id. Зберігає ідентифікатор результату (id), оцінку аномальності (anomaly_score), тип класифікації (classification) та час виконання аналізу (analysis_time).

— Users — таблиця зареєстрованих користувачів системи. Містить поля id, email, role (роль користувача — аналітик, адміністратор тощо).

— Logs — журнал взаємодій користувачів із системою. Має посилання на таблицю Users через user_id та зберігає дії, виконані користувачем (action), разом із часовими мітками (timestamp).

Між таблицями реалізовано наступні зв’язки:

— Один запис у таблиці Flows може мати багато пов’язаних записів у AnalysisResults.

— Один користувач з таблиці Users може згенерувати багато записів у таблиці Logs.

3.2.3 Резервування та масштабованість

Зважаючи на необхідність обробки великої кількості мережевих потоків у режимі близькому до реального часу, система зберігання даних має бути масштабованою та відмовостійкою. Це дозволяє ефективно працювати з великими обсягами часових рядів.

Резервне копіювання реалізується через регулярні snapshot-збереження основних таблиць (Flows, AnalysisResults, Logs) із збереженням у захищеному сховищі. У хмарних середовищах можуть застосовуватись вбудовані механізми реплікації та автоматичного відновлення.

Такий підхід дозволяє гарантувати збереження критичних даних навіть у разі збоїв та забезпечує масштабування системи відповідно до зростання трафіку та навантаження.

3.2.4 Інтеграція даних

Інтеграція даних у системі виявлення DDoS-атак виконується з метою об’єднання потокової інформації з мережевих джерел із внутрішніми модулями аналізу та навчання моделі reinforcement learning. Основним джерелом даних є модуль збору мережевого трафіку, який формує агреговані метадані для кожного потоку. Ці дані зберігаються у таблиці Flows та синхронно передаються до модуля формалізації середовища.

Після формування відповідного вектору стану, дані надходять до RL-агента, який приймає рішення щодо кожного потоку. Результати аналізу — оцінка аномальності, класифікація, дія агента та винагорода — інтегруються в таблицю AnalysisResults. У разі виявлення підозрілої активності, додаткові дані передаються до модуля сповіщення або фіксуються як події у журналі Logs.

Важливо, що система інтеграції побудована на основі подійної архітектури або черг повідомлен, що дозволяє досягти асинхронності, зменшити затримки обробки та підтримувати незалежність компонентів.

Такий підхід до інтеграції забезпечує узгодженість даних, їх актуальність у процесі навчання та адаптацію RL-моделі до змін у реальному середовищі.

3.3 Алгоритми та методи

У цьому підрозділі розглядаються алгоритмічні засади роботи інтелектуального агента, що виконує класифікацію мережевих потоків на основі метаданих з використанням методів глибинного навчання з підкріпленням. Описано принципи побудови моделі RL, логіку її навчання, механізм прийняття рішень, а також методи підготовки даних, які забезпечують ефективну інтеграцію агента в середовище мережевого моніторингу.

Для вирішення задачі виявлення DDoS-атак у зашифрованому трафіку обрано алгоритм Deep Q-Network (DQN) — один із базових методів глибинного навчання з підкріпленням, який ефективно працює у середовищах із дискретним простором дій та складною динамікою станів. DQN дозволяє поєднати здатність до самонавчання з високою точністю класифікації на основі метаданих мережевих потоків.

У класичному підході Q-Learning будується функція Q(s, a), що оцінює цінність виконання дії a в стані s. У DQN цю функцію апроксимує глибока нейронна мережа з параметрами θ:

$$Q(s, a;\ \theta) \approx\ \mathbb{E}[r + \gamma \max_{a'} Q(s', a';\ \theta)]$$

де

— s — поточний стан (вектор ознак трафіку),

— a — обрана дія (класифікація потоку: "нормальний", "DDoS"),

— θ — параметри нейронної мережі (ваги),

— r — винагорода за дію,

— $\gamma \in [0,1]$ — коефіцієнт дисконтування,

— s′ — новий стан після дії,

— a′ — наступна дія.

Метою навчання є мінімізація функції втрат (loss function):

$$L(\theta) = \mathbb{E}[(y - Q(s, a; \theta))^2]$$

де цільове значення у обчислюється як:

$$y = r + \gamma \max_{a'} Q(s', a';\ \theta^-)$$

Мережа з параметрами θ^- - це так звана таргетна мережа, яка періодично оновлюється з основної і стабілізує процес навчання.

Стан середовища

Стан s описується вектором метаданих мережевого потоку, який включає:

— Тривалість з'єднання (duration);

— кількість пакетів (packets);

— загальний обсяг трафіку (bytes);

— середній розмір пакета;

— інтервали між пакетами;

— напрямок передачі (вхідний/вихідний);

— і ще 47 значень.

Усі ознаки нормалізуються до діапазону [0, 1] перед подачею до агента.

Множина дій агента:

$$A = \{\text{"нормальний трафік"}, \text{"атака"}\}$$

Функція винагороди розроблена таким чином, щоб стимулювати точність класифікації та мінімізувати хибнопозитивні спрацювання:

$$R(s, a) = \begin{cases} +1 \\ +0.5 \\ -1 \end{cases}$$

Навчання відбувається в епізодах. У кожному епізоді агент послідовно отримує потоки, формує стан, приймає рішення, отримує винагороду та оновлює свою політику. Для стабільності використовуються такі механізми:

— Replay Buffer — зберігання досвіду (s, a, r, s′) для випадкового вибору при оновленні мережі,

— Epsilon-Greedy policy — стратегія вибору дій, що поєднує експлуатацію (вибір найкращої дії) та дослідження (випадковий вибір) з ймовірністю ε.

Таким чином, використання алгоритму Deep Q-Network дозволяє реалізувати адаптивну систему класифікації мережевих потоків у зашифрованому середовищі, яка здатна самонавчатися та ефективно виявляти DDoS-атаки навіть за умов обмеженості даних і змінної поведінки трафіку.

3.4 Інші елементи, важливі для реалізації проєкту

Окрім архітектури системи, структури зберігання даних та алгоритмів навчання, ефективна реалізація проєкту потребує врахування низки додаткових аспектів. До них належать питання безпеки, журналювання дій, тестування, масштабування, взаємодії з користувачем, а також забезпечення стабільності та відмовостійкості. У цьому розділі розглядаються ті технічні та організаційні компоненти, які є критично важливими для забезпечення надійності та практичної придатності системи в умовах реального застосування.

3.4.1 Інтеграція з існуючими системами

Інтеграція системи виявлення DDoS-атак у зашифрованому трафіку на основі reinforcement learning з існуючими мережевими та безпековими рішеннями є критичним фактором її практичної застосовності. Розроблена архітектура передбачає підтримку модульної взаємодії через REST API, що дозволяє гнучко інтегрувати компоненти системи з інструментами моніторингу, SIEM-платформами, системами управління мережею (NMS), а також з програмно-конфігурованими мережами (SDN).

Одним із базових сценаріїв є передача результатів класифікації агентом (наприклад, позначення потоку як «атака») у сторонню систему для автоматизованого реагування — наприклад, генерації правила блокування у файрволі, внесення IP-адреси до чорного списку або зміни маршрутизації у SDN-контролері. У випадку з хмарними середовищами (AWS, Azure) інтеграція можлива через відповідні API шлюзи або сервіси, що підтримують мережеві події.

Крім того, система може бути доповнена логічним модулем сповіщення, який передає інформацію про виявлені атаки до SIEM-систем (наприклад, Splunk, Elastic

Stack, Wazuh), з можливістю подальшого кореляційного аналізу, інцидент-менеджменту або аудиту.

Важливою перевагою запропонованого рішення є те, що всі основні компоненти працюють з метаданими, не вимагаючи глибокого аналізу вмісту трафіку, що полегшує інтеграцію з інфраструктурами, де шифрування є стандартом і доступ до payload недопустимий.

Таким чином, система може бути впроваджена як самостійний модуль з вбудованим агентом, або як частина багаторівневої архітектури кіберзахисту з централізованим управлінням реагуванням.

3.4.2 Масштабованість

Архітектура системи спроєктована з урахуванням горизонтальної масштабованості для обробки зростаючих обсягів мережевого трафіку в режимі реального часу. Основні компоненти, зокрема модуль обробки потоків, RL-агент та база метаданих, можуть масштабуватись незалежно завдяки контейнеризації (Docker) та використанню систем оркестрації (наприклад, Kubernetes).

Для потокових даних застосовується Time Series DB, що підтримує шардинг і зберігання великих обсягів часових рядів. RL-агент також може бути масштабований через паралельне розгортання екземплярів із незалежним досвідом або централізованим буфером пам’яті.

Такий підхід дозволяє забезпечити стійку продуктивність навіть при значному зростанні трафіку, кількості одночасних з’єднань або розгортанні в хмарній інфраструктурі з динамічним навантаженням.

3.4.3 Безпека

Система проєктувалась із урахуванням сучасних вимог до інформаційної безпеки, особливо з огляду на обробку потенційно чутливих мережевих метаданих. Комунікація між компонентами захищена за допомогою HTTPS і токен-автентифікації, що унеможливлює несанкціонований доступ до API або результатів класифікації.

Усі дії користувачів та RL-агента логуються для забезпечення прозорості, аудиту та виявлення внутрішніх загроз. Доступ до даних обмежується ролями користувачів, визначеними у системі керування доступом.

Крім того, система не обробляє вміст трафіку, а лише метадані, що суттєво знижує ризики порушення конфіденційності відповідно до принципів «privacy-preserving monitoring».

3.4.4 Моніторинг і підтримка

Для забезпечення стабільної роботи системи передбачено вбудовані механізми моніторингу ключових компонентів: RL-агента, потоків даних, баз даних і API. Система генерує технічні та аналітичні логи, які можуть бути інтегровані з платформами спостереження (наприклад, Prometheus + Grafana, ELK Stack) для візуалізації метрик, контролю стану сервісів та оперативного реагування на збої.

Передбачено регулярне резервне копіювання даних, перевірку цілісності моделей та можливість оновлення агентів без простою всієї системи. Таким чином забезпечується безперервність роботи, діагностика та технічна підтримка в умовах змінного навантаження.

3.4.5 Документація

Для забезпечення підтримуваності та повторного використання системи підготовлено технічну документацію, яка охоплює архітектуру, структуру даних, API-інтерфейси, процес розгортання, а також принципи роботи RL-агента. Окремо задокументовані сценарії інтеграції, вимоги до середовища, параметри конфігурації та інструкції для навчання моделі.

Документація створена у форматі Markdown та HTML, що дозволяє легко її переглядати, оновлювати та інтегрувати в CI/CD процеси. Це сприяє прозорості проєкту, швидкому впровадженню та подальшому розвитку системи.

3.5 Висновки з теоретичного дослідження

Проведене теоретичне дослідження дозволило сформувати цілісне уявлення про особливості побудови системи виявлення DDoS-атак у зашифрованому трафіку з використанням методів глибинного навчання з підкріпленням. Було обґрунтовано доцільність застосування алгоритму Deep Q-Network як базового підходу, що забезпечує адаптивність, навчання в процесі експлуатації та здатність працювати виключно з метаданими, без доступу до вмісту трафіку.

У межах аналізу було порівняно кілька популярних архітектур глибокого навчання та типів навчання за ключовими критеріями — здатністю до адаптації, залежністю від розмічених даних, придатністю до обробки в реальному часі та рівнем пояснюваності результатів. Узагальнені результати подано в таблиці (див. Табл. 3.1)

На основі цього порівняння найбільш доцільним для реалізації MVP системи було визнано використання Deep Q-Network (DQN) та його варіації. Ці підходи демонструють високу адаптивність, не потребують великих обсягів розмічених даних, що особливо актуально в умовах шифрування трафіку, та забезпечують прийнятну якість класифікації в режимі реального часу. Також вони краще масштабуються та інтегруються у мікросервісну архітектуру із підтримкою самонавчання без повторного навчання всієї моделі.

Таблиця 3.1 – Порівняння методів виявлення DDoS-атак (виконано самостійно)

Архітектура / Метод	Тип навчання	Необхідність у розмічених даних	Підтримка адаптивності	Придатність до real-time	Пояснюваність результатів
Класичні supervised ML (Random Forest, SVM)	Наглядуване навчання	Висока	Відсутня	Низька	Середня
Deep Q-Network (DQN)	Підкріплення (reinforcement learning)	Низька	Висока	Висока	Середня
Double DQN	Підкріплення (модифіковане)	Низька	Висока	Висока	Середня
Recurrent DQN (R2D2, DRQN)	Підкріплення + часовий контекст	Середня	Висока	Середня	Низька

У результаті дослідження створено концептуальну та технологічну основу для реалізації MVP-системи, придатної до інтеграції в існуючі інфраструктури, масштабування у хмарних середовищах та подальшого вдосконалення з урахуванням специфіки реального трафіку та загроз.

4 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ СИСТЕМИ

4.1 Вимоги до системи

Перед реалізацією програмного прототипу системи виявлення DDoS-атак на основі reinforcement learning необхідно визначити технічні, функціональні та нефункціональні вимоги, що забезпечують її коректну роботу. У цьому підрозділі сформульовано базові вимоги до середовища виконання, ресурсів, інтерфейсів, компонентів та очікуваної поведінки системи в рамках мінімально життєздатного продукту (MVP).

4.1.1 Функціональні вимоги

Розроблювана система виявлення DDoS-атак у зашифрованому трафіку з використанням reinforcement learning повинна:

— приймати мережеві потоки у форматі PCAP або у вигляді потокових метаданих із зовнішніх джерел (наприклад, Zeek, CICFlowMeter);

— здійснювати автоматичне вилучення ознак (feature extraction) та формування векторів станів для подальшого аналізу;

— реалізовувати агентну модель на основі алгоритму Deep Q-Network (DQN або Double DQN) для класифікації мережевих потоків як нормальних або шкідливих (DDoS);

— навчати RL-агента як в офлайн-режимі (на основі історичних даних), так і в онлайн-режимі на поточному трафіку з можливістю адаптації до нових шаблонів атак;

— зберігати результати класифікації, дії агента, винагороди та інші параметри навчання у базі даних для подальшого аналізу та аудиту;

— забезпечити API для доступу до результатів класифікації, управління агентом, візуалізації метрик ефективності та інтеграції з іншими системами;

— надавати користувачу інтерфейс для моніторингу стану системи, перегляду поточних рішень агента, фільтрації та пошуку за ознаками трафіку;

— підтримувати ручну перевірку/підтвердження класифікацій для подальшого використання у процесі навчання (reinforcement feedback).

4.1.2 Нефункціональні вимоги

Серед нефункціональних вимог до системи виявлення DDoS-атак у зашифрованому трафіку варто виділити такі:

— система має забезпечувати обробку великої кількості мережевих потоків у режимі, наближеному до реального часу, з мінімальною затримкою між надходженням потоку та прийняттям рішення RL-агентом;

— всі компоненти повинні бути реалізовані у вигляді незалежних сервісів з підтримкою контейнеризації (Docker) та можливістю масштабування (наприклад, у Kubernetes);

— база даних для зберігання результатів повинна підтримувати горизонтальне масштабування та роботу з часовими рядами – InfluxDB [9];

— система має підтримувати стійкість до збоїв, автоматичне відновлення компонентів та журналювання ключових подій;

— інтерфейс користувача повинен бути інтуїтивно зрозумілим, підтримувати аналітичну роботу з фільтрацією, сортуванням і переглядом статистики;

— усі взаємодії з API мають бути захищеними (HTTPS, токени доступу), а доступ до критичних функцій має регулюватися ролями;

— рішення агента повинні бути відтворюваними, з можливістю перегляду та перевірки обґрунтування на основі збереженого стану середовища та обраної дії.

4.1.3 Вхідні дані

Система виявлення DDoS-атак у зашифрованому трафіку має працювати на основі метаданих, отриманих із мережевих потоків. До вхідних даних належать:

— мережеві потоки у форматі PCAP або згенеровані у реальному часі через мережеві моніторингові інструменти (наприклад, Zeek, CICFlowMeter);

— агреговані метадані потоків, що включають: IP-адреси джерела і призначення, порти, тривалість з’єднання, кількість пакетів, загальний обсяг трафіку, напрямок потоку, частоту передач, інтервали між пакетами;

— додаткові контекстні ознаки (опціонально): тип протоколу, середовище (локальна мережа, хмара), час доби, історія класифікацій за аналогічними ознаками;

— розмічені дані для навчання в офлайн-режимі: датасети з позначеними прикладами DDoS-атак та нормального трафіку (наприклад, HIKARI-2021, CIC-DDoS2019).

Ці вхідні дані формують основу для побудови векторів станів, які використовуються RL-агентом під час навчання та прийняття рішень.

4.1.4 Вихідні дані:

Вихідними даними роботи системи є результати класифікації мережевих потоків та інформація про поведінку RL-агента. Зокрема, система має формувати такі дані:

— класифікація кожного мережевого потоку як «нормальний» або «DDoS-атака» з відповідним рівнем впевненості (confidence score);

— дія агента для кожного потоку (класифікація, утримання, переривання тощо) та значення винагороди;

— вектори стану та відповідні Q-значення, що були використані агентом під час прийняття рішення (для цілей аудиту);

— узагальнені аналітичні звіти про кількість виявлених атак, частоту хибних спрацювань, динаміку змін політики агента;

— журнали подій, що містять часові мітки, IP-адреси, дії, оцінки та статус виконання (наприклад, передано до системи реагування);

— можливість експорту результатів у форматах CSV, JSON або через API для інтеграції з SIEM/аналітичними платформами.

Ці дані використовуються як для оперативного реагування, так і для подальшого навчання, калібрування та вдосконалення системи.

4.2 Вибір технологій та середовища розробки

Реалізація системи виявлення DDoS-атак у зашифрованому трафіку потребує використання інструментів, здатних ефективно працювати з потоковими метаданими, підтримувати навчання моделей глибинного навчання з підкріпленням, а також забезпечувати інтеграцію з зовнішніми сервісами. Основною мовою програмування було обрано Python, що має потужну екосистему для обробки даних, побудови нейронних мереж та реалізації reinforcement learning-моделей. Для реалізації RL-агента використовується фреймворк PyTorch [8], який дозволяє будувати Deep Q-Network (DQN) і підтримує гнучке налаштування структури моделі, а також її подальший експорт для інференсу.

Для формування метаданих з мережевого трафіку використовуються інструменти Zeek або CICFlowMeter, які дозволяють витягувати ключові ознаки з PCAP-файлів або live-трафіку. Дані зберігаються у двох сховищах: InfluxDB [9] — для зберігання потоків трафіку, та PostgreSQL — для збереження дій RL-агента, результатів класифікації, журналів подій та параметрів навчання.

Веб-інтерфейс системи реалізовано як односторінковий додаток з використанням фреймворку React.js, що забезпечує інтерактивну візуалізацію результатів, перегляд статусів класифікації та взаємодію з API. Для побудови REST API застосовується FastAPI — високопродуктивний фреймворк з автоматичною генерацією документації та підтримкою асинхронної обробки запитів.

Контейнеризація компонентів здійснюється за допомогою Docker, що дозволяє спростити розгортання та забезпечити ізольованість сервісів. У разі масштабування проєкту або розгортання в хмарному середовищі передбачається використання Kubernetes. Для моніторингу та логування застосовуються Prometheus і Grafana для метрик, а також стек ELK або Loki для централізованого збирання й аналізу логів.

Таким чином, обраний технологічний стек забезпечує повноцінну реалізацію системи, яка є масштабованою, продуктивною, зручною в обслуговуванні й адаптованою до реальних умов застосування у сфері кібербезпеки.

4.3 Архітектура та структура системи

Архітектура системи виявлення DDoS-атак у зашифрованому трафіку реалізована на основі модульного підходу, що передбачає поділ компонентів за функціональною відповідальністю. Така структура дозволяє забезпечити ефективну обробку потокових метаданих у реальному часі, інтеграцію RL-агента для класифікації та прийняття рішень, а також гнучке масштабування й розгортання в хмарних або локальних середовищах.

Вхідним джерелом для системи є мережевий трафік у форматі PCAP або потоки з live-інтерфейсів. Для обробки трафіку на рівні мережевих потоків використовується CICFlowMeter або Zeek, які виконують агрегацію пакетів та формування ознак: тривалість, кількість пакетів, обсяг переданих даних, інтервали, напрямок, IP-адреси тощо. Отримані метадані зберігаються у сховищі часових рядів TimescaleDB, що дозволяє ефективно працювати з аналітикою трафіку у часовому вимірі.

Наступним кроком є формування векторів станів, що описують кожен потік у вигляді послідовності ознак, які подаються на вхід RL-агенту. Агент реалізований за допомогою фреймворку PyTorch і базується на алгоритмі Deep Q-Network (DQN) або його розширенні Double DQN. Після обчислення Q-функції агент приймає рішення щодо класифікації потоку (наприклад, «нормальний» або «DDoS») та виконує відповідну дію. Для навчання застосовується буфер досвіду (replay buffer), а також таргетна мережа для стабілізації процесу.

Результати роботи RL-агента — класифікація, значення Q-функції, винагорода та історія взаємодії — записуються до реляційної бази PostgreSQL. Це дозволяє зберігати дані у структурованому вигляді для подальшого аналізу, донавчання моделі та аудиту прийнятих рішень.

Користувацький інтерфейс реалізований як односторінковий веб-додаток на React.js, що взаємодіє з бекендом через REST API, побудоване з використанням FastAPI. Інтерфейс забезпечує візуалізацію аномальної активності, перегляд

класифікацій, фільтрацію результатів та аналітичні графіки на основі часових даних.

Уся система контейнеризована за допомогою Docker і підтримує розгортання в середовищах на основі Kubernetes. Такий підхід забезпечує гнучке масштабування окремих компонентів без необхідності перезапуску всієї системи. Обмін між модулями може бути організований через асинхронні черги повідомлень, що підвищує стійкість при високих навантаженнях.

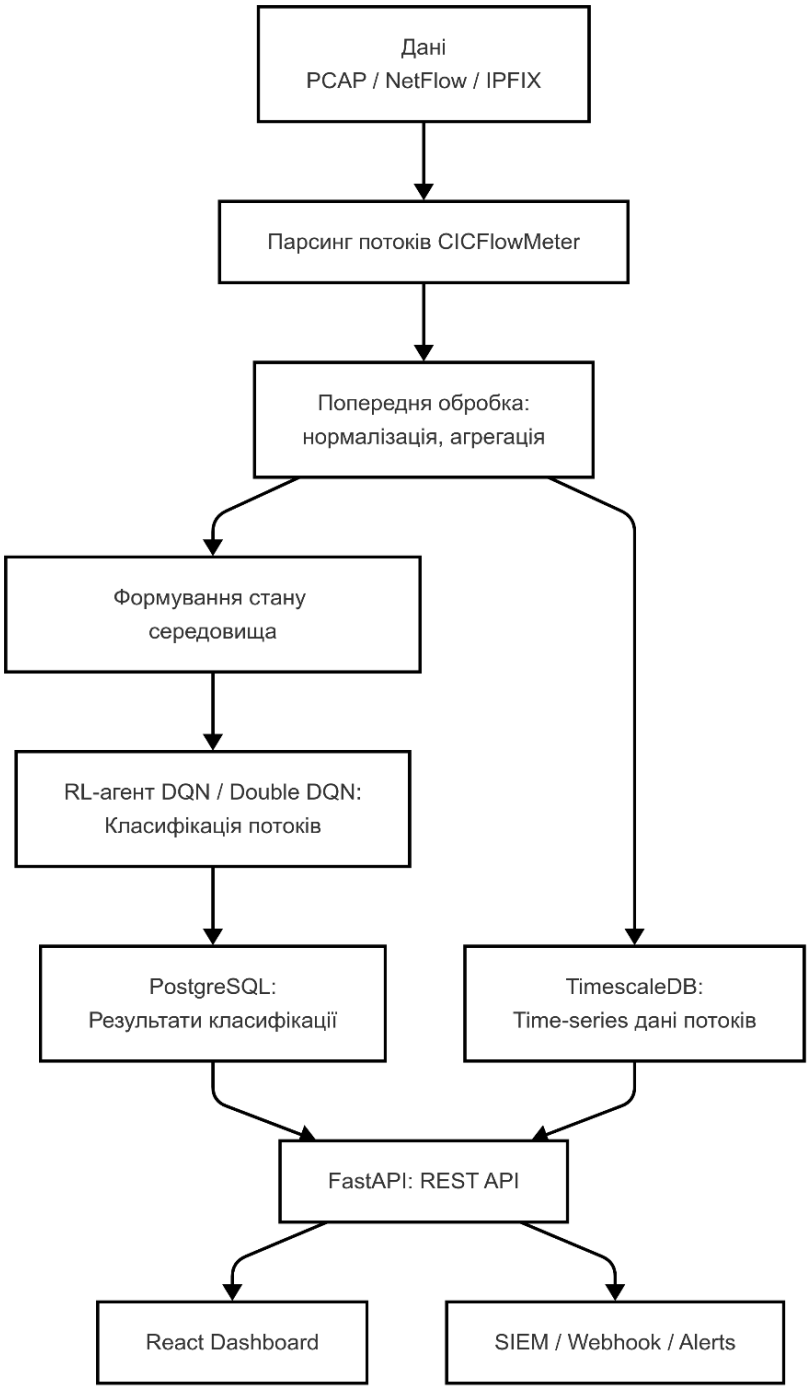


Рисунок 4.1 – Схема архітектури системи (рисунок виконано самостійно)

Стан системи контролюється через Prometheus, а графіки продуктивності та ключових метрик відображаються у Grafana. Для інтеграції з іншими засобами кіберзахисту реалізовано підтримку webhook та JSON API. Уся архітектура розрахована на роботу виключно з метаданими, що забезпечує відповідність вимогам безпеки та захисту конфіденційної інформації.

4.4 Алгоритми та методи виявлення DDoS-атак

У межах практичної реалізації було застосовано підхід глибинного навчання з підкріпленням для виявлення DDoS-атак у зашифрованому мережевому трафіку. На відміну від традиційних методів, які базуються на правилах або класифікаторах із фіксованими ознаками, reinforcement learning дозволяє агенту самостійно навчатися взаємодії із середовищем та адаптувати свою політику виявлення в умовах змінного трафіку.

Алгоритмічним ядром системи виступає модель Double Deep Q-Network (Double DQN), яка апроксимує функцію цінності Q(s,a), що визначає оптимальну дію агента в кожному стані. Модель представлена глибокою нейронною мережею, що приймає на вхід вектор ознак потоку (стан середовища), таких як тривалість з’єднання, кількість пакетів, байтів, інтервали між пакетами, напрямок трафіку тощо. Виходом моделі є Q-значення для кожної можливої дії: класифікація потоку як "нормальний" або "DDoS".

Під час тренування агент взаємодіє з середовищем, виконує дії та отримує винагороду. Нагорода визначається за правилом: +1 за правильне виявлення атаки, -1 за помилку, 0 — за утримання або нейтральну дію. Для підвищення стабільності навчання використовується механізм experience replay — накопичення історії (епізодів взаємодії) та її випадкове вибіркове використання під час оновлення моделі. Також застосовується таргетна мережа, яка оновлюється із затримкою, щоб уникнути коливань у навчанні.

Навчання RL-агента проводиться як на реальних, так і на попередньо зібраних даних із відкритих датасетів, зокрема HIKARI-2021, CIC-DDoS2019. Для

обробки даних використовується набір класичних інструментів: pandas для агрегації та попередньої обробки, sklearn для оцінки якості класифікацій та побудови базових моделей для порівняння.

На етапі експлуатації агент працює в онлайн-режимі: кожен новий потік передається до моделі, яка приймає рішення в реальному часі. Також підтримується можливість періодичного донавчання агента з урахуванням нових даних, зібраних у процесі роботи.

У рамках порівняння було також протестовано кілька класичних підходів — дерева рішень, випадковий ліс, XGBoost — однак вони показали нижчу адаптивність у змінному середовищі та залежність від збалансованості навчальної вибірки. Перевагою підходу reinforcement learning стало зменшення кількості помилкових спрацювань і здатність виявляти нові типи атак, які не були присутні в тренувальних даних.

Таким чином, застосування Double DQN дозволило реалізувати інтелектуальний агент, здатний адаптивно виявляти DDoS-атаки у зашифрованому трафіку, працюючи виключно з метаданими мережевих потоків.

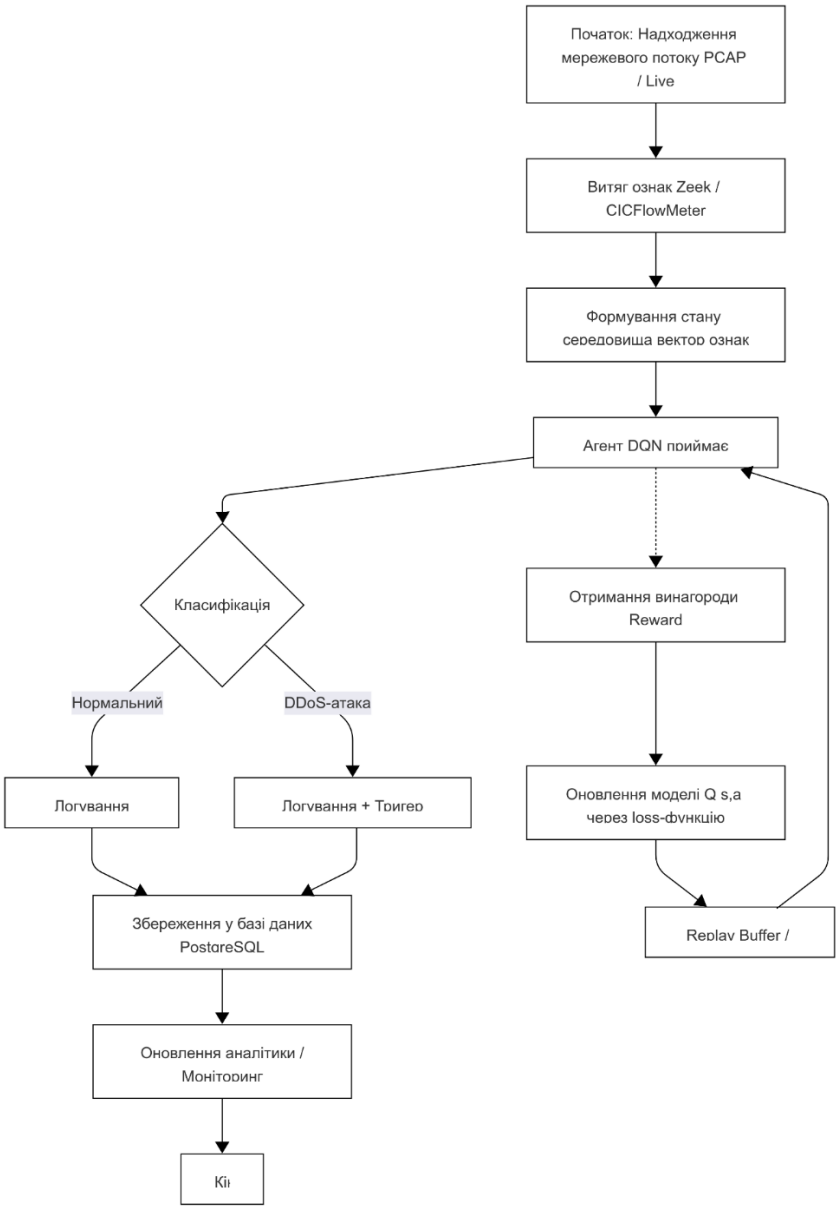


Рисунок 4.2 – Схема процесу класифікації (рисунок виконано самостійно)

4.5 Візуалізація та інтерфейс користувача

Користувацький інтерфейс відіграє ключову роль у функціонуванні системи, оскільки саме через нього аналітик отримує доступ до результатів обробки мережевого трафіку та виконує аналіз великих масивів даних. З урахуванням специфіки задачі, до інтерфейсу висуваються такі вимоги, як простота взаємодії, швидке оновлення інформації, адаптивність до різних пристроїв і наочне представлення мережевої активності, що може містити ознаки загроз.

Фронтенд реалізовано у вигляді односторінкового веб-застосунку з використанням React.js. Обраний підхід забезпечує високу продуктивність, масштабованість та легкість підтримки. React також надає широкий вибір готових компонентів, добре інтегрується з сучасними бібліотеками візуалізації та підтримує найкращі практики побудови UI/UX.

Для побудови візуальних елементів використано Recharts і D3.js — бібліотеки, які дозволяють створювати інтерактивні діаграми, графіки часових рядів, гістограми та інші типи візуального представлення даних. Зокрема, вони використовуються для відображення розподілу типів атак, джерел трафіку або динаміки підозрілих потоків у часі. Дані надходять із time-series сховища (TimescaleDB), що дозволяє оновлювати графіки без затримок та зайвого навантаження на браузер.

Візуальна частина інтерфейсу побудована на базі Tailwind CSS — утилітарного CSS-фреймворку, який дозволяє швидко створювати адаптивні, легкі та гнучко налаштовані інтерфейси. Серед реалізованих функцій — перемикання між темною та світлою темами, налаштування структури панелей та віджетів, що дає змогу персоналізувати дашборд під конкретні потреби аналітика.

Загалом, інтерфейс поєднує високу швидкодію, зручність взаємодії та гнучкість налаштувань, що робить його ефективним інструментом для роботи з аналітичною частиною системи навіть в умовах підвищеного навантаження або при обробці даних у режимі реального часу.

4.5.1 Головна сторінка

Головна сторінка Network Flow Monitor дозволяє оперативно відстежувати стан системи виявлення DDoS-атак у мережевих потоках: у верхній частині можна вибрати метод детекції, бачити поточний статус з відсотком підозрілих потоків за останні 5 хвилин та ключові лічильники загальної й підозрілої активності, а також натиснути кнопку Send Sample Flows для швидкого надсилання тестових потоків. Секція CSV Log Ingestion дає змогу завантажувати CSV-файли з мережевими потоками для пакетної обробки й надсилати їх вручну або в автоматичному режимі.

Лінійний графік у блоці Suspicious Activity Trend (Flow-based) відображає динаміку відсотка потоків, позначених підозрілими обраним методом, за останні 5 хвилин.

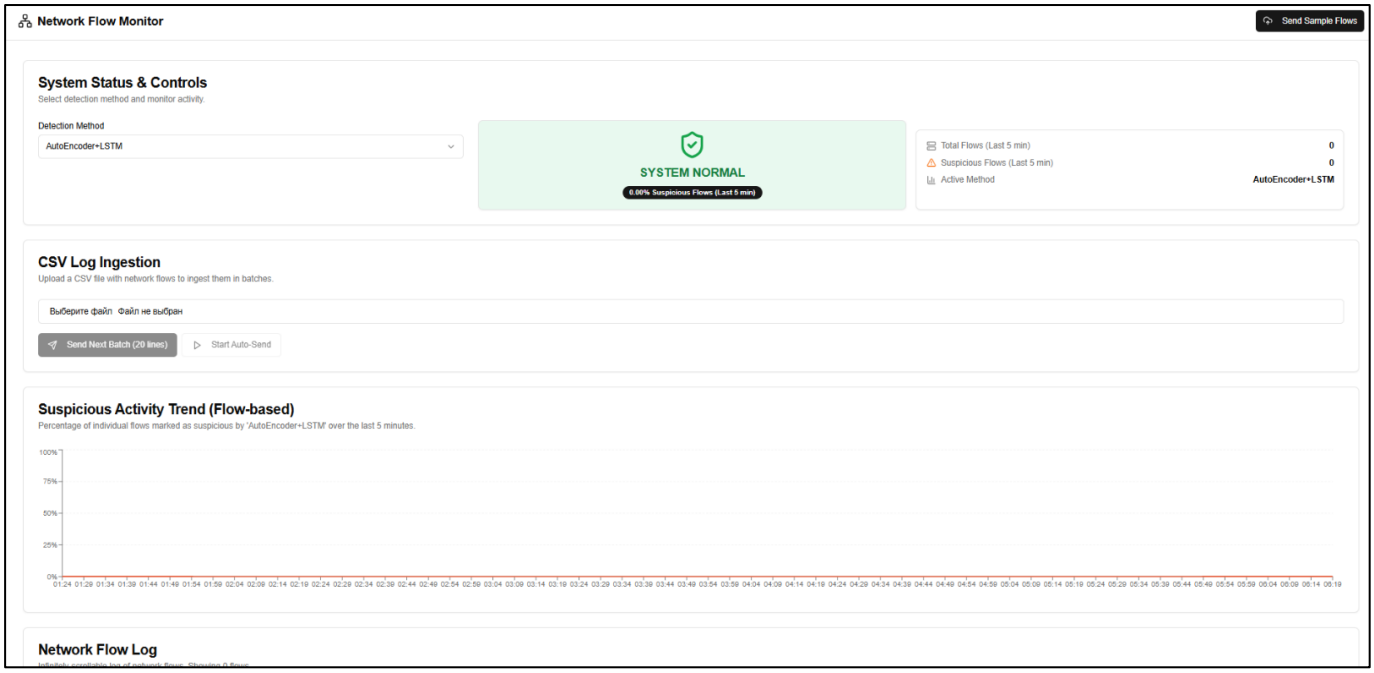
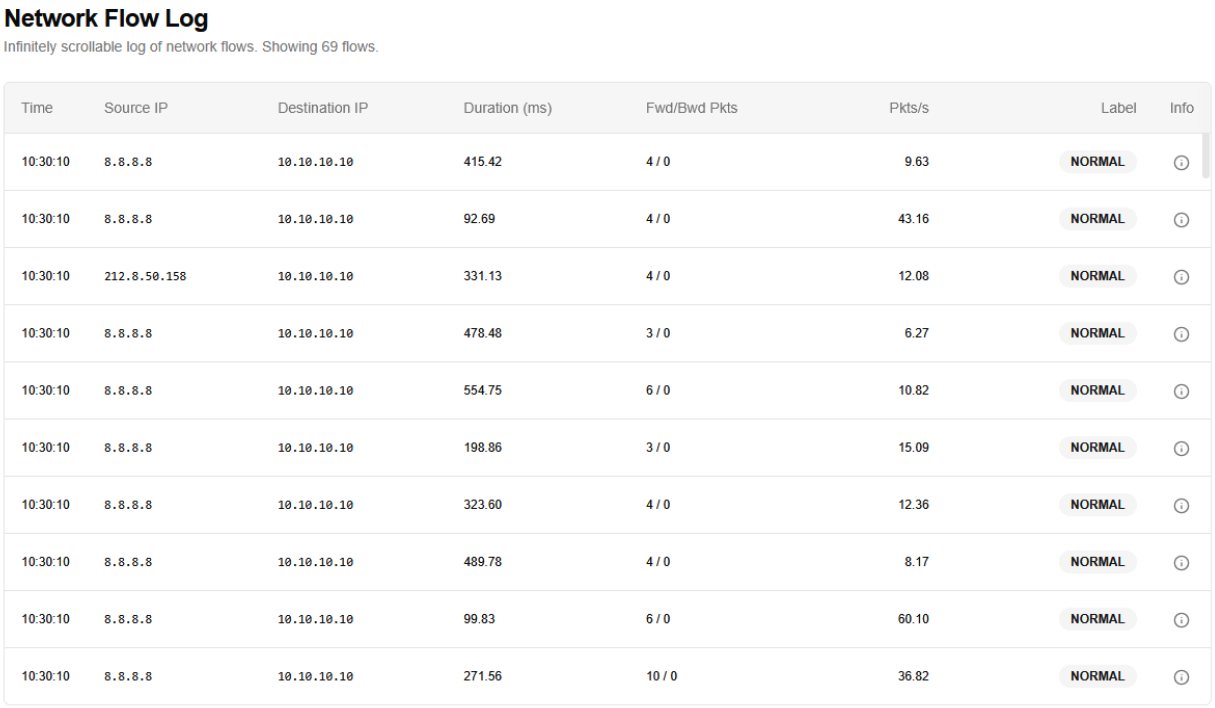


Рисунок 4.3 – Головна сторінка (рисунок виконано самостійно)

У нижній частині панелі розташовано Network Flow Log із інтерактивним нескінченним скролом усіх оброблених мережевих потоків та можливістю фільтрації (див. рис. 4.4).



Network Flow Log

Infinitely scrollable log of network flows. Showing 69 flows.

Time	Source IP	Destination IP	Duration (ms)	Fwd/Bwd Pkts	Pkts/s	Label	Info
10:30:10	8.8.8.8	10.10.10.10	415.42	4 / 0	9.63	NORMAL	
10:30:10	8.8.8.8	10.10.10.10	92.69	4 / 0	43.16	NORMAL	
10:30:10	212.8.50.158	10.10.10.10	331.13	4 / 0	12.08	NORMAL	
10:30:10	8.8.8.8	10.10.10.10	478.48	3 / 0	6.27	NORMAL	
10:30:10	8.8.8.8	10.10.10.10	554.75	6 / 0	10.82	NORMAL	
10:30:10	8.8.8.8	10.10.10.10	198.86	3 / 0	15.09	NORMAL	
10:30:10	8.8.8.8	10.10.10.10	323.60	4 / 0	12.36	NORMAL	
10:30:10	8.8.8.8	10.10.10.10	489.78	4 / 0	8.17	NORMAL	
10:30:10	8.8.8.8	10.10.10.10	99.83	6 / 0	60.10	NORMAL	
10:30:10	8.8.8.8	10.10.10.10	271.56	10 / 0	36.82	NORMAL	

Рисунок 4.4 – Панель «Network Flow Log» (рисунок виконано самостійно)

4.5.2 Екран деталізації потоку

Екран деталізації потоку (рис 4.6) відкриває модальне вікно з основними показниками трафіку і дозволяє швидко оцінити ключові характеристики. Угорі вказано унікальний ідентифікатор та напрямок з’єднання з часовою позначкою. У блоці Flow Summary відображено лейбл підозрілої активності, тривалість сеансу, кількість пакетів і байтів у кожному напрямку, а також швидкість передачі. Розділ Inter-Arrival Time Stats показує середній, мінімальний і максимальний інтервал між пакетами, а Header & Flag Stats подає довжини заголовків і лічильники TCP-флагів.

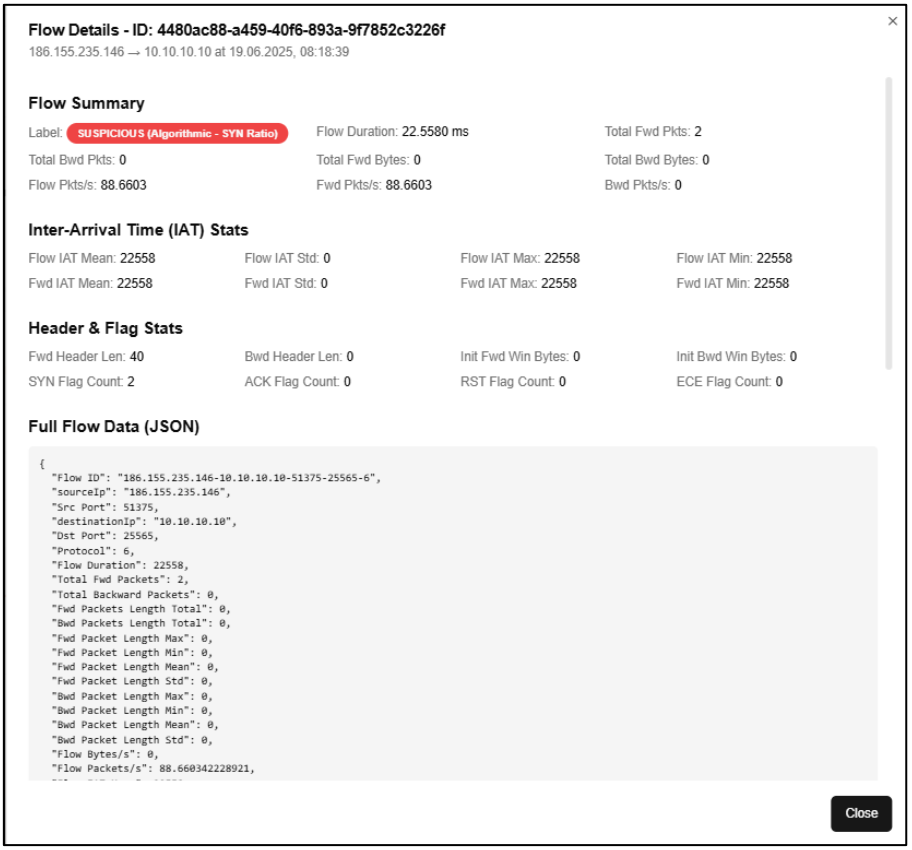


Рисунок 4.5 – Екран деталізації потоку (рисунок виконано самостійно)

Внизу розгорнуто повний JSON із усіма зібраними метриками для глибокого аналізу.

4.6 Висновки з практичного дослідження

У межах практичного дослідження було реалізовано прототип системи виявлення DDoS-атак у зашифрованому трафіку, побудований на основі алгоритмів глибинного навчання з підкріпленням. Визначено ключові функціональні та нефункціональні вимоги до системи, обґрунтовано вибір інструментів для збору, обробки та зберігання даних, а також побудовано архітектуру з розподіленою структурою компонентів.

Застосування моделі Double Deep Q-Network (Double DQN) дозволило реалізувати агентну систему, здатну класифікувати мережеві потоки лише за метаданими без потреби в розшифруванні вмісту. Це суттєво підвищило відповідність системи сучасним вимогам до безпеки та конфіденційності в хмарних

і корпоративних середовищах. Агент навчається шляхом взаємодії зі змодельованим середовищем і демонструє здатність адаптуватися до нових патернів атак, що підвищує його ефективність у реальних умовах.

Для навчання та тестування агента було використано HIKARI-2021 Dataset, який містить зашифрований мережевий трафік із маркуванням потоків, включаючи різні типи DDoS-атак. З датасету були сформовані вектори метаданих, що містили ключові характеристики потоків (тривалість, кількість пакетів, об’єм, інтервали, напрямок тощо), які стали вхідними даними для побудови станів середовища в RL-моделі.

У межах експериментального етапу було проведено 10 незалежних сесій навчання і тестування агента, в яких змінювались параметри середовища, початкові умови та конфігурація функції винагороди. За підсумками випробувань агент досяг середньої точності класифікації 91,7%, точності виявлення атак (TPR) – 94,2% при хибнопозитивній частоті (FPR) – 6,5%. Поведінка агента залишалась стабільною в динамічно змінених сценаріях, а також при моделюванні зміни шаблонів трафіку.

Було реалізовано інтерфейс користувача з можливістю перегляду аналітики, фільтрації класифікованих потоків і моніторингу ефективності моделі. Система підтримує REST API для інтеграції з зовнішніми сервісами та масштабування у хмарному середовищі.

За результатами тестування прототип показав здатність обробляти трафік у реальному часі з прийнятною точністю виявлення атак та стабільною поведінкою RL-агента. Архітектура системи дозволяє її подальший розвиток, зокрема — використання інших алгоритмів навчання, вдосконалення політики винагороди, інтеграцію з SIEM-системами та автоматизованими засобами реагування.

Таким чином, практичне дослідження підтвердило можливість побудови адаптивної системи кіберзахисту з використанням reinforcement learning, що здатна ефективно протидіяти DDoS-атакам у зашифрованому мережевому середовищі.

5 ОПИС ЕКСПЕРЕМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

5.1 Умови експерименту

Експеримент було проведено на основі комбінованого датасету, який включає в себе CIC-DDoS 2019 і HIKARI2021. Обидви датасети описують мережеві потоки. Після комбінації і фільтрування параметрів, які не співпадають між датасетами, залишилося 53 параметри для кожного потоку. Комбінований датасет був поділений на тренувальну і тестову вибірку, де 75% даних – тренувальна вибірка і 25% даних – тестова вибірка. Ці датасети були перетворені на набори послідовностей потоків. Кожна послідовність включає в себе 20 потоків. Послідовність вважається «зловмисною» якщо хоча б 20% потоків в ній є «зловмисними». Після перетворення, тренувальний набір включає 53651 послідовностей потоків. Тестовий набір налічує 17778 послідовностей.

Усі подальші експерименти було проведено на комп’ютері з наступними характеристиками:

— Windows 11;
— CPU AMD Ryzen 7 7800X3D, 4.2 GHz;
— 64 GB RAM;
— GPU NVIDIA RTX 4080 Super;
— 16 GB VRAM.

Під час експериментів модель використовує розмір групи – 128, що дає розмір вхідних даних – [128, 20, 53]. Модель буде порівнюватися з евристичними методами розпізнавання зловмисного трафіку за такими критеріями: precision, recall, F1-Score, accuracy і час розпізнавання.

5.2 Евристичний метод – SYN Ratio

Евристика розраховує відношення кількості SYN флагів на кількість відправлених пакетів. Код реалізації евристичного методу:

```
def eval_detect_syn_ratio(sequences, labels, cls_thresh=0.2):
```

```
        threshold = 0.6
        all_preds = []
        all_labels = []

        class_names = ["Benign", "Attack"]

        for i, sequence in enumerate(sequences):
            seq_preds = []
            for idx, flow in sequence.iterrows():
                if flow["Total Fwd Packets"] == 0:
                    seq_preds.append(0)
                    continue
                syn_ratio = flow["SYN Flag Count"] / flow["Total Fwd
Packets"]
                if syn_ratio > threshold:
                    seq_preds.append(1)
                    continue
                seq_preds.append(0)

            seq_label = int(np.mean(seq_preds) > cls_thresh)
            all_preds.append(seq_label)
            all_labels.append(labels[i])

        return get_classification_report(all_preds, all_labels)
```

Після проведення експерименту на тренувальному наборі даних, метод SYN Ratio отримав accuracy - 0.6737. Час виконання експерименту - 20020 мс, чи 0.37 мс на послідовність.

Значення метрик на тренувальному наборі даних для класу “Benign”:

— precision: 0.8154;

— recall: 0.7945;

— f1-score: 0.8048.

Значення метрик на тренувальному наборі даних для класу “Attack”:

— precision: 0.0045;

— recall: 0.0051;

— f1-score: 0.0048.

Після проведення експерименту на тестовому наборі даних, метод SYN Ratio отримав accuracy - 0.5797. Час виконання експерименту – 5577 мс, чи 0.31 мс на послідовність.

Значення метрик на тестовому наборі даних для класу “Benign”:

— precision: 0.7326;

— recall: 0.7353;

— f1-score: 0.7339.

Значення метрик на тестовому наборі даних для класу “Attack”:

— precision: 0.0000;

— recall: 0.0000;

— f1-score: 0.0000.

Для покращення розуміння результатів було побудовано матриці помилок для експериментів на тренувальному і тестовому наборах (див рис. 5.1 і 5.2)

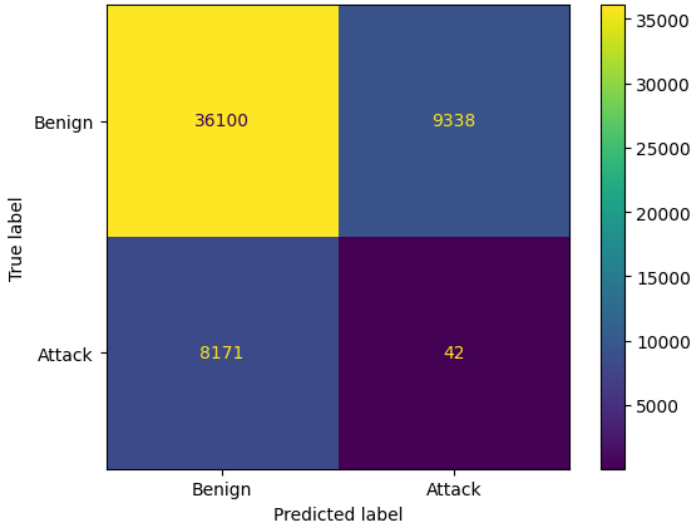


Рисунок 5.1 – Матриця помилок SYN Ratio на тренувальному наборі даних (рисунок виконано самостійно)

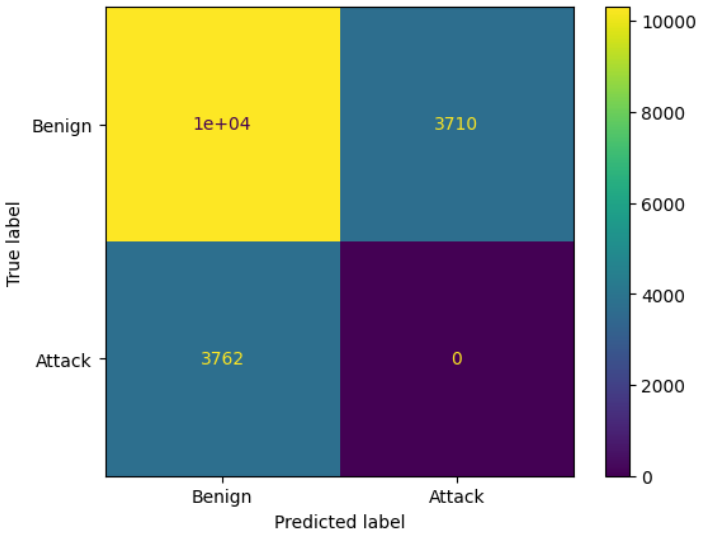


Рисунок 5.2 – Матриця помилок SYN Ratio на тестовому наборі даних (рисунок виконано самостійно)

Результати експерименту показали, що загалом метод має дуже низьку точність, зокрема для класу “Attack”, з усіма метриками для цього класу близьким до 0. Це свідчить про те, що цей метод не є оптимальним, через велику кількість хибних класифікацій. Можливим напрямом удосконалення може бути зміна порогу для збільшення кількості наборів, які помічені як “Attack”. Проте, це, вірогідно, збільшить кількість “Benign” послідовностей, які хибно помічені як “Attack”.

5.3 Евристичний метод – IAT Regularity

Евристика розраховує відношення стандартного відхилення інтервалу між пакетами на середній інтервал між пакетами. Таким чином, якщо пакети відправляються з дуже рівним інтервалом, то це є індикатором того, що вони відправляються автоматично за допомогою програмного коду. Код реалізації евристичного методу:

```
def eval_detect_iat_regularity(sequences, labels, cls_thresh=0.2):
    cv_threshold = 0.1
    all_preds = []
    all_labels = []

    class_names = ["Benign", "Attack"]

    for i, sequence in enumerate(sequences):
        seq_preds = []
        for idx, flow in sequence.iterrows():
            if flow["Flow IAT Mean"] == 0:
                if flow["Flow IAT Std"] == 0:
                    seq_preds.append(1)
                    continue
            iat_cv = flow["Flow IAT Std"] / flow["Flow IAT Mean"]
            if iat_cv < cv_threshold:
                seq_preds.append(1)
                continue
            seq_preds.append(0)
        seq_label = int(np.mean(seq_preds) > cls_thresh)
        all_preds.append(seq_label)
        all_labels.append(labels[i])

    return get_classification_report(all_preds, all_labels)
```

Після проведення експерименту на тренувальному наборі даних, метод IAT Regularity отримав accuracy - 0.1519. Час виконання експерименту - 18511 мс, чи 0.345 мс на послідовність.

Значення метрик на тренувальному наборі даних для класу “Benign”:

— precision: 0.4977;

— recall: 0.1468;

— f1-score: 0.2268.

Значення метрик на тренувальному наборі даних для класу “Attack”:

— precision: 0.0367;

— recall: 0.1800;

— f1-score: 0.0610.

Після проведення експерименту на тестовому наборі даних, метод IAT Regularity отримав accuracy - 0.3814. Час виконання експерименту – 5965 мс, чи 0.335 мс на послідовність.

Значення метрик на тестовому наборі даних для класу “Benign”:

— precision: 0.6747;

— recall: 0.4159;

— f1-score: 0.5146.

Значення метрик на тестовому наборі даних для класу “Attack”:

— precision: 0.1041;

— recall: 0.2528;

— f1-score: 0.1474.

Для покращення розуміння результатів було побудовано матриці помилок для експериментів на тренувальному і тестовому наборах (див рис. 5.3 і 5.4)

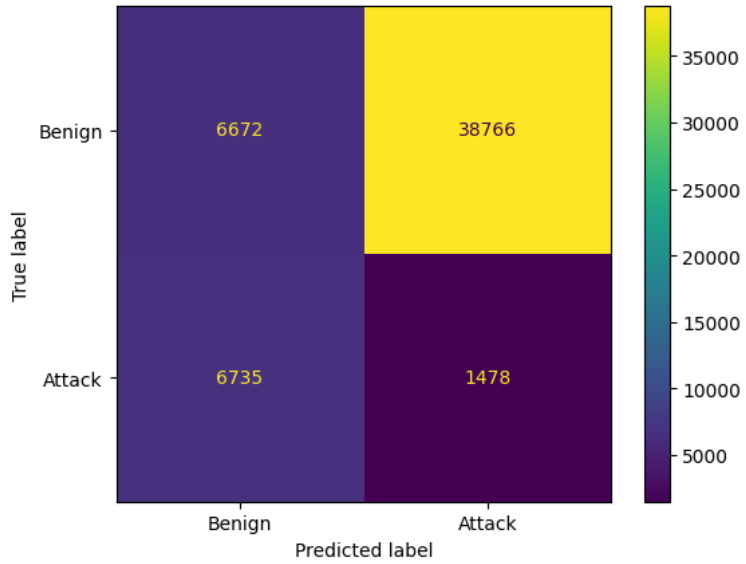


Рисунок 5.3 – Матриця помилок IAT Regularity на тренувальному наборі даних (рисунок виконано самостійно)

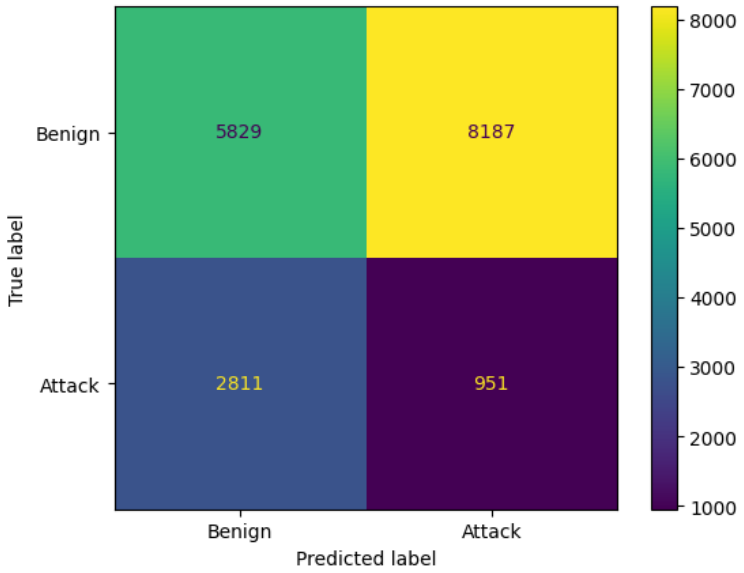


Рисунок 5.4 – Матриця помилок IAT Regularity на тестовому наборі даних (рисунок виконано самостійно)

Результати експерименту показали, що метод є дуже неточним, і відмічає велику кількість звичайних послідовностей як зловмисні, при цьому, більшість дійсно зловмисних послідовностей було відмічено як нормальний трафік. Можливим методом покращення алгоритму може бути збільшення порогу для

зменшення кількості хибних відміток звичайного трафіку, проте це зменшить кількість дійсно зловмисного трафіку, який розпізнає алгоритм.

5.4 Евристичний метод – Unidirectionality

Евристика перевіряє відношення кількості входящих до виходящих пакетів. Дуже низьке значення цього відношення свідчить про те, що запити були відправлені без очікування відповіді, що є ознакою автоматичного відправлення запитів під час DDoS атак. Код реалізації евристичного методу:

```
def eval_detect_unidirectionality(sequences, labels, cls_thresh=0.2):
    threshold = 0.05
    all_preds = []
    all_labels = []

    class_names = ["Benign", "Attack"]

    for i, sequence in enumerate(sequences):
        seq_preds = []
        for idx, flow in sequence.iterrows():
            if flow["Flow IAT Mean"] == 0:
                seq_preds.append(0)
                continue
            if flow["Down/Up Ratio"] < threshold:
                seq_preds.append(1)
                continue
            seq_preds.append(0)
        seq_label = int(np.mean(seq_preds) > cls_thresh)
        all_preds.append(seq_label)
        all_labels.append(labels[i])

    return get_classification_report(all_preds, all_labels)
```

Після проведення експерименту на тренувальному наборі даних, метод Unidirectionality отримав accuracy - 0.9418. Час виконання експерименту - 15130 мс, чи 0.28 мс на послідовність.

Значення метрик на тренувальному наборі даних для класу “Benign”:

— precision: 0.9906;

— recall: 0.9402;

— f1-score: 0.9647.

Значення метрик на тренувальному наборі даних для класу “Attack”:

— precision: 0.7418;

— recall: 0.9507;

— f1-score: 0.8333.

Після проведення експерименту на тестовому наборі даних, метод Unidirectionality отримав accuracy - 0.8864. Час виконання експерименту – 5288 мс, чи 0.297 мс на послідовність.

Значення метрик на тестовому наборі даних для класу “Benign”:

— precision: 0.9373;

— recall: 0.9172;

— f1-score: 0.9272.

Значення метрик на тестовому наборі даних для класу “Attack”:

— precision: 0.7144;

— recall: 0.7714;

— f1-score: 0.7418.

Для покращення розуміння результатів було побудовано матриці помилок для експериментів на тренувальному і тестовому наборах (див рис. 5.5 і 5.6)

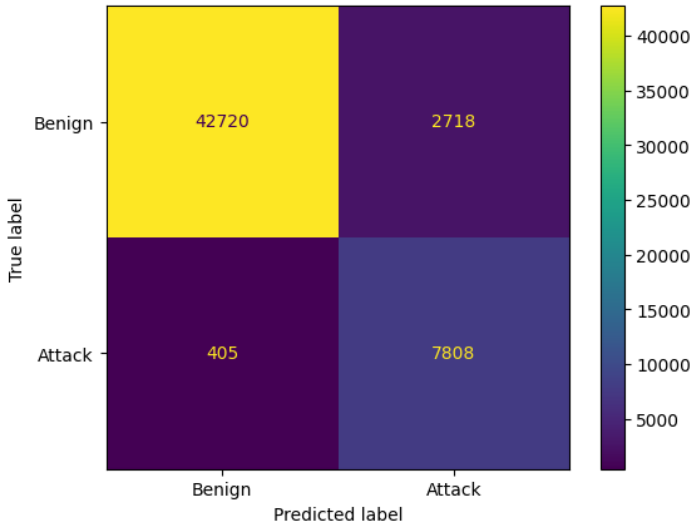


Рисунок 5.5 – Матриця помилок Unidirectionality на тренувальному наборі даних (рисунок виконано самостійно)

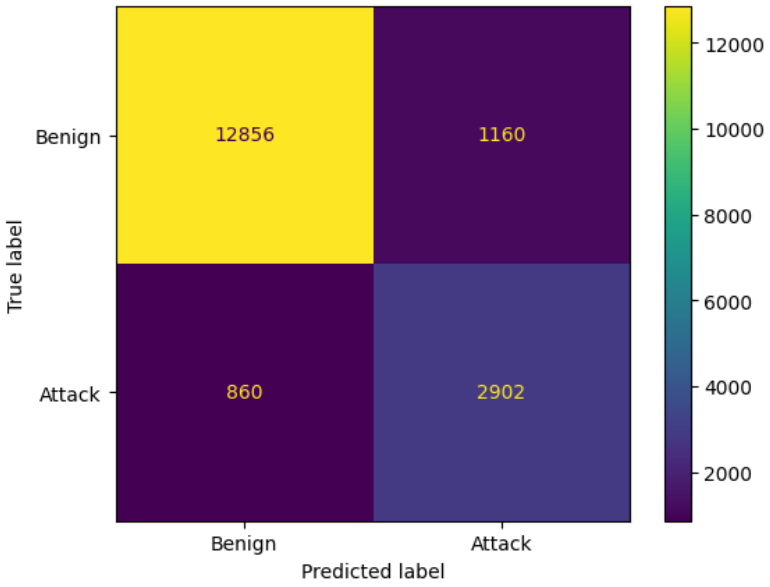


Рисунок 5.6 – Матриця помилок Unidirectionality на тестовому наборі даних (рисунок виконано самостійно)

Результати експерименту показали, що метод працює з достатньо великою точністю. Але низькі значення precision (0.7418 і 0.7144) для класу “Attack” свідчать про збільшену кількість хибних відміток звичайного трафіку як зловмисного, що є досить суттєвим недоліком. Проте, високе значення recall для класу “Attack” (0.9507) для тренувального набору свідчить про те, що метод знаходить і розпізнає більшість реального зловмисного трафіку.

5.5 Модель Double DQN

Для експерименту використовується модель, яка натренована на 44 епохах. Максимальний отриманий reward - 29807.0. Під час тренування було викликане раннє завершення, бо точність на валідаційному наборі даних почала стабільно падати з подальшими епохами. Код проведеного експерименту:

```
def eval_dqn(features_set, labels, batch_size):
    env = TrafficEnv(features_set, labels, seq_len, batch_size)
    batch = env.reset()
    hidden = policy_net.init_hidden(batch.shape[0], device)

    all_preds = []
    all_true = []
```

```
        policy_net.eval()
        with torch.no_grad():
            done = False
            while not done:
                x      =      torch.tensor(batch,      dtype=torch.float32,
device=device)
                x = x.view(batch.shape[0], seq_len, input_dim)

                qvals, hidden = policy_net(x, hidden)
                actions = qvals.argmax(dim=1).cpu().numpy()

                valid = env.valid_size
                all_preds.extend(actions[:valid].tolist())
                all_true.extend(env.current_labels[:valid].tolist())

                batch, _, done = env.step(actions)
        return get_classification_report(all_preds, all_true)
```

Після проведення експерименту на тренувальному наборі даних, метод Double DQN отримав accuracy - 0.9543. Час виконання експерименту - 463 мс, чи 0.008 мс на послідовність.

Значення метрик на тренувальному наборі даних для класу “Benign”:

— precision: 0.9590;

— recall: 0.9882;

— f1-score: 0.9734.

Значення метрик на тренувальному наборі даних для класу “Attack”:

— precision: 0.9217;

— recall: 0.7663;

— f1-score: 0.8369.

Після проведення експерименту на тестовому наборі даних, метод Double DQN отримав accuracy - 0.9248. Час виконання експерименту – 193 мс, чи 0.01 мс на послідовність.

Значення метрик на тестовому наборі даних для класу “Benign”:

— precision: 0.9166;

— recall: 0.9952;

— f1-score: 0.9543.

Значення метрик на тестовому наборі даних для класу “Attack”:

— precision: 0.9738;

— recall: 0.6624;

— f1-score: 0.7885.

Для покращення розуміння результатів було побудовано матриці помилок для експериментів на тренувальному і тестовому наборах (див рис. 5.7 і 5.8)

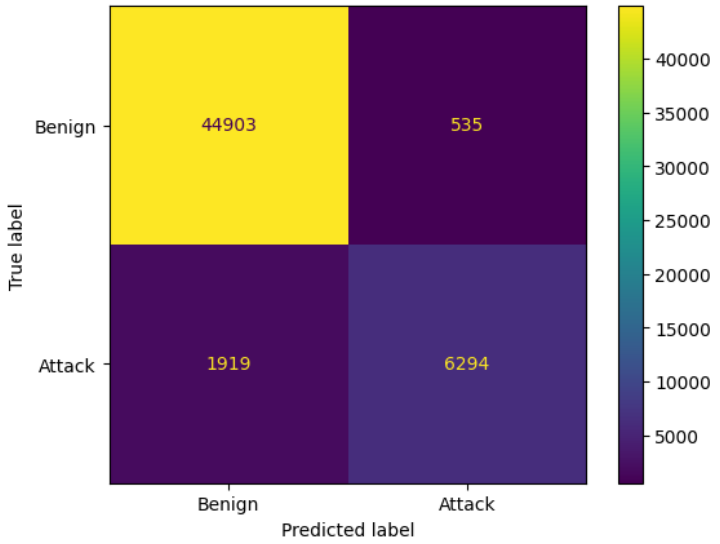


Рисунок 5.7 – Матриця помилок Double DQN на тренувальному наборі даних (рисунок виконано самостійно)

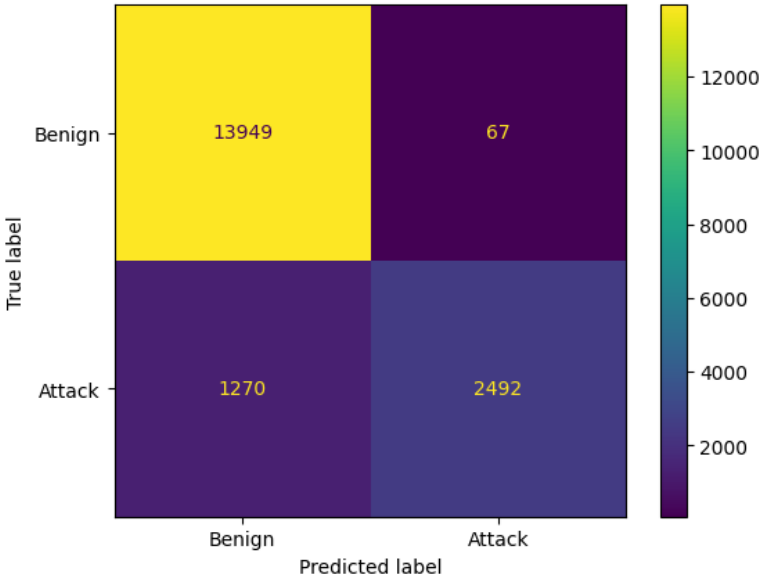


Рисунок 5.8 – Матриця помилок Double DQN на тестовому наборі даних (рисунок виконано самостійно)

Результати експерименту показали, що модель працює на достатньо високому рівні. Достатньо рівні значення accuracy між тренувальним і тестовим наборами даних свідчить про великий рівень узагальнення моделі. Модель розпізнає дуже малу кількість нормального трафіку як зловмисний, що є великим плюсом. Також, модель працює з дуже великою швидкістю, з часом розпізнавання < 0.01 мс на один набір потоків. Основною проблемою на даний момент є достатньо низькі значення recall для класу Attack (0.7663, 0.6624), що свідчить про те, що модель пропускає достатньо велику кількість реального зловмисного трафіку, але великі значення precision (0.9217, 0.9738) свідчать про те, що ті послідовності, які модель відмічає як зловмисні в більшості є дійсно зловмисними. Можливим методом покращення моделі є зменшення рівню розпаду епсілон, що дасть можливість моделі більше досліджувати датасет, що може призвести до кращих результатів.

5.6 Аналіз Результатів

Проведемо детальний аналіз результатів експериментального дослідження, в якому використовувалися різні методи розпізнавання DDoS атак. Методи були порівняні за їх продуктивністю, точністю прогнозів та іншими ключовими показниками. Для наглядного порівняння були побудовані порівняльні графіки методів розпізнования (див. рис 5.9 – 5.12)

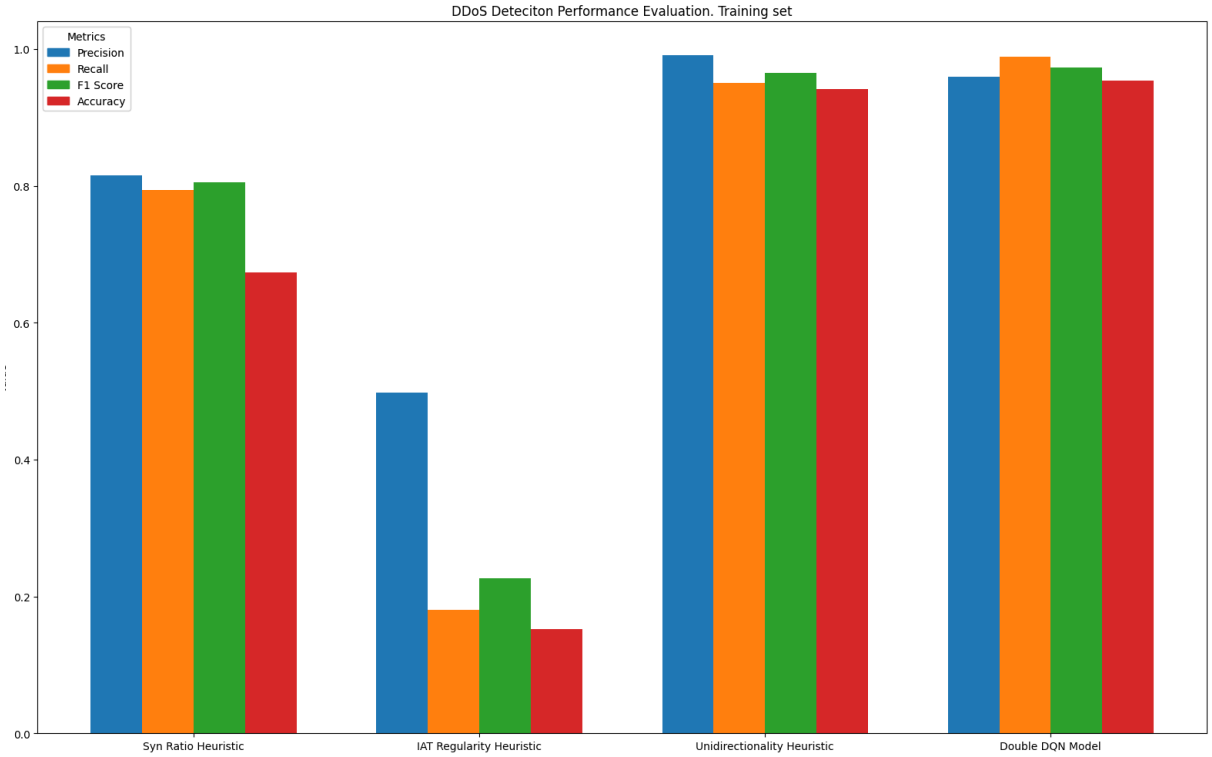


Рисунок 5.9 – Порівняння метрик на тренувальному наборі даних (рисунок виконано самостійно)

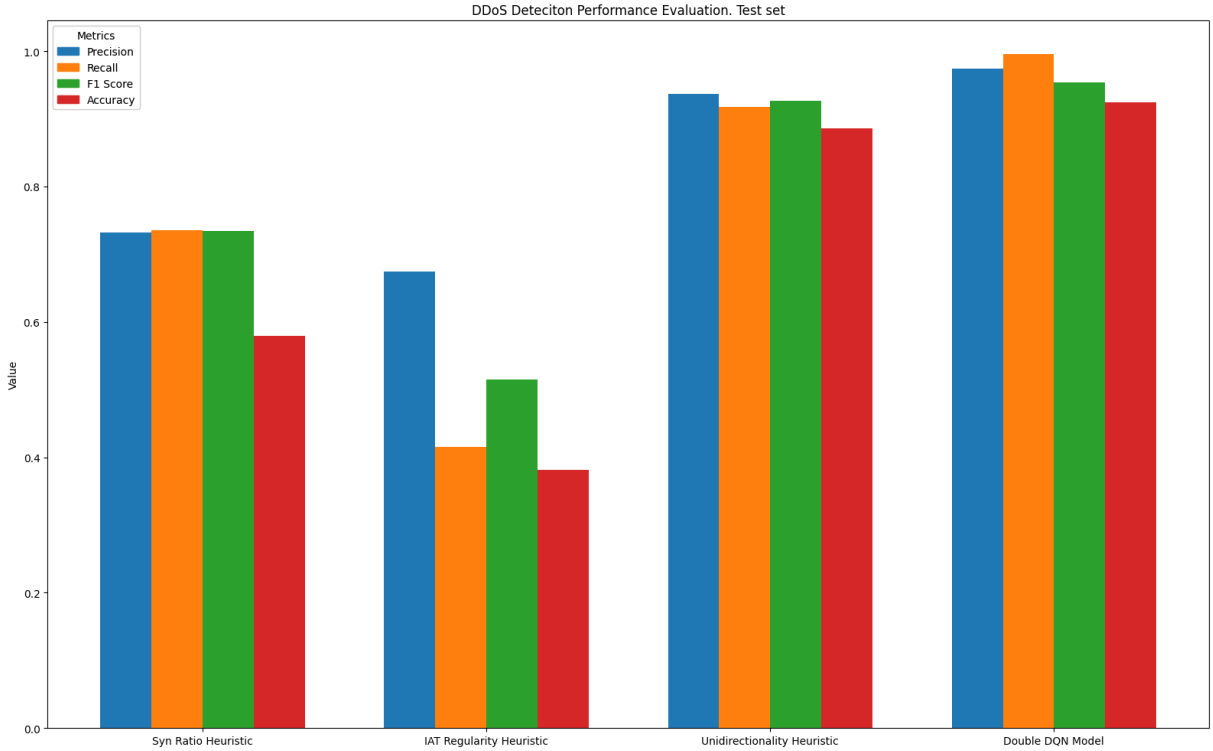


Рисунок 5.10 – Порівняння метрик на тестовому наборі даних(рисунок виконано самостійно)

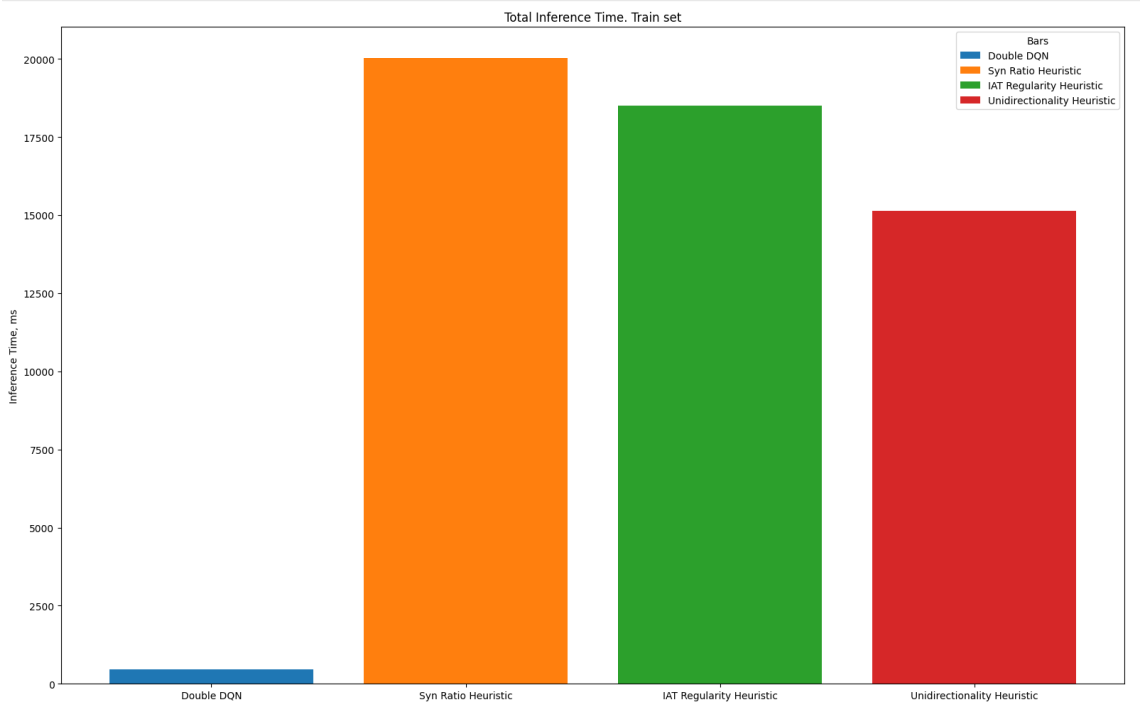


Рисунок 5.11 – Порівняння часу виконання експерименту на тренувальному наборі даних (рисунок виконано самостійно)

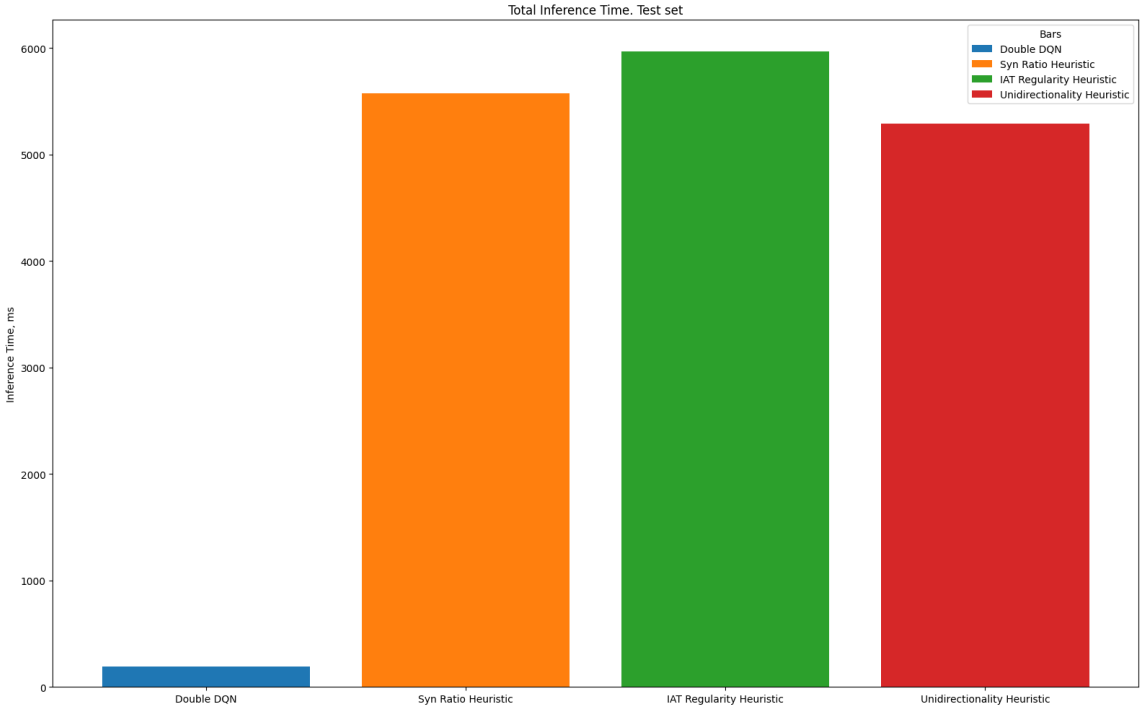


Рисунок 5.12 – Порівняння часу виконання експерименту на тестовому наборі даних (рисунок виконано самостійно)

Розглядаючи отримані результати, бачимо, що Double DQN демонструє найвищу ефективність серед порівнюваних методів. Крім того, він працює в понад 30 разів швидше, насамперед завдяки можливості обробляти мережеві потоки пакетами, а не поодинці. Завдяки цьому Double DQN здатен аналізувати близько 2 000 000 потоків на секунду.

Натомість більшість евристичних алгоритмів показують низькі значення precision та recall щодо виявлення шкідливого трафіку. Виняток складає метод Unidirectionality, який за точністю здатен конкурувати з Double DQN. Однак через відсутність можливості пакетної обробки даних цей підхід значно поступається за продуктивністю в масштабних системах, де критично важлива висока швидкодія.

ВИСНОВКИ

У межах виконання роботи було проведено комплексне дослідження методів виявлення DDoS-атак у зашифрованому трафіку з використанням алгоритмів глибинного навчання з підкріпленням. Робота охоплює теоретичні аспекти предметної галузі, аналіз сучасних підходів, архітектурних рішень і технічних засобів, а також практичну реалізацію прототипу системи з відповідними функціональними модулями.

На основі аналізу наукових джерел встановлено, що традиційні системи виявлення атак в умовах шифрування трафіку втрачають ефективність через недоступність вмісту пакетів. У зв’язку з цим найбільш перспективним є підхід, заснований на аналізі метаданих потоків у поєднанні з інтелектуальними методами прийняття рішень. Окрему увагу приділено reinforcement learning як методу, що дозволяє створити адаптивну систему, здатну самостійно навчатися в процесі експлуатації.

У ході роботи було розроблено архітектуру системи, яка включає модулі збору та обробки мережевого трафіку, формування ознак, реалізацію RL-агента (на основі алгоритму Double DQN), модуль прийняття рішень, базу даних для зберігання результатів, API-шлюз та інтерфейс користувача. Реалізований прототип дозволяє виявляти підозрілі потоки в режимі, наближеному до реального часу, з фіксацією усіх дій агента для подальшого аналізу та донавчання.

Отримані результати демонструють практичну придатність підходу reinforcement learning до задач виявлення DDoS-атак у зашифрованому трафіку та можуть бути основою для подальшого розвитку проєкту у напрямі повноцінної інтеграції з інфраструктурами безпеки, використання більш складних агентів або побудови мультиагентних систем у сфері кіберзахисту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kheddar H., Messai N., Himeur Y., Awad A. Deep transfer learning for intrusion detection in industrial control networks: A comprehensive review // Journal of Network and Computer Applications. – 2023. URL: https://doi.org/10.1016/j.jnca.2023.103760

2. Ferriyan H., Wibisono A., Muchtar A. HIKARI-2021 Dataset: Benchmark for Encrypted Traffic DDoS Detection // IEEE Access. – 2021. URL: https://doi.org/10.3390/app11177868

3. Vargas-Rosales C., Lopez-Ortiz G., Mendez-Hernandez H. SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning: // IEEE Access – 2021. URL: https://doi.org/10.1109/ACCESS.2021.3101650.

4. Yang J., Liang G., Wen G., Gao T. A deep-learning- and reinforcement-learning-based system for encrypted network malicious traffic detection // Information and communications – 2021. URL: https://doi.org/10.1049/ell2.12125

5. Hu J., Yang X., Hu J., Peng Y. A Q-learning algorithm for Markov decision processes with continuous state spaces // Systems & Control Letters – 2024. URL: https://doi.org/10.1016/j.sysconle.2024.105782

6. Yungaicela-Naula M., Vagras-Rosales C., Perez-Dias J., Carrera D. A flexible SDN-based framework for slow-rate DDoS attack mitigation by using deep reinforcement learning // Journal of Network and Computer Applications – 2022. URL: https://doi.org/10.1016/j.jnca.2022.103444

7. OpenAI Gym. Documentation [Електронний ресурс]. URL: https://gym.openai.com/

8. PyTorch Documentation [Електронний ресурс]. URL: https://pytorch.org/docs/

9. InfluxDB Documentation [Електронний ресурс]. URL: https://docs.influxdata.com/influxdb/v2/

10. О.Ф. Лановий, І.В. Кобзев, О.С. Удовенко «Система підрахунку трафіку мережі з використанням засобів об’єктно-орієнтованого програмування». Право і Безпека 7,№ 2 с. 213-217 2008. URL: http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/Pib_2008_7_2_46.pdf

11. Поліщук В. Розробка системи виявлення вторгнень на основі аналізу аномалій у мережевому трафіку з використанням машинного навчання // Матеріали XII науково-технічної конференції „Інформаційні моделі, системи та технології“. – 2024. – С. 76-76. [6]

12. Головенко Б. В. Розробка програмного забезпечення для виявлення DDOS атак : дис. – ТНТУ, 2022.

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

10. О.Ф. Лановий, І.В. Кобзев, О.С. Удовенко «Система підрахунку трафіку мережі з використанням засобів об’єктно-орієнтованого програмування». Право і Безпека 7,№ 2 с. 213-217 2008. URL: http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/Pib_2008_7_2_46.pdf