Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель та методи виявлення широкомасштабної атаки в середовищі IoT

(тема)

Виконав:

студент II курсу, групи СПм-22-6

Великодний І.А.
(прізвище, ініціали)

Спеціальність

123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: доц. Ляшенко О.С
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

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

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____ 20 ___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Великодному Ігорю Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Модель та методи виявлення широкомасштабної атаки в середовищі IoT

затверджена наказом по університету від " 01 " квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 черня 2024 р.

3. Вхідні дані до роботи Реалізувати систему виявлення вторгнень, яка сповіщатиме користувача про кібератаку.

4. Перелік питань, що потрібно опрацювати у роботі

1. Розглянути методи виявлення вторгнень;
2. Запропонувати власну модель системи;
3. Проведення експериментальних досліджень нейронних мереж;
4. Створення системи виявлення вторгнень;
5. Реалізація інтерфейсу командного рядка системи;
6. Висновки;

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Слайд-презентація – 12 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Розглянути методи виявлення вторгнень	02.04.24-08.04.24	
2	Запропонувати власну модель системи	09.04.24-16.04.24	
3	Проведення експериментальних досліджень	17.04.24-22.04.24	
	нейронних мереж		
4	Створення системи виявлення вторгнень	23.04.24-05.05.24	
5	Реалізація інтерфейсу командного рядка	06.05.24-23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-30.05.24	
7	Подання кваліфікаційної роботи керівникові та	31.05.24-04.06.24	
	її попередній захист		
8	Подання кваліфікаційної роботи на рецензу-	05.06.24-12.06.24	
	вання		

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Ляшенко О.С

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 65 с., 17 рис., 2 дод., 14 джерел.

IDS, IOT, СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕНЬ, НЕЙРОННА МЕРЕЖА, МАШИННЕ НАВЧАННЯ.

Метою даної роботи є запропонування системи виявлення вторгнень в режимі реального часу, яка буде навчена на наборі з великим обсягом даних, за допомогою нейронної мережі з використанням ансамблевого методу машинного навчання

У ході виконання кваліфікаційної роботи було проведено аналіз існуючих концепцій систем та методів виявлення вторгнень в інфраструктурі IoT. Розроблено власну систему виявлення вторгень. Крім того було створено програмне забезпечення у вигляді CLI інтерфейсу.

ABSTRACT

Master’s thesis: 65 pages, 17 figures, 2 appendix, 14 sources.

IDS, IOT, INTRUSION DETECTION SYSTEM, NEURAL NETWORK, MACHINE LEARNING.

The purpose of this work is to propose a real-time intrusion detection system, which will be trained on a set with a large amount of data, using a neural network using an ensemble method of machine learning

In the course of the qualification work, an analysis of existing concepts of systems and methods for detecting intrusions in the IoT infrastructure was carried out. A proprietary intrusion detection system has been developed. In addition, software was created in the form of a CLI interface.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

IDS – система виявлення вторгнень (англ., Intrusion Detection System)

IoT – Інтернет речей (англ., Internet of things)

RF – модель ансамблевого методу машинного навчання (англ., Random Forest)

CLI – інтерфейс командного рядка (англ., Command-line Interface)

ВСТУП

Інтернет речей – концепція мережі, яка об’єднує фізичні пристрої з вбудованими датчиками, а також програмним забезпеченням, що забезпечує ефективну та спрощену взаємодію між фізичним світом і комп’ютерними системами, за допомогою, найчастіше, стандартних протоколів зв’язку. Протягом останніх років він стрімко зростав та продовжує зростати у різних галузях. Пристрої IoT функціонують у сферах освіти, охорони здоров’я, сільському господарстві, транспортних системах та промисловості. Кількість підключених пристроїв по всьому світу стрімко росте. Системи включають в себе масу датчиків, які дозволяють збирати дані в реальному часі. Отримані дані, це свого роду фундамент для створення інтелектуальних алгоритмів прийняття рішень. Ростуча кількість пристроїв, ціна і важливість інформації збільшує ризик кіберзагроз і викраденню інформації в корисних цілях. Виходячи з цього, розробка інтелектуальних методів та систем виявлення вторгнень для пристроїв IoT стає необхідною для їх ефективного захисту. Тема безпеки інформаційного середовища стає дедалі актуальною і кібербезпека набуває життєвої важливості, з огляду на те що IoT є драйвером промислової революції та системою для збору живих даних [1]. Таким чином, система виявлення вторгнень є необхідною для виявлення і захисту мережі та пов’язаних систем від поточних і майбутніх кібератак.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд поточного стану IoT технологій та їх застосування

Як ми вже визначили, Інтернет речей став однією з найважливіших технологій сучасності, що впливає на всі сфери життя: від побутових пристроїв до промислових систем. Далі розберемось більш детально з архітектурою та визначемось з ключовими термінами:

Мережа зв’язку (Communication Network) – інфраструктурна мережа, що з’єднує пристрої та додатки.

Річ (Thing) – предмет фізичного світу (фізичні речі) або інформаційного світу (віртуальні речі), який може бути ідентифікований та інтегрований в мережі зв’язку.

Пристрої (Devices) – відіграють ключову роль у Інтернеті речей. Це елементи обладнання, які володіють обов’язковими можливостями зв’язку та додатковими можливостями вимірювання, спрацьовування, а також введення, зберігання і обробки даних. Ці пристрої включають в себе сенсори, актуатори, камери, побутову техніку та інші компоненти.

Сенсори є основними пристроями для збору даних в IoT. Вони перетворюють фізичні явища у цифрові сигнали, які в свою чергу можуть мути передані і оброблені. Наведемо приклади сенсорів, які використовуються для різних завдань:

- температурні сенсори, вимірюють температуру навколишнього середовищі або конкретних об’єктів. Їх застосовують в системах контролю клімату, холодильнику або в інших виробничих процесах;

- вологісні сенсори, вимірюють рівень вологості у повітрі або в матеріалах. Їх застосовують у сільському господарстві, системах кондиціонування повітря, зберігання продуктів, тощо;

- сенсори тиску, вимірюють тиск газів або рідин. Їх застосовують в промисловому обладнанні, автомобільних системах, системах водопостачання, тощо;

- сенсори світла, вимірюють інтенсивність освітлення. Їх застосовують в системах автоматичного освітлення, моніторингу енергоефективності, системах безпеки;

- рухові сенсори, виявляють рух у заданій області. Їх застосовують в системах безпеки, автоматичного освітлення, спортивних трекерах, тощо;

- сенсори якості повітря, вимірюють концентрацію забруднюючих речовин у повітрі. Застосовують у моніторингу навколишнього середовища, системах вентиляції, розумних будинках.

Актуатори (Actuators) – це пристрої, які здійснюють фізичні дії у відповідь на команди з мережі. Вони перетворюють цифрові електричні сигнали у мехачнічні рухи або інші фізичні дії. Наведемо приклади актуаторів:

- моторні актуатори, прилади, які перетворюють електричний сигнал у обертальний або поступальний рух. Їх застосовують в робототехніці, автоматизованих виробничих лініях, системах кондиціонування повітря тощо;

- електромагнітні клапани, керують потоком рідин або газів у системах трубопроводів. Застосовують у системах зрошення, водопостачання та інших промислових процесах;

- п’єзоелектричні актуатори, які використовують п’єзоелектричний ефект для створення механічних рухів. Застосовують у прецизійних інструментах, медичних пристроях, акустичних системах тощо;

- термоелектричні актуатори, використовують електричний струм для створення температурних змін. Застосовують у термостатах та системах контролю температури.

Камери (Cameras) – використовують для збору зображень або відео, а також для візуального моніторингу. Наведемо приклади:

- відеокамери, записують відео у режимі реального часу. Застосовуються в системах безпеки, моніторингу транспорту та навколишнього середовища, медичних діагностичних системах тощо;

- фотокамери, збирають статичні зображення та застосовуються у моніторингу навколишнього середовища, контролювання виробничих процесів, автоматизації систем інспекції тощо;

- тепловізійні камери, вимірюють і візуалізують температурні поля. Основні області застосування: безпека, медичні обстеження, промислова діагностика.

Наступним ключовим компонентом в архітектурі Інтернету речейвідокремимо гейтвей, або шлюз. Елемент що забезпечує зв’язок між кінцевими пристроями і хмарними сервісами, об’єднуючи різноманітність цих пристроїв в єдину систему. Гейтвеї такоє виконують функції попередньої обробки даних та забезпечення безпеки.

Гейтвеї можуть підтримувати різні протоколи зв’язку, такі як Wi-Fi, Zigbee, LoRa, Bluetooth, NB-IoT, LTE, Ethernet тощо. За допомогою цього можливо підключати різні типі кінцевих пристроїв до єдиної мережі. Гейтвеї перетворюють дані з одного протоколу в інший забезпечуючи безперервний потік даних між пристроями і хмарними сервісами. Вони можуть фільтрувати дані зменшуючи обсяг переданих до хмари даних. Завдяки цьому зменшується навантаження на мережу та зберігаються обчислювані ресурси. Використовуючи вбудовані обчислювальні можливості, гейтвеї можуть виконувати аналіз даних на місці, приймати рішення у реальному часі та виявляти аномалії. Вони можуть зберігати дані у випадку перебоїв зв’язку з хмарними сервісами, забезпечуючи цілісність даних після відновлення зв’язку.

Гейтвеї забезпечують шифрування даних між кінцевими пристроями та хмарою, захищаючи інформацію від несанкціонованого доступу, здійснюють аутентифікацію пристроїв і користувачів, контролюючи доступ до мережі та даних. Вони можуть виявляти загрози та аномалії у трафіку, здійнюючи моніторинг безпеки у реальному часі.

Гейтвеї поділяються за типами:

- периферійні, які знаходяться ближче до кінцевих пристроїв та виконують більшу частину попередньої обробки даних на місці, зменшуючи затримку і підвищуючи ефективність. Застосовуються у промислових системах, смарт-будинках та транспортних системах;

- глобальні, які розміщені у хмарі або великих дата-центрах, оброблюють дані з великох кількості пристроїв і здійснюють складний аналіз та зберігання. Застосовуються у централізованому управлінні великими IoT інфраструктурами.

Хмарні платформи – є центральним елементом архітектури Інтернету речей.

Забезпечують потужні обчислювальні ресурси і великий обсяг сховища для зберігання та аналізу даних, які були зібрані з кінцевих пристроїв. Вони дозволяють здійснювати складний аналіз даних, управляти IoT пристроями на відстані та забезпечувати безпеку. До основних компонентів хмарних платформ відносять:

- сховища даних (Data Storage), які в свою чергу діляться на реляційні бази даних (SQL), які використовуються для зберігання структурованих даних, що організовані у таблиці; нереляційні бази даних (NoSQL), які використовуються для зберігання неструктурованих або напівструктурованих даних, що можуть бути більш гнучкими для масштабування та хмарні сховища, розподілені системи зберігання даних, такі як Amazon S3, Google Cloud Storage, Azure Blob Storage, що забезпечують надійність та доступність даних;

- обчислювальні ресурси (Compute Resources), серед яких: віртуальні машини (Virtual Machines), емуляції фізичних машин, які дозволяють запускати різні операційні системи і додатки; контейнери, легкі віртуалізації, що ізолюють додатки в окремі середовища, забезпечуючи їх швидке розгортання і масштабування; безсерверні обчислення (Serverless Computing),

платформи, що виконують код у відповідь на події, без необхідності управління серверами (наприклад, AWS Lambda, Google Cloud Functions);

- аналітичні сервіси (Analytics Services), які займаються обробкою потокових даних (stream processing), аналізом великих даних (big data analytics) та сервісами з машинним навчанням;

- управління пристроями (Device Management), який включає в себе реєстрацію і моніторинг, інструменти для реєстрації нових пристроїв, моніторингу їх стану і управління ними; оновлення прошивки (firmware updates), можливість дистанційного оновлення програмного забезпечення на пристроях; управління конфігураціями, інтсрументи для налаштування і зміни параметрів пристроїв;

- безпека, яка має аутентифікацію і авторизацію, шифрування даних та виявлення загроз і аномалій.

1.2 Системи виявлення вторгнень

Визначимо концепцію IDS (Intrusion Detection System, або система виявлення вторгнень). Це програмний або апаратний засіб, який виявляє або запобігає несанаційному доступу до комп’ютерної мережі чи системи. Головна мета IDS полягає в реагуванні на небезпечні події, потенційно небезпечні, або аномалії, що можуть вказувати на вторгнення чи інші безпекові порушення. З основних завдань системи виявлення вторгнень можна виділити: виявлення аномалій - являє собою функцію моніторингу системи чи мережі для виявлення незвичайних патернів, подій або некоректних дій, які можуть бути ознакою вторгнення чи іншої загрози безпеці; виявлення вторгнень – розпізнавання несанкційного доступу, спроб атак на інформаційні системи, вірусів, троянських програм та іншого шкідливого коду; відслідковування і реагування – забезпечення можливості вжиття заходів до виявлених загроз, включаючи блокування доступу, відключення систем, які на думку є найбільш вразливі або відправлення сповіщень адміністраторам, котрі

відповідають за безпеку. Хоч існує декілька типів IDS, які за розміром варіюються від окремих комп’ютерів до великих мереж, найпоширенішими класифікаціями є ситеми виявлення вторгнень у мережу (NIDS) та ситеми виявлення вторгнень на аналізі хостів (HIDS)

1.2.1 Класифікація систем виявлення вторгнень

Мережеві IDS (Network-based IDS, NIDS) розташовуються в стратегічному місці або у таких місцях мережі, де можливий котроль трафіку всіх пристроїв у мережі. Вони здійнюють контроль усього трафіку даних всієї підмережі та порівнюють трафік, який передається у підмережі з бібліотекою відомих атак. Як тільки атака розпізнана або визначено відхилення у поведінці, відразу відсилається попередження адміністратору. Наприклад NIDS встановлюють у підмережі, де розташовані мережеві екрани, щоб побачити, чи намагається хось втрутитися в систему. NIDS-система може котролювати велике число TCP-запитів на з’єднання (SYN) з багатьма портами на обраному комп’ютері, виявляючи, таким чином, що хтось намагається здійнисти сканування TCP-портів. Мережева IDS може запускатися або на окремому комп’ютері, який контролює свій власний трафік, або на виділеному комп’ютері, переглядати весь трафік у мережі (маршрутизатор). Мережеві IDS контролюють багато комп’ютерів тоді як інші IDS контролюють тільки один.

Хостові IDS розташовуються і встановлюються на хості і виявляють зловмисні дії на ньому. Прикладом хостових IDS можуть бути системи контролю цілісності файлів, які перевіряють системні файлі з метою визначення, коли в них були внесені зміни. Монітори реєстраційних файлів (Log-file monitors, LFM), контролюють реєстраційні файли, створювані мережевими сервісами і службами. Обмані системи, що працюють з псевдосервісами, мета яких полягає у відтворенні добре відомих вразливостей для обману зловмисників.

Статичні і динамічні IDS:

- статичні засоби роблять “знімки”(snapshot) середовища та здійнюють їх аналіз, розшукуючи вразливе ПО, помилки в конфігураціях і т. д. Статичні IDSперевіряють версії прикладних програм на наявність відомих вразливостей і слабких паролів, перевіряють вміст спеціальних файлів в директоріях користувачів або перевіряють конфігурацію відкритих мережевих сервісів. Статичні IDS виявляють сліди вторгнення;
- динамічні IDS здійнюють моніторинг у реальному часі всіх дій, що відбуваються в системі, переглядаючи файли аудиту або мережні пакети, що передаються за певний проміжок часу. Динамічні IDS реалізують аналіз в реальному часі і дозволяють постійно стежити за безпекою системи.

Деякі системи виявлення вторгнень можуть виявити початок атаки на мережу, причому деякі з них здатні виявляти раніше невідомі атаки. Такі ситеми називають системами запобігання вторгнень (Intrusion Prevention System, IPS). Системи IPS можна розглядати як розширення систем виявлення вторгнень, так як завдання відстеження атак залишається однаковою. Однак, вони відрізняються в тому, що IPS повинна відслідковувати активність в реальному часі і швидко реалізовувати дії щодо запобігання атак.[10]

1.2.2 Класифікація систем запобігання вторгнень

Мережева система запобігання вторгненням (NIPS) – це система, яка використовується для моніторингу мережі, а також для захисту конфіденційності, цілісності та доступності мережі. Її основні функції включають захист мережі від загроз, таких як відмова в обслуговуванні (DoS) і несанкціоноване використання.

NIPS відстежує мережу на наявність шкідливих дій або підозрілого трафіку, аналізуючи активність протоколу. Після встановлення NIPS у мережі він використовується для створення фізичних зон безпеки. Це, у свою чергу, робить мережу інтелектуальною та швидко розрізняє хороший трафік від

поганого. Іншими словами, NIPS стає схожою на в’язницю для ворожого трафіку, такого як трояни, хробаки, віруси та поліморфні загрози. Система запобігання вторгненням працює в мережі та контролює трафік. Коли відбувається підозріла подія він вживає заходів на основі певних встановлених правил. IPS – це активний пристрій, що працює в режимі реального часу, на відміну від системи виявлення вторгнень, яка не вбудована і є пасивним пристроєм. IPS вважаються еволюцією системи виявлення вторгнень.[11]

Система запобігання бездротовому вторгненню (Wireless Intrusion Prevention System, WIPS) – це рішення безпеки, призначене для моніторингу, захисту та запобігання зловмисним атакам і загрозам для бездротових мереж. WIPS зосереджується на моніторингу та реагуванні на аномальну діяльність у бездротових мережах, таким чином захищаючи мережу від несанкційного доступу, зловмисних атак та інших загроз безпеці.

WIPS забезпечує безпеку бездротових мереж за допомогою низки передових технологій:

- моніторинг мережі: WIPS постійно відстежує радіочастоти (наприклад 2.4 ГГц і 5 ГГц), виявляючи присутність і діяльність бездротових точок доступу (AP) і клієнтських пристроїв;

- класифікація бездротових пристроїв: WIPS перевіряє всі виявлені пристрої в мережі, підтверджуючи, чи є вони авторизованими. Це стосується точок доступу, клієнтів та інших бездротових пристроїв;

- виявлення загроз: WIPS аналізує шаблони бездротового трафіку та поведінку пристроїв, використовуючи попередньо визначені політики безпеки та бібліотеку сигнатур для виявлення потенційних загроз, таких як неавторизоване підключення пристроїв до мережі, несанкціоновані точки доступу, DoS-атаки та спроби проникнення в мережу;

- захист і реагування: після виявлення загроз або порушень WIPS може вжити захисних заходів, включаючи відключення потенційно зловмисних пристроїв, ізоляцію атак або надсилання сповіщень мережевим адміністраторам;

- аналіз даних і звітність: WIPS надає докладні журнали подій безпеки та продуктивності, аналізуючи дані для виявлення моделей атак і вразливостей у безпеці мережі. Він також може створювати звіти, щоб допомогти у дотриманні правил і політики безпеки;

- автоматизація та інтеграція. Сучасні системи WIPS зазвичай можуть інтегруватися з іншими системами безпеки, такими як системи контролю і доступу до мережі (NAC) і системи керування інформацією та подіями безпеки (SIEM), щоб автоматично реагувати та обробляти події безпеки;

- керування політикою та оновлення: WIPS дозволяє визначити та підтримувати політики безпеки бездротового зв'язку, які регулярно оновлюються на основі постійно змінюваного середовища загроз безпеці.[12]

Аналіз поведінки мережі (Network Behavior Analysis, NBA) – це техніка для підвищення безпеки приватної мережі шляхом моніторингу трафіку та запису неочікуваної поведінки або відхилень від типового функціонування. У той час як традиційні системи запобігання вторгненням використовують перевірку пакетів, виявлення сигнатур і блокування в реальному часі для захисту периметра мережі, системи NBA стежать за тим, що відбувається всередині мережі, збираючи дані з багатьох джерел, щоб забезпечити аналіз в автономному режимі. Після встановлення базової лінії для регулярного трафіку програмне забезпечення NBA спостерігає за мережевою аткивністю у фоновому режимі та позначає будь-які невідомі, неочікувані або незвичайні моделі, які можуть сигналізувати про наявність загрози. Системи NBA можуть виявляти підозрілу активність у корпоративних мережах за допомогою розширеної аналітики, машинного навчання та методів на основі правил. Вони також можуть відстежувати та записувати шаблони використання пропускної здатності та протоколу. Аналіз поведінки мережі особливо корисний для виявлення вразливостей нульового дня та нових шкідливих програм.

Система аналізу поведінки мережі надає адміністраторам мережі та безпеки такі можливості:

- видимість мережі: допомагає адміністраторам мережі мати повне розуміння того, що відбувається в мережі. NBA надає командам NetOps суттєве бачення невідомих і не підозрюваних ризиків на основі ненормальної мережевої активності, дозволяючи їм визначити пріоритетність розслідування і заходів реагування на основі серйозності загрози;

- виявлення мережевої поведінки: виявляє зловмисну поведінку за допомогою статистики мережевого трафіку, експортованої маршрутизаторами / комутаторами або мережевими зондами (NetFlow, jFlow, IPFIX, NetStream та інші стандарти потокових даних). Коли поведінка організації вдхиляється від базової лінії, оцінюється ризик і якщо ризик виглядає високим, NBA може надіслати сигнал тривоги на інформаційну панель, яку спостерігають оператори мережі;

- ідентифікація та пом’якшення загроз: це додаткове рішення для виявлення розширених загроз, які не виявляються типовими рішеннями, таких як ботнети, невідоме зловмисне програмне забезпечення, інсайдерські загрози, витік даних і DDoS-атаки, у колі інформаційної безпеки. NBA особливо цінний для виявлення нових, невідомих зловмисних програм, експлойтів нульового дня та атак, що повільно розвиваються, а також шахрайської поведінки інсайдерів мережі. Ця стратегія також корисна, коли трафік загрози зашифровано, наприклад канал командування та керування (C&C);

- усунення несправностей мережі: це оптимізує операції з усунення несправностей мережі шляхом автоматичного виявлення аномалій і операційних проблем.[13]

IPS для окремих комп’ютерів (Host-based Intrusion Prevention System, HIPS) – продукти цього класу управляють правами застосунків на виконання тих чи інших дій, подібно до того, як брандмауер керує мережевим доступом, схожість принципів роботи обумовлює той факт, що нерідко HIPS об’єднуються з брандмауером в складі одного захисного продукту.

HIPS є засобом проактивного захисту, тобто не містить бази даних сигнатур вірусів і не здійснює їх детектування, таким чином, HIPS оперує не

поняттями “легітимний файл – шкідливий файл”, а поняттями “дозволена дія – заборонена дія”. Ефективність HIPS може добігати до 100% запобігання пошкодження або інфікування системи, однак більшість програм цього класу вимагає від користувача певних знань для управління ними. Відповідно до принципу організації захисту HIPS можуть бути поділені на три істотні групи:

- класичні HIPS. Системи оснащені відкритою таблицею правил. На підставі цієї таблиці драйвери HIPS дозволяють або забороняють певні дії з боку застосунків або запитують користувача про те, що необхідно зробити по відношенню до даної дії. Такий пристрій системи орієнтований на ручне керування дозволами та активну взаємодію з користувачем, що пред’являє високі вимоги до компетентності останнього;

- експертні HIPS, або поведінкові евристики. Здійснюють аналіз активності працюючого застосунку. Якщо сукупність дій, що виконуються набуває підозрілий або небезпечний характер, продукт даного типу повідомляє про ймовірну присутність шкідливої програми;

- HIPS типу Sandbox(пісочниця). Реалізують принцип мінімальної взаємодії з користувачем. В їхній основі лежить поділ застосунків на довірені і недовірені. На роботу довірених застосунків HIPS не надає ніякого впливу, в той час як недовірені запускаються в спеціальному просторі, відмежовані від системи. Це дозволяє працювати з підозрілими застосунками без ризику інфікування або пошкодження систем і вивчати звіти про їхню активність.[14]

1.3 Методи систем виявлення вторгнень

Системи виявлення вторгнень можуть використовувати різні методи, такі як сигнатурний аналіз, виявлення аномалій, використання інтелектуальних технологій, включаючи машинне навчання та евристичний аналіз. Ефективний захист включає в себе інтеграцію системи виявлення вторгнень з іншими методами безпеки, та поєднання цих методів, для створення комплексного захисту інформаційного стеку. Зазвичай системи

виявлення вторгнень використовують два основні підходи для виявлення потенційних загроз: сигнатурний аналіз та виявлення аномалій. Розглянемо ці методи більш детально.

Сигнатурний аналіз – метод який ґрунтується на використанні визначених сигнатур або патернів для ідентифікації або розпізнавання конкретних відомих загроз. Сигнатури можуть представляти з себе конкретні приклади або вирази в шкідливому програмному коді, унікальні характеристики того чи іншого вірусу чи способу вторгнення, які раніше вже були визначені або вивчені [2]. Спеціалісти з безпеки аналізують атаки і розробляють сигнатури для кожного виду. Зазвичай це може бути характеристика конкретних строк коду, значень в певних полях або якийсь інший ідентифікатор, який буде унікальним для деяких типів атак. Система виявлення вторгнень застосовує ці сигнатури для пошуку вхідних даних чи активності в мережі, які відповідають зазначеним сигнатурам. Якщо є збіг, система дає сповіщення про потенційне вторгнення. Сигнатурний аналіз ефективний і має високу точність проти відомих векторів атак та відомих загроз, але не ефективний проти нових та невідомих загроз. Він потребує постійного оновлення бази сигнатур для визначення нових загроз, більш того, зловмисники можуть уникати виявлення, шляхом зміни або шифрування свого коду. В сучасному середовищі сигнатурний аналіз залишається надійним засобом для виявлення вторгнень, але йому важко справлятися з векторами атак, що постійно змінюються, які все частіше використовують нові техніки та методи, тому в сучасних IDS його часто доповнюють інші методи, тобто використовується комбінація різних методів для комплексного захисту, такі як виявлення аномалій, для більшої ефективності виявлення нових атак.

Виявлення аномалій – метод який базується на аналізі звичайної поведінки мережі, системи, користувачів чи інших об’єктів. Система методу будує модель так званої “норми” на основі історичних даних, фокусується на виявленні незвичайностей, відхиленню від цієї норми, яке може бути ознакою нових загроз або підозр [3]. До підходів виявлення можемо віднести:

статистичні методи, які використовуються для аналізу величин, таких як середнє значення, середнє відхилення, тощо. Відхилення від норми цих величин може вказувати на присутність аномалії; методи машиного навчання – створення моделей за допомогою алгоритмів машинного навчання, які можуть визначати незвичайні патерни в даних та вказувати на аномалію, наприклад, за допомогою алгоритмів кластеризації або нейронної мережі; методи порівняння зразків – ґрунтуються на порівнянні поточної поведінки з історичними даними, якщо виявляється відхилення від звичайної моделі, це може бути зафіксовано як аномалія.

Виявлення аномалій може проводитися на основі патернів мережевого трафіку, неправильних адрес або портів, незвичних об’ємів даних, аналіз лог файлів, що містять інформацію про дію та поведінку системи чи користувачів, надто часті або великі запити, невластиві часові рамки, тощо. Застосування методу виявлення аномалій допомагає виявляти атаки, які можуть бути невідомими(нуль-день) та непередбаченими, що робить його ефективним і корисним для захисту від нових атак та загроз [3].

Сигнатурний аналіз і виявлення аномалій часто використовують в комплексі, як частина більших систем виявлення вторгнень. Комбінація цих методів дозволяє створити більш ефективну систему виявлення вторгнень, здатну протидіяти різноманітним загрозам безпеки.

1.4 Машинне навчання в системах виявлення вторгнень

Машинне навчання відіграє важливу роль у покращені ефективності та адаптивності в системах виявлення вторгнень. Воно дозволяє системам аналізувати дані, навчатися на їх основі, та виявляти нові невідомі загрози, класифікувати події як безпечні чи підозрілі. Навчання моделі на основі історичних даних та поведінки допомогає автоматично розпізнати нові атаки чи загрози. Щоб адаптуватися до змін у поведінці системи чи користувачів системи, виявлення вторгнень можуть використовувати онлайн навчання. Це

дозволить системі навчатися в реальному часі, а також підтримувати актуальність моделей. Машинне навчання ефективно працює з великими обсягами даних, що дозволяє виявляти складні патерни та взаємодії, які може бути важко виявити за допомогою традиційних методів [4]. Застосування машинного навчання дозволяє створювати інтелектуальні IDS, які можуть взаємодіяти та розпізнавати атаки на високому рівні. Використання машинного навчання в IDS є ключовим елементом для підвищення рівня захисту від сучасних загроз та забезпечення реактивності на нові типи атак. Розглянемо дві основні парадигми, які використовуються для розв'язання різних задач в машинному навчанні:

Supervised Learning (Навчання з вчителем) — спрямоване на розуміння зв'язку між вхідними та вихідними даними. Алгоритм, після встановлення цього зв'язку, може передбачити вихід для нових вхідних даних на основі того, що він дізнавався і зосереджується на методах класифікації та регресії. Групи класифікацій розбивають точки даних на різні класи. Цей підхід знаходить найкращій спосіб відокремити точки даних і призначити їх певним класам. Регресія відрізняється від класифікацій тим, що вона виводить число замість присвоєння точок даних класам. Класифікація фокусується на виведенні класу, тоді як регресія дає числовий вихід. Методи навчання з вчителем використовуються для виявлення відомих загроз і класифікації нових загроз за категоріями, як спам, фішинг та зловмисне програмне забезпечення [5].

Unsupervised learning (Навчання без вчителя) — набір даних містить лише вхідні дані та має справу з даними без, так званих, міток. Метою його є виявлення закономірностей або подібностей у наборі даних. Після отримання характеристик він групує дані на основі подібностей. Різниця від навчання з вчителем полягає в тому, що навчальний процес унікальний, оскільки алгоритм навчається на власному досвіді, а не на попередньо визначеному наборі вхідних даних із встановленим зв'язком. Методи навчання без вчителя використовуються для виявлення невідомих загроз і аномалій, які не належать до категорій відомих загроз [6,7].

Оскільки кількість і складність кіберзагроз зростає, ці типи машинного навчання особливо корисні для виявлення загроз, тому що вони можуть ідентифікувати аномалії та закономірності, які можуть бути виявленні не відразу.

1.5 Аналіз запропонованих IDS для IoT

На сьогоднішній день концепція IDS застосована до IoT не є чимось новим. Було розроблено і запропоновано багато рішень і систем, які використовують різні підходи та технології. Відзначимо деякі системи виявлення вторгнень для IoT:

- Cisco IoT Threat Defense: запропоноване рішення від Cisco, яке використовує аналіз трафіку, машинне навчання та інтелектуальні алгоритми для виявлення аномалій в мережі. Також вони акцентують на захисті від різноманітних атак, включаючи ті, в яких використовуються віруси та зловмисні програми;

- Darktrace Industrial: спеціалізується на застосуванні технологій штучного інтелекту для виявлення відхилень від звичайного патерну поведінки пристроїв. Їх система враховує контекст і адаптується до змін в мережі;

- Bastille Networks: спеціалузується на безпеці радіочастотного спектру для IoT пристроїв, таких як бездротові сенсори. Вони аналізують радіохвилі для виявлення аномалій та загроз;

- Check Point IoT Protect: пропонує рішення, яке включає виявлення вторгнень для IoT пристроїв. Вони використовують технології штучного інтелекту та аналіз трафіку;

- ARM mbed OS Sexurity: виходячи з назви, надає захист на рівні ОС для IoT пристроїв через свою платформу. Вони включають заходи безпеки, такі як аутентифікація та шифрування;

-

- Symantec IoT Security: надає рішення для захисту IoT пристроїв від різних загроз, включаючи вторгнення. Вони використовують аналіз поведінки та підписів для ідентифікації потенційно небезпечних дій.

Важливо відзначити, що ефективність кожної системи може залежати від конкретних задач та вимог використання. Для того щоб вибрати найкращу систему для конкретного випадку, потрібно ретельно ознайомитись з можливостями та різними рішеннями.

2 МОДЕЛЬ ТА ЇЇ РЕАЛІЗАЦІЯ

2.1 Вибір набіру даних та підготовка

Після детального розгляду методів систем виявлення вторгнень було вирішено обрати метод виявлення аномалій з використанням технології машинного навчання. В комплексі ця система буде більш адаптивною та здатною реагувати як на старі, так і на нові, раніше невідомі, загрози.

Для навчання моделі було обрано набір даних від Канадського інституту кібербезпеки – CIC(Canadian Institute of Cybersecurity) IoT 2023, який був зібраний у реальному часі для масштабних атак у середовищі IoT. Це достатньо новий і розширений набір даних про атаки в IoT для сприяння розробці додатків, аналітик і безпеки. В наборі відзначені дані з 33 атак, розділених на 7 класів.

DDoS	ACK Fragmentation
	UDP Flood
	SlowLoris
	ICMP Flood
	RSTFIN Flood
	PSHACK Flood
	HTTP Flood
	UDP Fragmentation
	ICMP Fragmentation
	TCP Flood
	SYN Flood
	SynonymousIP Flood
Brute Force	Dictionary Brute Force
Spoofing	Arp Spoofing
	DNS Spoofing

DoS	TCP Flood
	HTTP Flood
	SYN Flood
	UDP Flood
Recon	Ping Sweep
	OS Scan
	Vulnerability Scan
	Port Scan
	Host Discovery
Web-Based	Sql Injection
	Command Injection
	Backdoor Malware
	Uploading Attack
	XSS
	Browser Hijacking
Mirai	GREIP Flood
	Greeth Flood
	UDPPlain

Рисунок 2.1 – Класи атак

За допомогою нього буде навчена нейронна мережа, яка буде виконуватися в системі виявлення вторгнень, яка є метою цього документу, щоб класифікувати та виявляти мережевий трафік IoT, як зловмисний або безпечний.

Робота з підготовик даних, навчання та тестування буде проводитись у середовищі Jupyter Notebook на мові Python, версії 3.11.

Набір даних розділен на піднабори, тож напочатку роботи об’єднаємо їх в один великий, це зменшить продуктивність системи з точки зору пам’яті, але дасть нам мобільності при виконанні тих чи інших операцій в процесі навчання чи підготовки до навчання. Набір даних містить 46686579 записів, 46 ознак для навчання і клас-ознаку для класифікації.

Переходимо до фази підготовки даних, яка є найважливішою у процесі машинного навчання, бо якість та обсяг даних безпосередньо впливають на результати роботи моделі. Очищаємо дані, видаляємо відсутні значення, скидаємо індекс нашого фрейму даних і використовуємо замість нього стандартний, видаляємо рядки, які повторюються.

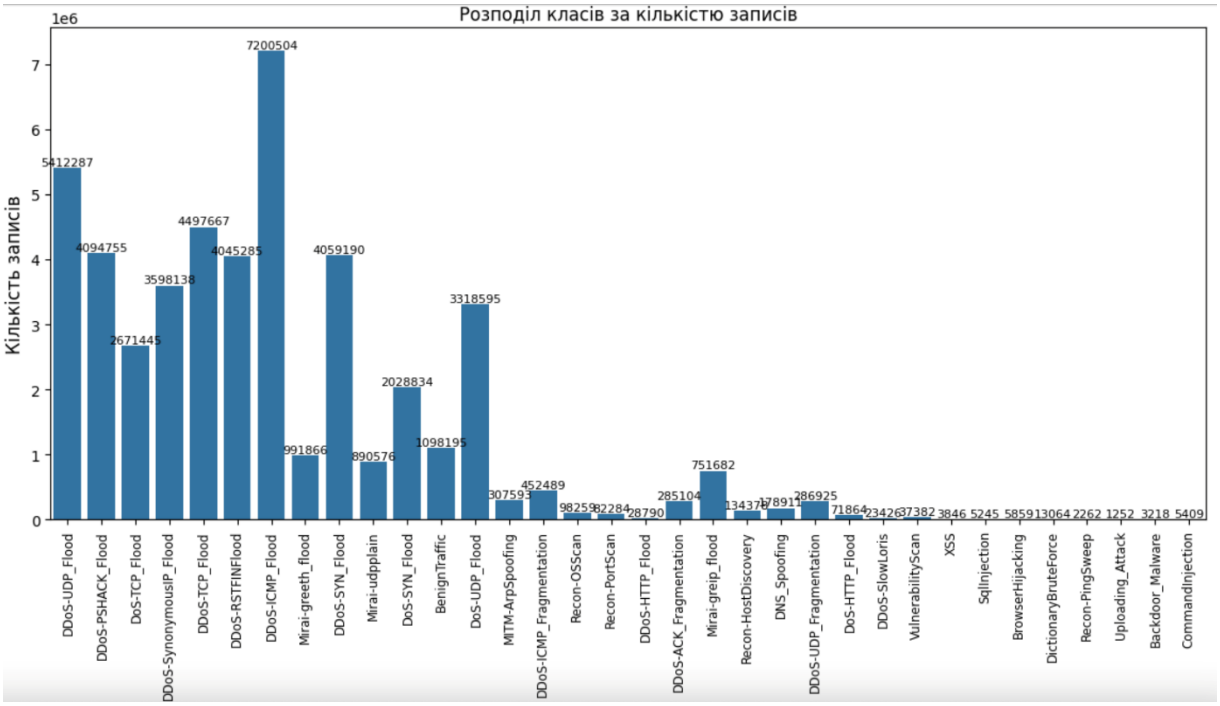


Рисунок 2.2 – Гістограма розподілу класів за кількістю записів

На гістограмі вище показано частоту кожного типу атаки в наборі даних. Як ми бачимо, деякі типи атак, такі як DDoS-ICMP_Flood, DDoS-UDP_Flood та інші в категорії DDoS, зустрічаються набагато частіше, ніж інші.

Об’єднаємо класи атак за типом:

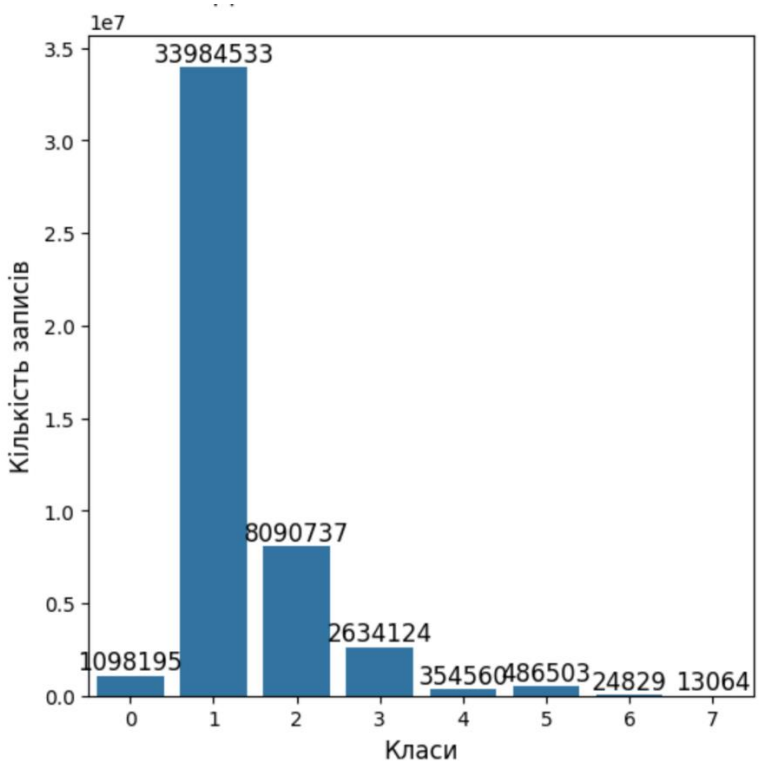


Рисунок 2.3 – Розподіл класів за кількістю записів: 0 – звичайний трафік; 1 – DDoS; 2 – DoS; 3 – Mirai; 4 – Recon; 5 – Spoofing; 6 – Web-Based, 7 – BruteForce.

Продемонструємо додаткові графіки поширення ознак тривалості потоку, темпу та типу протоколу. Згідно графіку більшість тривалостей потоку зосереджені на нижньому кінці, близько нуля, що свідчить про те, що багато потоків є дуже короткочасними, що характерно для деяких типів атак, які генерують багато швидкого трафіку. Ознака темпу також показує високу концентрацію близько нуля, що вказує на те, що багато потоків мають низькі швидкості, за кількома винятками, що досягають вищих значень.

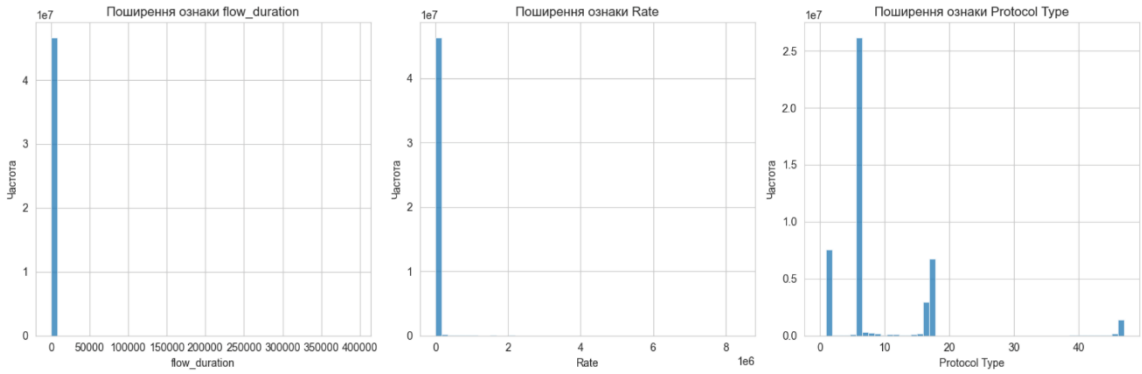


Рисунок. 2.4 – Поширення ознак тривалості потоку(flow_duration), темпу(rate) та типу протоколу(protocol type)

Тип протоколу є більш розповсюдженим, але є чітка концентрація навколо певних номерів протоколів, які, ймовірно, відповідають поширеним протоколам, таким як TCP і UDP.

Рисунок 2.5 – Розподіл типів атак за основними протоколами

На наступному графіку продемонструємо показник дисперсії, який використовується для визначення ступеня розподілу величини вимірюваних параметрів, у нашому випадку це довжина вхідних і вихідних пакетів у потоку.

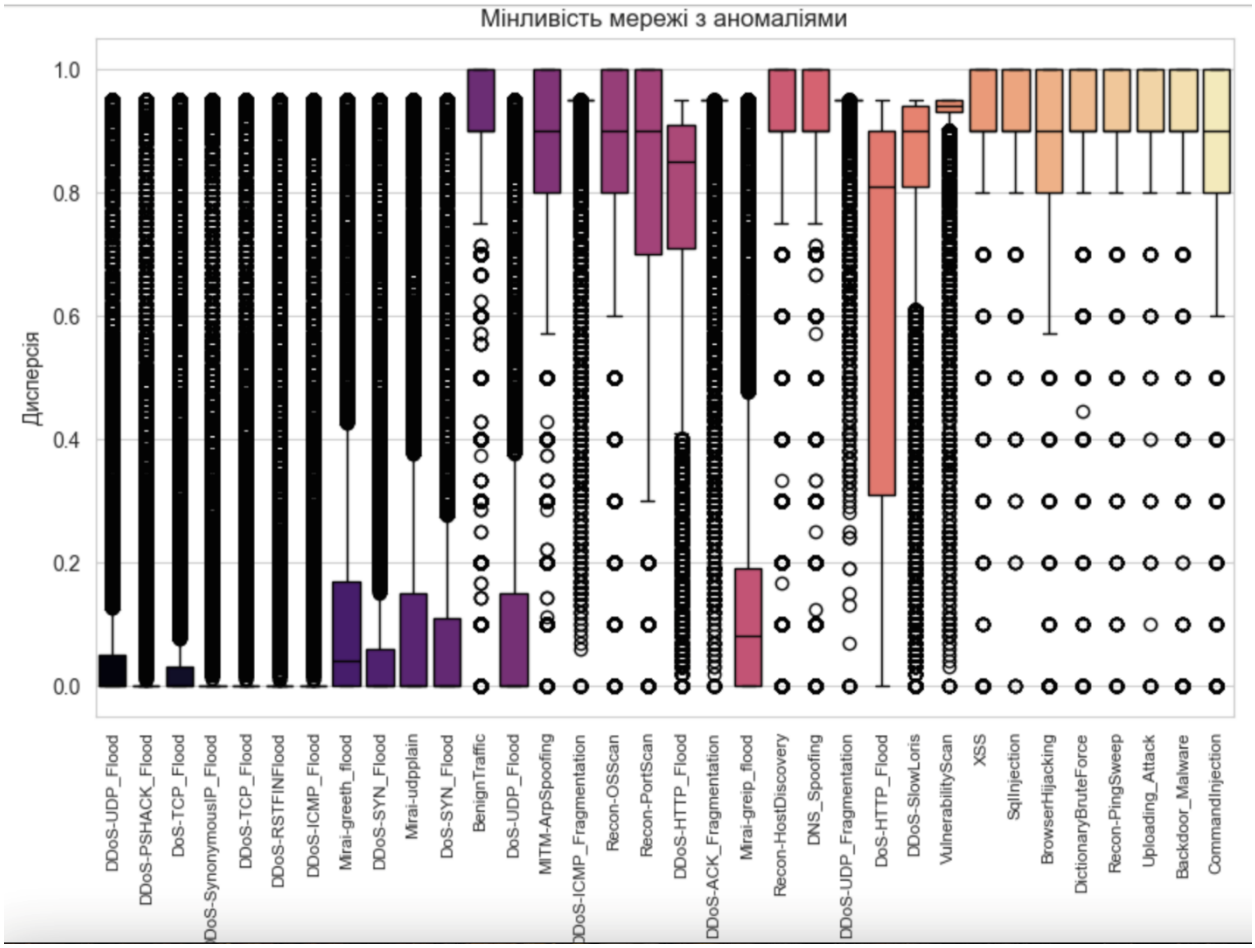


Рисунок 2.6 – Графік мінливості мережі з аномаліями. Дисперсія

Зробивши аналіз далі визначаємо важливі ознаки (певні характеристики з набіру даних), які далі будуть використовуватися для навчання моделі, нормалізуємо дані, для забезпечення стабільності та швидкості навчання, розділяємо на тренувальні та тестові набори. Правильна підготовка даних є критичним етапом, який може визначити невдачу чи успіх моделі в подальшому навчанні та роботою з реальними даними.

2.2 Аналіз архітектур і визначення найбільш підхожої

В ході виконання роботи і аналізу набору даних було вирішено скористатися ансамблевими методами навчання нейронної мережі. Вибір пав саме на ансамблеві з декількох причин:

- зменшення ризику перенавчання. Вони можуть допомогти у зменшенні ефекту перенавчання, коли модель адаптується надмірно до навчальних даних і втрачає здатність до узагальнення на нові дани. Поєднання прогнозів дозволяє створити біль універсальний ансамбль, який буде стійкий до перенавчання;

- покращення точності прогнозів. Ансамблеві методи можуть об’єднувати прогнози з різних моделей для отримання більш надійного та точного результату;

- робастність та її забезпечення. Ансамблеві методи можуть бути менш чутливими до випадкових варіацій у навчальних даних або шуму, що присутній в них. Цей пункт дуже важливий для нас з точки зору моделі для виявлення аномалій.

В даному розділі проведемо детальний аналіз різноманітних ансамблевих архітектур, а саме XGBClassifier, VotingClassifier, AdaBoostClassifier, RandomForestClassifier та GradientBoostingClassifier. Коротко розглянемо кожен з них:

XGBClassifier є частиною ансамблевого методу, класифікатор, що базується на алгоритмі градієнтного бустінгу, який включає побудову ансамблю дерев рішень. Він навчається ітеративно, додавши нові дерева до ансамблю і коригуючи прогнозу, щоб зменшити функцію втрат. Потужний і популярний алгоритм машинного навчання, який здатний працювати з різноманітними типами даних та вирішувати різні задачі.

Після проведених експериментів продемонструємо результати метрик у вигляді таблиць і рисунків.

Таблиця 2.1 – Метрики ефективності XGBClassifier

Train Score	Test Score	Accuracy	Precision	Recall	F1 Score
0.999942167	0.995082683	0.995082683	0.850198726	0.768425217	0.794582470

Таблиця 2.2 – Кількісна оцінка якості XGBClassifier

	precision	recall	f1-score	support
0	0.92	0.96	0.94	878773
1	1.00	1.00	1.00	27188627
2	1.00	1.00	1.00	6470759
3	1.00	1.00	1.00	2107532
4	0.85	0.80	0.82	283896
5	0.86	0.84	0.85	389289
6	0.53	0.23	0.32	19893
7	0.66	0.31	0.42	10467
accuracy			1.00	37349236
macro avg	0.85	0.77	0.79	37349236
weighted avg	0.99	1.00	0.99	37349236

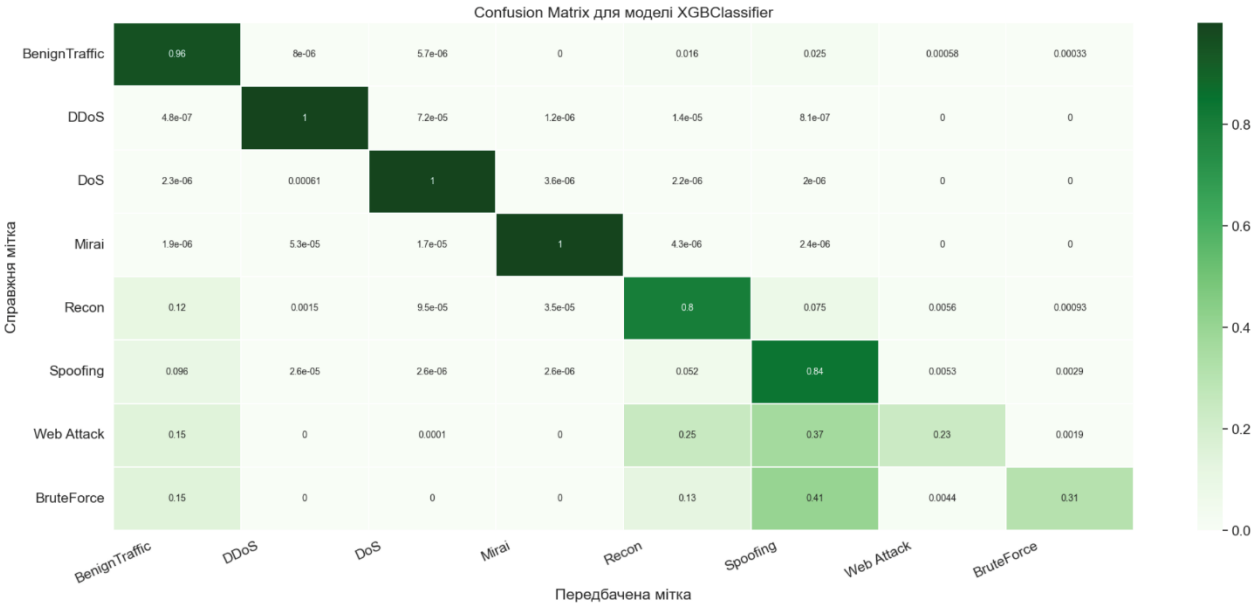


Рисунок 2.7 – Матриця помилок для моделі XGBClassifier

VotingClassifier – це класифікатор, який дозволяє об’єднувати прогнози декількох класифікаторів та приймати рішення за допомогою “голосування більшості”(hard voting) або за “вагомою схемою”(soft voting). У випадку “відбору за вагою”, кожен класифікатор може мати вагу і прогноз робиться на основі вагових прогнозів усіх класифікаторів. У випадку “відбору більшості”, прогнозом стає клас, який набрав найбільшу кількість голосів серед всіх класифікаторів.

В якості субкласифікаторів будемо використовувати: GaussianNB(Gaussian Naive Bayes) – простий, але ефективний алгоритм класифікації на основі методу наївного Баєса, який припускає, що всі ознаки в наборі незалежні між собою та мають нормальний розподіл. Він особоливо підходить коли дані мають велику кількість ознак і розмірність; LogisticRegression — алгоритм бінарної класифікації, який використовує логістичну функцію для передбачення ймовірностей належності об’єктів до одного з двух класів.

Один з найпоширеніших та добре вивчених алгоритмів у машинному навчанні; DecisionTreeClassifier – алгоритм машинного навчання для класифікації, який використовує дерево прийняття рішень для прогнозів. Широко використовується завдяки своїй простоті та здатності до автоматичного виконання важливого відбору ознак.

Таблиця 2.3 – Метрики ефективності VotingClassifier

Train Score	Test Score	Accuracy	Precision	Recall	F1 Score
0.988288381	0.986425933	0.986425933	0.807797248	0.782563142	0.764115346

Таблиця 2.4 – Кількісна оцінка якості VotingClassifier

	precision	recall	f1-score	support
0	0.83	0.96	0.89	878773

Продовження таблиці 2.4

1	1.00	0.99	1.00	27188627
2	0.97	0.99	0.98	6470759
3	1.00	1.00	1.00	2107532
4	0.89	0.73	0.80	283896
5	0.93	0.71	0.80	389289
6	0.16	0.61	0.26	19893
7	0.69	0.27	0.39	10467
accuracy			0.99	37349236
macro avg	0.81	0.78	0.76	37349236
weighted avg	0.99	0.99	0.99	37349236

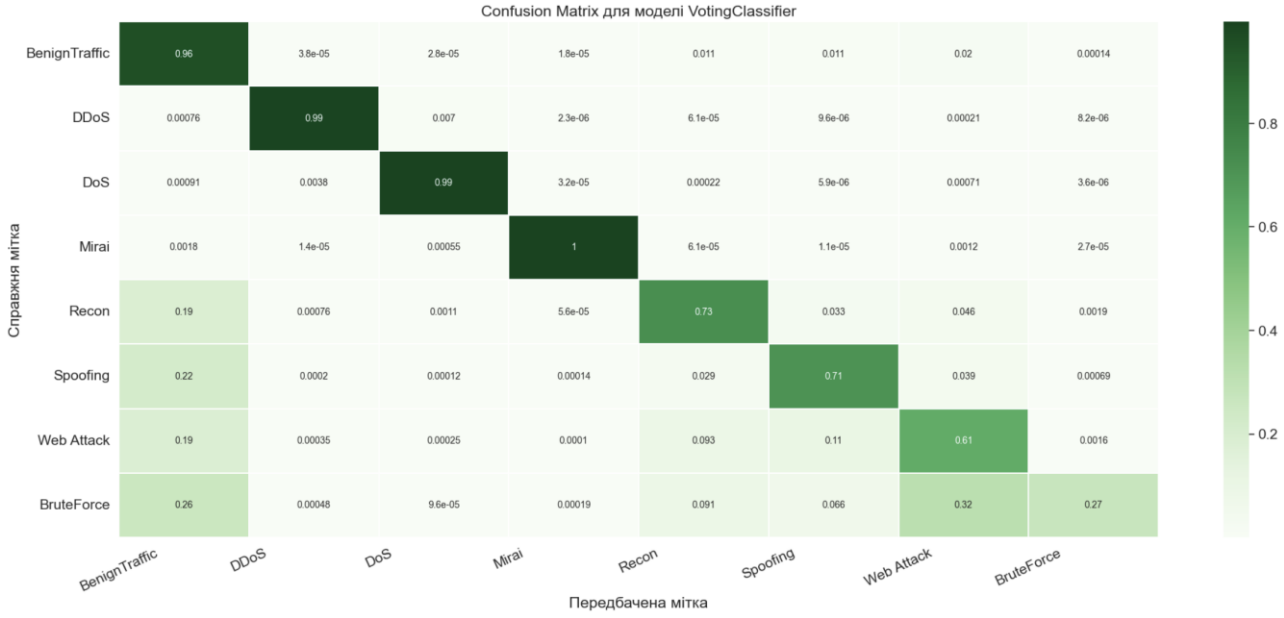


Рисунок 2.8 – Матриця помилок для моделі VotingClassifier

AdaBoostClassifier – один із найпоширеніших алгоритмів ансамблевого навчання у машинному навчанні. Його алгоритм використовує адаптивне зростання для побудови сильного класифікатора шляхом комбінування декількох слабких класифікаторів. Кожна наступна модель навчається на основі попередньої, зосереджуючись на таких прикладах, які попередні моделі класифікували неправильно. Це дозволяє сисематично покращувати свій прогноз, концентруючись на складних для класифікаціх прикладах. AdaBoost

надає ваги кожному прикладу в навчальному наборі. На початку всі приклади мають однакові ваги, а потім ці ваги адаптивно оновлюються після кожної ітерації навчання в залежності від того, як вони були класифіковані попередніми моделями.

Таблиця 2.5 – Метрики ефективності AdaBoostClassifier

Train Score	Test Score	Accuracy	Precision	Recall	F1 Score
0.975648765	0.975662474	0.975662474	0.79864038	0.743716979	0.727331686

Таблиця 2.6 – Кількісна оцінка якості AdaBoostClassifier

	precision	recall	f1-score	support
0	0.77	0.99	0.87	878773
1	0.99	1.00	0.99	27188627
2	1.00	0.98	0.99	6470759
3	0.96	0.97	0.96	2107532
4	0.78	0.72	0.74	283896
5	0.73	0.69	0.80	389289
6	0.21	0.58	0.26	19893
7	0.65	0.25	0.39	10467
accuracy			0.98	37349236
macro avg	0.80	0.74	0.72	37349236
weighted avg	0.98	0.98	0.97	37349236

RandomForestClassifier – модель ансамблевого методу, яка буде використовуватися у цій роботі. Детально буде розглянута а наступному розділі.

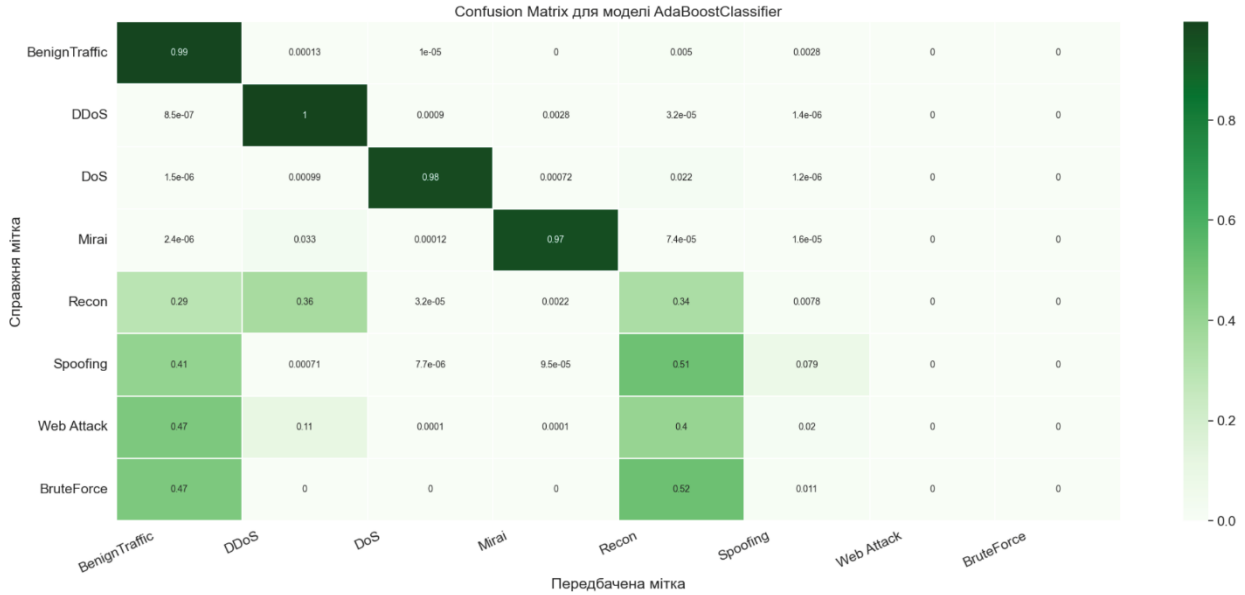


Рисунок 2.9 – Матриця помилок для моделі AdaBoostClassifier

GradientBoostingClassifier — алгоритм машинного навчання, з класу ансамбевих методів, спеціально до методів градієнтного бустінгу. Використовує градієнтний спуск для навчання послідовної серії слабких моделей, які підтримуються їхніми помилками. Один з найпотужніших алгоритмів у своєму класі і зазвичай демонструє високу точність та робастність. Його головна мета — мінімізувати функцію втрат за допомогою градієнтного спуску.

Використовує градієнт функції втрати для пошуку оптимальних параметрів моделі. Для попередження перенавчання підтримує різні методи регуляризації, такі як зменшення кроку, використання випадкових підвибірок та обмеження максимальної глибини дерев. Має вбудовану крос-валідацію, що дозволяє оцінити ефективність моделі та підібрати оптимальні гіперпараметри.

Таблиця 2.7 – Метрики ефективності GradientBoostingClassifier

Train Score	Test Score	Accuracy	Precision	Recall	F1 Score
0.999999250	0.994031310	0.994031310	0.855600616	0.850930646	0.852881582

Таблиця 2.8 – Кількісна оцінка якості GradientBoostingClassifier

	precision	recall	f1-score	support
0	0.96	0.88	0.92	878773
1	1.00	1.00	1.00	27188627
2	1.00	1.00	1.00	6470759
3	1.00	1.00	1.00	2107532
4	0.84	0.84	0.84	283896
5	0.89	0.85	0.87	389289
6	0.59	0.61	0.60	19893
7	0.58	0.63	0.61	10467
accuracy			0.99	37349236
macro avg	0.86	0.85	0.85	37349236
weighted avg	0.99	0.99	0.99	37349236

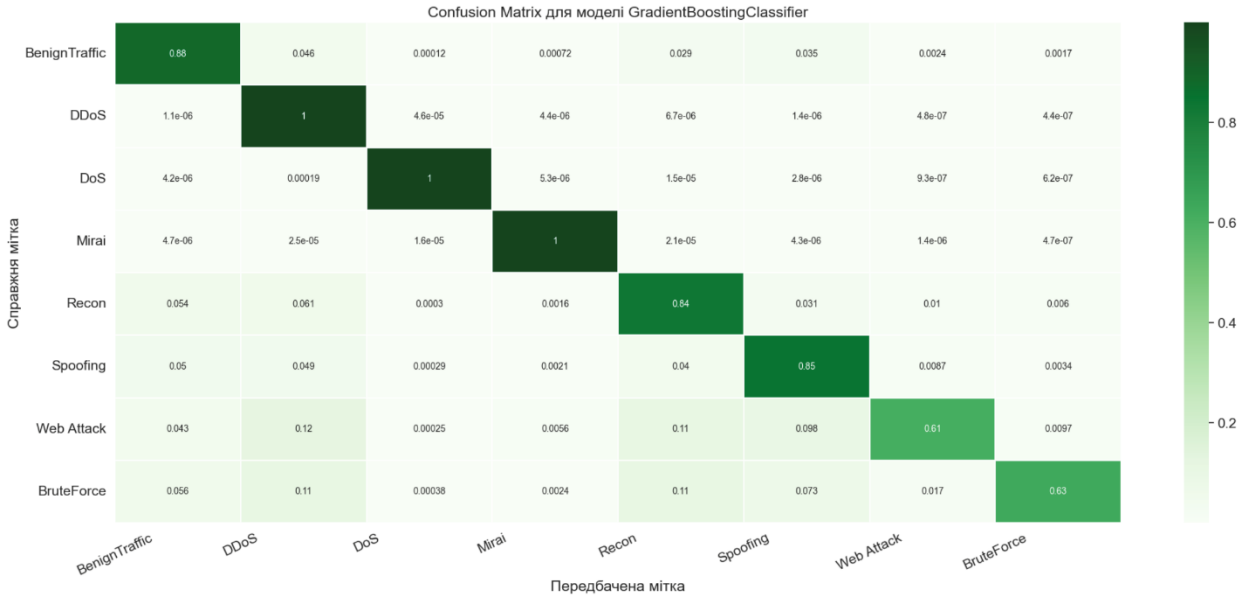


Рисунок 2.10 – Матриця помилок для моделі GradientBoostingClassifier

2.3 Архітектура мережі і оптимізація параметрів

В ході досліджень було вирішено використовувати модель ансамблевого методу RandomForest. Це метод машинного навчання, який використовується для класифікації та регресії. Він є типом і відповідає ряду класифікаторів дерева рішень на різних підвибірках набору даних. Використовує техніку випадковості і усереднення для підвищення точності прогнозування, покращення продуктивності та стабільності моделі. RandomForest включає в себе кілька дерев рішень, кожне з яких навчається на випадковій підмножині даних та ознак. Коли треба прийняти рішення, модель об’єднує прогнози всіх дерев, зазвичай за допомогою класифікації або середнього значення для регресії. Модель має властивість стійкості до перенавчання, оскільки кожне дерево навчається на випадковій підмножині даних та ознак. Це дозволяє ансамблю підтримувати генералізацію на нових, раніше не бачених ознаках [7]. Перед навчанням подбаємо про гіперпараметри та оптимізуємо їх за допомогою бібліотеки optuna. Для моделі RandomForest нас цікавлять:

- max_depth – максимальна глибина кожного дерева в ансамблі, яка визначає кількість рівнів у дереві рішень;
- max_features – визначає максимальну кількість ознак, які випадково обираються для розгляду при побудові кожного дерева в “лісі”;
- n_estimators – гіперпараметр, який вказує кількість дерев, які мають бути побудовані в ансамблі.

Коли гіперпараметри визначені і оптимізацію завершено, починається навчання, яке буде займати деякий час. Для оцінки ефективності моделі використовуємо метрики: Accuracy, Precision, Recall, F score, які враховують різні показники результатів класифікацій та дозволяють отримати більш повну картинку продуктивності моделі. Розберемо більш детально кожний з них:

Accuracy(правильність) – частка прогнозів, яку наша модель отримало правильно. Математично, це співвідношення між кількістю правильних прогнозів до загальної кількості прогнозів. Це корисно, коли всі класи мають

однакову важливість, як у нашому випадку але є недолік зі сторони незбалансованого набіру даних.

Таблиця 2.9 – Метрики ефективності моделі.

Train Score	Test Score	Accuracy	Precision	Recall	F1 Score
0.997416280	0.996321077	0.996321077	0.964647026	0.847356563	0.890306751

Таблиця 2.10 – Кількісна оцінка якості

	precision	recall	f1-score	support
0	0.91	0.98	0.94	878773
1	1.00	1.00	1.00	27188627
2	1.00	1.00	1.00	6470759
3	1.00	1.00	1.00	2107532
4	0.91	0.84	0.87	283896
5	0.92	0.86	0.89	389289
6	0.98	0.54	0.70	19893
7	0.99	0.57	0.72	10467
accuracy			1.00	37349236
macro avg	0.96	0.85	0.89	37349236
weighted avg	1.00	1.00	1.00	37349236

Precision(точність) – це співвідношення $\frac{TP}{TP+FP}$, де TP – кількість справжніх спрацьовувань, а FP – кількість хибних спрацьовувань [8].

Точність – це інтуїтивно зрозуміла здатність класифікатора не позначати негативний зразок як позитивний.

Recall(запам’ятовування) – це відношення $\frac{TP}{TP+FN}$, де TP – кількість справжніх позитивних результатів, а FN – кількість помилкових негативних результатів.

Запам’ятовування – це інтуїтивно зрозуміла здатність класифікатора знаходити всі позитивні зразки [5].

F-оцінка може бути інтерпретована як зважене гармонічне середнє значення точності та запам’ятовування, досягає найкращого значення при 1, а

найгіршого при 0. Визначається як: $F_1 = \frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

За підсумком навчання маємо результати наведенні в таблиці 2.9 та в таблиці 2.10.

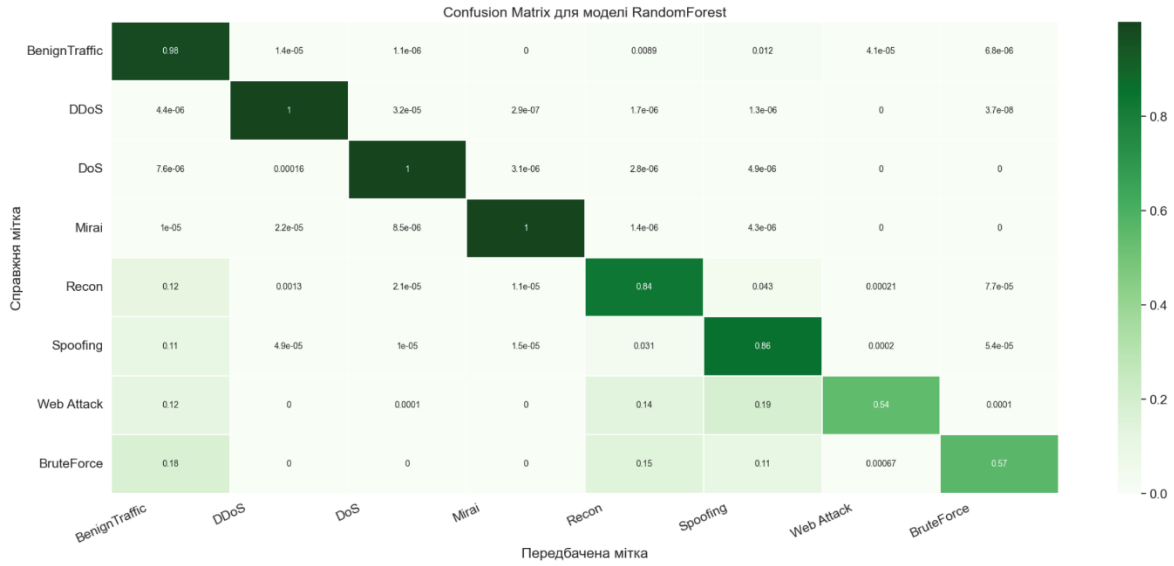


Рисунок 2.11 – Матриця помилок для моделі Random Forest

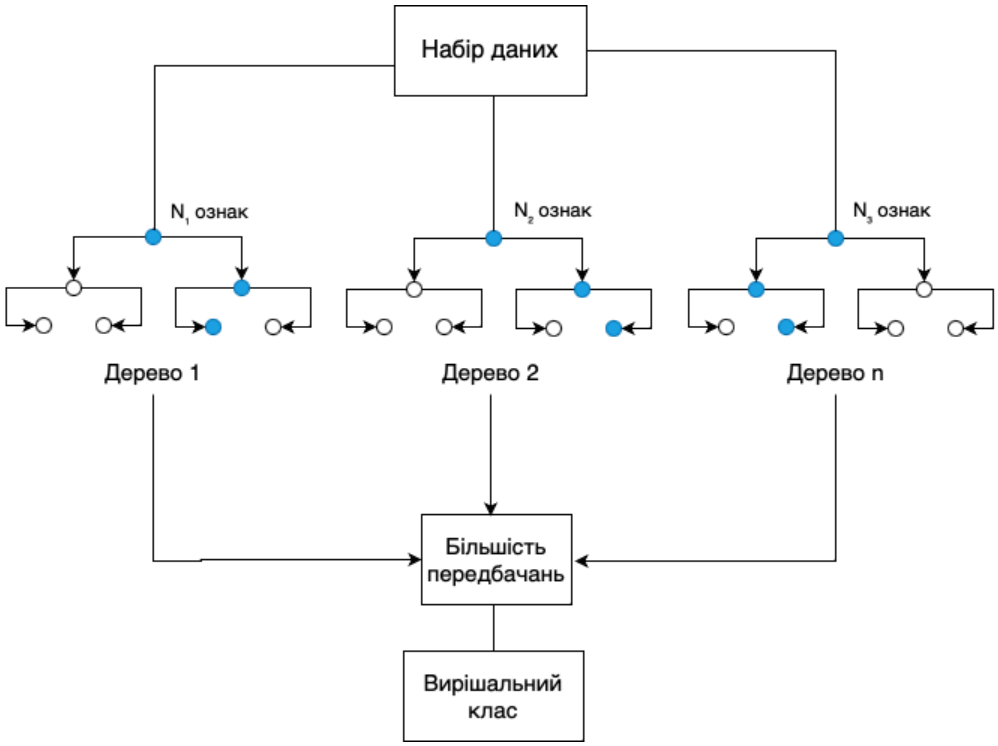


Рисунок 2.12 – Архітектура моделі Random Forest

Після тестування модель зберігається локально за допомогою бібліотеки pickle. На виході модель RandomForest, це набір дерев-предикторів $\{t(x_{in},\theta_n), n=1,...\}$, які індивідуально роблять передбачення на заданому параметрі x_{in}. Кожен предиктор залежить від випадкового набору змінних $\{\theta_n\}$, які незалежно відбираються з однаковим розподілом.

3 РЕАЛІЗАЦІЯ СИСТЕМИ З НЕЙРОННОЮ МЕРЕЖЕЮ

3.1 Реалізація додатку

Додаток буде в реальному часі зчитувати інтернет пакети або .pcap файли, діставити з них усі необхідні ознаки та відправляти моделі для отримання передбачень. За допомогою бібліотеки pyshark, для захоплення та аналізу інтернет-пакетів, витягуємо необхідні ознаки для подальшого використання. Після того як всі ознаки витягнуті, відправляємо їх до попередньо навченної моделі для отримання передбачень. Обробляємо результати передбачень та приймаємо рішення щодо подальших дій, сповіщення або вжиття інших заходів безпеки.

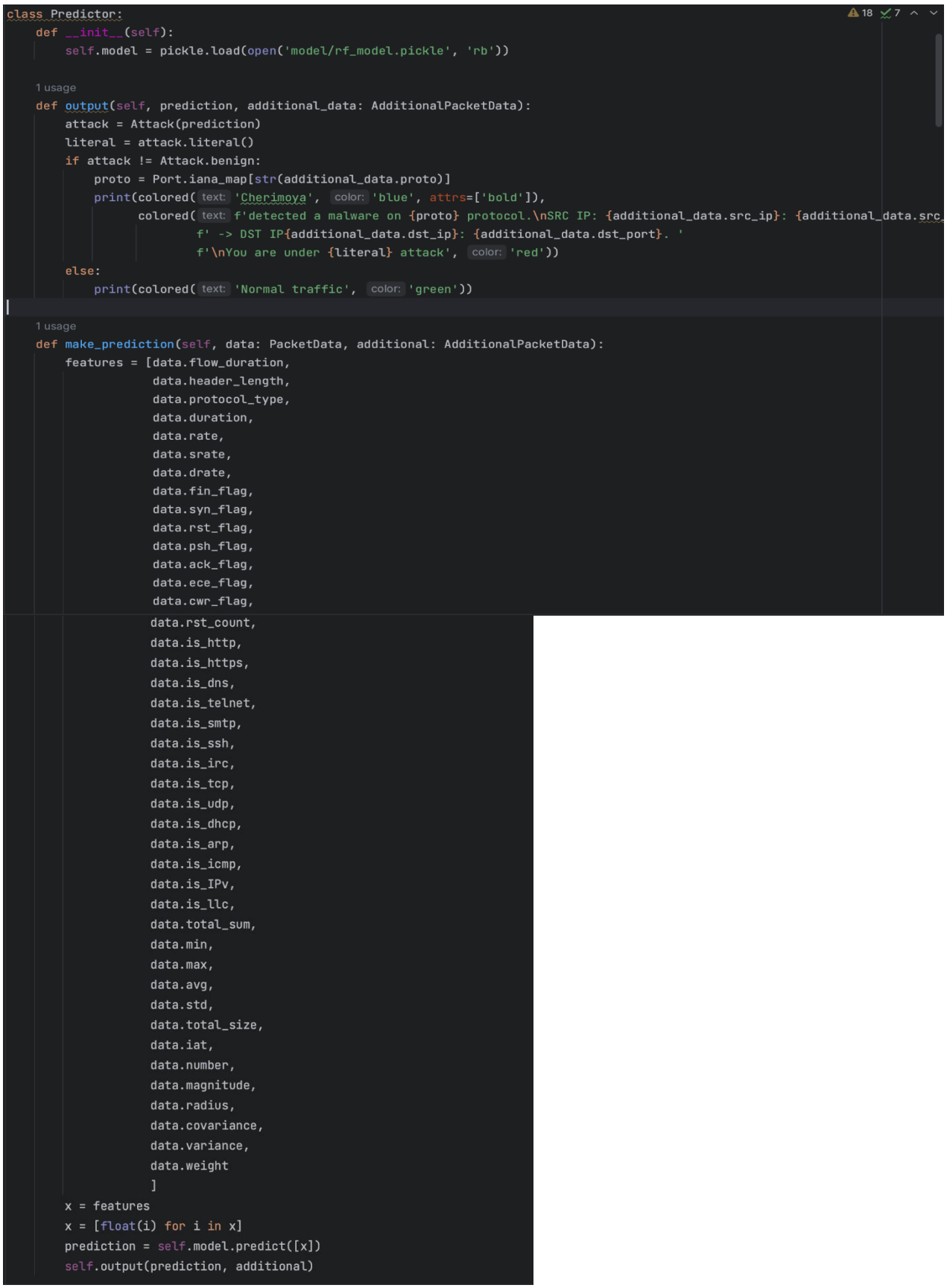
```python
def live_traffic_capture(interface=None):
    global capture
    if interface is None:
        interface = SnifferMode.online.interface
    capture = pyshark.LiveCapture(interface=interface, use_json=True, include_raw=True)
    for packet in capture.sniff_continuously():
        handle_sniffed_data(packet)


2 usages
def pcap_traffic(file=None):
    global capture
    if file is None:
        file = SnifferMode.offline.interface
    capture = pyshark.FileCapture(file, use_json=True, include_raw=True)
    for packet in capture:
        handle_sniffed_data(packet)


1 usage
def start_sniffer(mode):
    if mode == SnifferMode.online:
        live_traffic_capture()
    else:
        pcap_traffic()


1 usage
def cli(input_file, input_interface):
    if input_file is not None:
        pcap_traffic(file=input_file)
    else:
        live_traffic_capture(interface=input_interface)
```

Рисунок 3.1 – Скріншот з PyCharm IDE частини коду проекта який відповідає за зчитування пакетів з Ethernet інтерфейсу або з .pcap файлу



```
class Predictor:
    def __init__(self):
        self.model = pickle.load(open('model/rf_model.pickle', 'rb'))

    1 usage
    def output(self, prediction, additional_data: AdditionalPacketData):
        attack = Attack(prediction)
        literal = attack.literal()
        if attack != Attack.benign:
            proto = Port.iana_map[str(additional_data.proto)]
            print(colored( text: 'Cherimoya', color: 'blue', attrs=['bold']),
                  colored( text: f'detected a malware on {proto} protocol.\nSRC IP: {additional_data.src_ip}: {additional_data.src_
                           f' -> DST IP{additional_data.dst_ip}: {additional_data.dst_port}. '
                           f'\nYou are under {literal} attack', color: 'red'))
        else:
            print(colored( text: 'Normal traffic', color: 'green'))

    1 usage
    def make_prediction(self, data: PacketData, additional: AdditionalPacketData):
        features = [data.flow_duration,
                    data.header_length,
                    data.protocol_type,
                    data.duration,
                    data.rate,
                    data.srate,
                    data.drate,
                    data.fin_flag,
                    data.syn_flag,
                    data.rst_flag,
                    data.psh_flag,
                    data.ack_flag,
                    data.ece_flag,
                    data.cwr_flag,
                    data.rst_count,
                    data.is_http,
                    data.is_https,
                    data.is_dns,
                    data.is_telnet,
                    data.is_smtp,
                    data.is_ssh,
                    data.is_irc,
                    data.is_tcp,
                    data.is_udp,
                    data.is_dhcp,
                    data.is_arp,
                    data.is_icmp,
                    data.is_IPv,
                    data.is_llc,
                    data.total_sum,
                    data.min,
                    data.max,
                    data.avg,
                    data.std,
                    data.total_size,
                    data.iat,
                    data.number,
                    data.magnitude,
                    data.radius,
                    data.covariance,
                    data.variance,
                    data.weight
                    ]
        x = features
        x = [float(i) for i in x]
        prediction = self.model.predict([x])
        self.output(prediction, additional)
```

Рисунок. 3.2 – Скріншот з PyCharm IDE класу, який передає усі ознаки до навченої моделі для отримання передбачень

3.2 CLI системи виявлення вторгнень

Було вирішено розробити інтерфейс коммадного рядка (CLI – Command-line interface). Користувач зможе встановити цей додаток за допомогою системи керування пакунками(pip) на операційну систему Windows або класу Linux. Він матиме змогу запускати його виконання з командного рядка або додати в автоматичний запуск за допомогою скриптів.

Рисунок 3.3 – Архітектура *IDS*

```
# Cherimoya – Intrusion Detection System

> This is a project CLI performed like a intrusion detection system. Trained by the CIC IOT 2023
    dataset and ensemble classifier RandomForest. The IDS provides a notifications to the terminal
    window when you have a some malicious traffic.


### Installation
```sh
cd to cherimoya directory
python/python3 setup.py install
```

Usage
```sh
usage: cherimoya [-h] (-i INPUT_INTERFACE | -f INPUT_FILE)

options:
  -h, --help            show this help message and exit
  -i INPUT_INTERFACE, --interface INPUT_INTERFACE
                        Cherimoya will capture online packet data from
                        INPUT_INTERFACE
  -f INPUT_FILE, --file INPUT_FILE
                        Cherimoya will capture offline packet data from .pcap
                        INPUT_FILE
```

```
cherimoya -f /path/to/file.pcap
```

```
cherimoya -i en0
```

- Dataset Reference: https://www.unb.ca/cic/datasets/iotdataset-2023.html
```

Рисунок 3.4 – Посібник з використання README.md

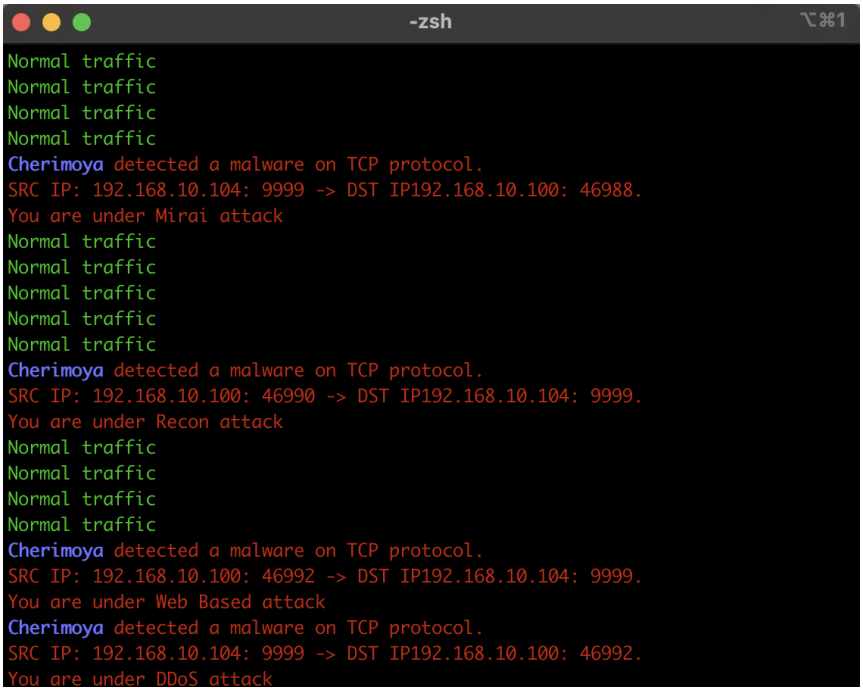


Рисунок 3.5 – Інтерфейс командного рядка (CLI) застосунку Cherimoya (назва системи виявлення вторгнень)

Після запуску програми користувач буде проінформований щодо доброякісного трафіку або шкідливого.

ВИСНОВКИ

У наслідок проведених досліджень і розглянутих концепцій передових систем та методів виявлення атак в інфраструктурі IoT, було запропоновано власну систему виявлення вторгнень в реальному часі, яка базується на методі виявленні аномалій та працює в комплексі з нейронної мережею ансамблевого методу RandomForest, яка була навчена на наборі з великим обсягом даних та має гарні показники: правильність – 0.996, точність – 0.964 запам’ятовування – 0.847, гармонічне середнє значення точності та запам’ятовування – 0.89. Для зручності використання застосунку системи був розроблений інтерфейс командного рядка, котрий сповіщає користувача або іншу систему про вторгнення чи атаку. В майбутньому запропонована модель може бути використана для систем побудованих в поєднанні з концепцією туманного обчислення(Fog Computing) та Інтернету речей, на принципах Fog-IoT архітектури.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. O. Liashenko, I. Velykodnyi, V. Znaidiuk, O. Zhurylo, “Модель та методи виявлення широкомасштабної атаки в середовищі iot”. Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2024. – Т. 1 (75). – С. 127-132. – DOI: https://doi.org/10.26906/SUNZ.2024.1.127.

2. T. Mazhar, D. B. Talpur, T. Al Shloul, Y. Y. Ghadi, I. Haq, I. Ullah, K. Ouahada, H. Hamam Analysis of IoT Security Challenges and Its Solutions Using Artificial Intelligence. 2023. DOI: https://doi.org/10.3390%2Fbrainsci13040683

3. A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman Survey of intrusion detection systems: techniques, datasets and challenges. 2019.
URL: https://cybersecurity.springeropen.com/articles/10.1186/s42400-019-0038-7

4. Рубан І. В. Класифікація методів виявлення аномалій в інформаційних системах / І. В. Рубан, В. О. Мартовицький, С. О. Партика // Системи озброєння і військова техніка. — 2016. — № 3. — С. 100-105

5. Verma Abhisher, Virender Ranga Machine learning based intrusion detection systems for IoT applications. 2020.
URL: https://link.springer.com/article/10.1023/A:1010933404324

6. J. Delua Supervised vs. Unsupervised learning. 2021.
URL: https://www.ibm.com/blog/supervised-vs-unsupervised-learning/

7. 1. I. U. Khan, M. Ouaissa, M. Ouaissa, Z. A. El Houda, M. Fazal Cyber Security for Next-Generation Computing. 2024.
DOI: https://doi.org/10.1201/9781003404361

8. Журило, О., Ляшенко, О. і Аветісова, К. 2023. ОГЛЯД РІШЕНЬ З АПАРАТНОЇ БЕЗПЕКИ КІНЦЕВИХ ПРИСТРОЇВ ТУМАННИХ ОБЧИСЛЕНЬ У ІНТЕРНЕТІ РЕЧЕЙ. СУЧАСНИЙ СТАН НАУКОВИХ ДОСЛІДЖЕНЬ ТА ТЕХНОЛОГІЙ В ПРОМИСЛОВОСТІ. 1 (23) (Квіт 2023), 57–71.
DOI: https://doi.org/10.30837/ITSSI.2023.23.057

9. V. Martovytskyi, I. Ruban, H. Lahutin, I. Ilina, V. Rykun and V. Diachenko, "Method of Detecting FDI Attacks on Smart Grid," 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2020, pp. 132-136,
DOI: 10.1109/PICST51311.2020.9468005

10. Система виявлення вторгнень.
URL: https://uk.wikipedia.org/wiki/Система_виявлення_вторгнень

11. Network-based Intrusion Prevention System.
URL:https://www.techopedia.com/definition/4030/network-based-intrusion-prevention-system-nips

12. Wireless Intrusion Prevention System.
URL:https://community.fs.com/article/wireless-intrusion-prevention-system-wips-boost-your-wifi-defenses-.html

13. Network Behavior Analysis.
URL: https://www.zenmor.com/docs/network-security-tutorials/what-is-nba

14. Host-based Intrusion Prevention System.
URL: https://uk.wikipedia.org/wiki/HIPS