ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«ПРОТИДІЯ КІБЕРЗАГРОЗАМ В УМОВАХ СТРІМКОГО РОЗВИТКУ ШТУЧНОГО ІНТЕЛЕКТУ»**

на здобуття освітнього ступеня магістр
за спеціальності 123 Комп'ютерна інженерія
(код,найменування спеціальності)
освітньо-професійної програми Комп'ютерні системи та мережі
(назва)

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

_____ Андрій ПОЛІЩУК
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав: здобувач вищої освіти гр.КСДМ-61
Андрій ПОЛІЩУК
(ім'я, ПРІЗВИЩЕ)
Керівник: Андрій ЛЕМЕШКО
Доктор філософії, (PhD) *(ім'я, ПРІЗВИЩЕ)*
Рецензент:
(ім'я, ПРІЗВИЩЕ)

Київ 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерної інженерії
Ступінь вищої освіти «Магістр»
Спеціальність 123 Комп'ютерна інженерія
Освітньо-професійна прорама Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедрою Комп'ютерної інженерії
Наталія ЛАЩЕВСЬКА
(ім'я, ПРІЗВИЩЕ)
"____" _____2023 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Поліщуку Андрію Руслановичу
(прізвище, ім'я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Протидія кіберзагрозам в умовах стрімкого розвитку штучного інтелекту
керівник роботи Андрій ЛЕМЕШКО доктор філософії (PhD)
(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від "19" 10 2023 р. №145
2. Строк подання кваліфікаційної роботи 22.12.2023р.
3. Вихідні дані кваліфікаційної роботи:
3.1. Штучний інтелект;
3.2. Машинне навчання;
3.3. Програмне забезпечення;
3.4. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
4.1. Огляд основних понять кібербезпеки та особливості застосування штучного інтелекту в кібербезпеці.
4.2. Дослідження систем виявлення вторгнень на основі хоста аналізу тексту та машинного навчання.
4.3. Експерементальні дослідження виявлення вторгнень на основі інтелектуальної системи безпеки.

5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання “ 20 ” жовтня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури по тематиці магістерської роботи	20.10.2023р. 28.10.2023р.	Виконано
2.	Огляд основних понять та застосування штучного інтелекту в кібербезпеці	28.10.2023р. 08.11.2023р.	Виконано
3.	Дослідження систем виявлення вторгнень на основі хоста аналізу тексту та машинного навчання	08.11.2023р. 17.11.2023р.	Виконано
4.	Виявлення вторгнень на основі інтелектуальної системи безпеки за допомогою методів машинного навчання	17.11.2023р. 26.11.2023р.	Виконано
5.	Оформлення висновків, реферат, оформлення роботи.	26.11.2023р. 08.12.2023р.	Виконано
6.	Розробка демонстраційних матеріалів, написання доповіді	08.12.2023р. 17.12.2023р.	Виконано

Здобувач вищої освіти _____ Андрій ПОЛІЩУК
(підпис) *(ім’я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Андрій ЛЕМЕШКО
(підпис) *(ім’я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступня магістр: 80 стор., 43 рис., 29 табл., 24 джерел.

Мета роботи – дослідження шляхів покращення механізму безпеки із застосуванням методів машинного навчання для виявлення кібер-атак.

Об'єкт дослідження – процес виявлення кібер-атак.

Предмет дослідження – методи машинного навчання для виявлення кібер-вторгнень.

Короткий зміст роботи: Виконуючи поставлені завдання у роботі проаналізовано основні поняття кібербезпеки, технології кібербезпеки та приведено приклад маштабних кібератак на підприємства.

Виконано дослідження особливостей застосування штучного інтелекту в кібербезпеці, проаналізоввно прикладні методи машинного навчання та наведено результати експериментальних досліджень застосування методів машинного навчання в кібербезпеці.

У третьому розділі досліджено виявлення вторгнень на основі інтелектуальної системи безпеки за допомогою методів машинного навчання, яка призначена для виявлення останніх зловмисних URL-адрес і розширену для атак розподіленої відмови в обслуговуванні (DDoS). Проведено експерементальні дослідження та оцінка продуктивності досліджуваної системи SIS-ID.

КЛЮЧОВІ СЛОВА: КІБЕРБЕЗПЕКА, АТАКА, МЕТОД ПРОТИДІЇ, МАШИННЕ НАВЧАННЯ, ХОСТ, КІБЕРЗЛОЧИН, РИЗИК, ЗАГРОЗА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ТУНЕЛЮВАННЯ, АНСАМБЛЬ.

ABSTRACT

The text part of the qualification work for obtaining a master's degree: 80 pages, 43 figures, 29 tables, 24 sources.

The purpose of the work is to research ways to improve the security mechanism using machine learning methods to detect cyber attacks.

The object of research is the process of detecting cyber-attacks.

The subject of research is machine learning methods for detecting cyber intrusions.

Brief content of the work: By fulfilling the assigned tasks, the work analyzed the main concepts of cyber security, cyber security technologies and gave an example of large-scale cyber attacks on enterprises.

The study of the peculiarities of the use of artificial intelligence in cyber security was carried out, the applied methods of machine learning were analyzed and the results of experimental studies of the use of machine learning methods in cyber security were given.

The third section explores intrusion detection based on an intelligent security system using machine learning techniques, which is designed to detect recent malicious URLs and extended to distributed denial of service (DDoS) attacks. Experimental studies and performance evaluation of the SIS-ID system under study were conducted.

KEY WORDS: CYBERSECURITY, ATTACK, COUNTERMETHOD, MACHINE LEARNING, HOST, CYBERCRIME, RISK, THREAT, SOFTWARE, TUNNELING, ENSEMBLE.

ЗМІСТ

ВСТУП……...9
РОЗДІЛ 1 ОГЛЯД ОСНОВНИХ ПОНЯТЬ ТА ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В КІБЕРБЕЗПЕЦІ…….11
1.1 Аналіз основних понять кібербезаеки……..11
1.2 Огляд термінів та технологій кібербезпеки та їх класифікація…….13
1.3 Особливостей застосування методів машинного навчання в кібербезпеці……...23
1.4 Система виявлення вторгнень на основі методів машинного навчання……....34
РОЗДІЛ 2 ДОСЛІДЖЕННЯ СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ ХОСТА АНАЛІЗУ ТЕКСТУ І МАШИННОГО НАВЧАННЯ……41
2.1 Аналіз архітектури Host-based Intrusion Detection System……..42
2.2 Огляд методу представлення ознак……..46
2.3 Прикладні методи машинного навчання та попередні відомості про класифікацію…….52
2.4 Експериментальні результати та їх обговорення……56
РОЗДІЛ 3 ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ БЕЗПЕКИ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ…….62
3.1 Загальний огляд та вимоги до інтелектуальних систем безпеки виявлення вторгнень……62
3.2 Аналіз реалізації даних і функцій…….63
3.3 Методологія навчання для системи SIS-ID……..64
3.4 Експерементальні дослідження та оцінка продуктивності SIS-ID……70
3.5 Апаратне моделювання в реальному часі……85
ВИСНОВКИ……88
ПЕРЕЛІК ПОСИЛАНЬ……...89
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)……..92

ВСТУП

Актуальність теми. В даний час зростання мережевих загроз представляє собою складну проблему проти механізму захисту в мережах для захисту конфіденційних даних і його компонентів. Тому інтелектуальним системам безпеки буде рекомендовано розгортати їх за допомогою методів машинного навчання проти постійно зростаючих кіберзагроз. Машинне навчання стало вирішальною технологією кібербезпеки для захисту комп'ютерних мереж і систем від кіберзлочинців.

Мета роботи – дослідження шляхів покращення механізму безпеки із застосуванням методів машинного навчання для виявлення кібер-атак.

Для виконання поставленої мети, у магістрській роботі розроблено та виконано наступні завдання:

- огляд основних понять кібербезпеки та особливості застосування штучного інтелекту в кібербезпеці;

- дослідження систем виявлення вторгнень на основі хоста аналізу тексту та машинного навчання;

- експеременталні дослідження виявлення вторгнень на основі інтелектуальної системи безпеки.

Об'єкт дослідження – процес виявлення кібер-атак.

Предмет дослідження – методи машинного навчання для виявлення кібер-вторгнень.

Методи дослідження. Для вирішення поставленої мети у магістерській кваліфікаційній роботі використані методи математичного моделювання, теорії множин, теорії ймовірностей і математичної статистики, теорії захисту інформації.

Джерела дослідження:

- https://www.altamira.ai/blog/artificial-intelligence-and-its-use-in-cybersecurity;
- https://www.kaspersky.com/resource-center/definitions/ai-cybersecurity;
- https://www.sciencedirect.com/science/article/abs/pii/S2214785320346861;
- https://arxiv.org/pdf/2203.04686.pdf;

- https://theses.hal.science/tel-02952954/document.

Наукова новизна одержаних результатів. У магістрській кваліфікаційній роботі досліджено застосування методики тестування на проникнення, яка дозволяє виявити вразливості для найпопулярніших атак.

Практична значущість одержаих результатів. Отримані результати можуть бути використані IT-адміністраторами як орієнтир для захисту системи від загроз кіберзлочинців.

Апробація результатів магістерської роботи. Основні положення і результати магістерської роботи доповідались і обговорювались на двох науково-практичних конференціях.

Публікації. За матеріалами роботи опубліковано одну статтю у науковому журналі.

РОЗДІЛ 1 ОГЛЯД ОСНОВНИХ ПОНЯТЬ ТА ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В КІБЕРБЕЗПЕЦІ

1.1 Аналіз основних понять кібербезаеки

В останні роки використання мережевих систем і взаємопов’язаних програм в Інтернеті стало свідком величезного розширення та перетворилося на невід’ємну частину нашого життя та використовується в усіх верствах суспільства. Із зростанням кібербезпеки глибокий аналіз загроз став основною вимогою, а також необхідністю. Його назва походить від двох взаємопов'язаних слів; кібер, що пов’язано з відповідною технологією, яка охоплює все мережеве обладнання, таке як системи та програми, і безпека, яка пов’язана з методологією безпеки, яка включає мережеву безпеку, системну та інформаційну безпеку.

Зі зростанням кіберзлочинності та труднощами обробки великих обсягів даних, що надходять до мереж, стан справ вимагає подальших методів протидії для збереження фундаментального захисту від кібератак [1]. Таким чином, підвищення безпеки стає більш значним завдяки здатності кібер-зловмисників використовувати кілька потенційних способів проникнення в конфіденційність інфраструктури, включаючи апаратне та програмне забезпечення для застосування незаконних дій і проникнення щодо цінної інформації [2]. Покращення кібербезпеки відіграє важливу роль у досягненні безперервних зусиль для захисту онлайн-інформації та обчислювальних ресурсів від несанкціонованого використання чи загрози. Крім того, він спеціалізується на захисті хостингової та мережевої інфраструктури від пошкодження внаслідок хакерських і атакуючих дій і шкідливого програмного забезпечення, такого як віруси, хробаки та трояни.

Кібербезпека використовує набір технічних методів і процесів, призначених для захисту мереж, комп’ютерного програмного забезпечення та даних від атак. Це запобігає незаконним цілям із несанкціонованим доступом до інформації, яка використовується для зміни або знищення кіберсистем [3]. Системи кібербезпеки складаються з кількох компонентів безпеки, найпоширеніший з яких може бути

пов’язаний із безпекою мережі або локальними системами безпеки. Ці пристрої часто містять пропонований рівень систем безпеки, наприклад брандмауер, деякі шкідливі антивірусні програми та системи виявлення вторгнень (IDS). Останнім часом ці технології були доповнені методами штучного інтелекту та машинним навчанням, щоб надійно виявляти проникнення, а також фундаментальними процедурами аналізу мереж.

Домени кібербезпеки. Кібербезпека стосується заходів захисту, які полегшують і зміцнюють процес забезпечення новітніх технологій кібербезпеки. Таким чином, основна вимога була запропонована шляхом узгодження її з іншими областями бар’єрного захисту [4]. Відносини між різними кібер-доменами викладені нижче та проілюстровані на рис.1.1.

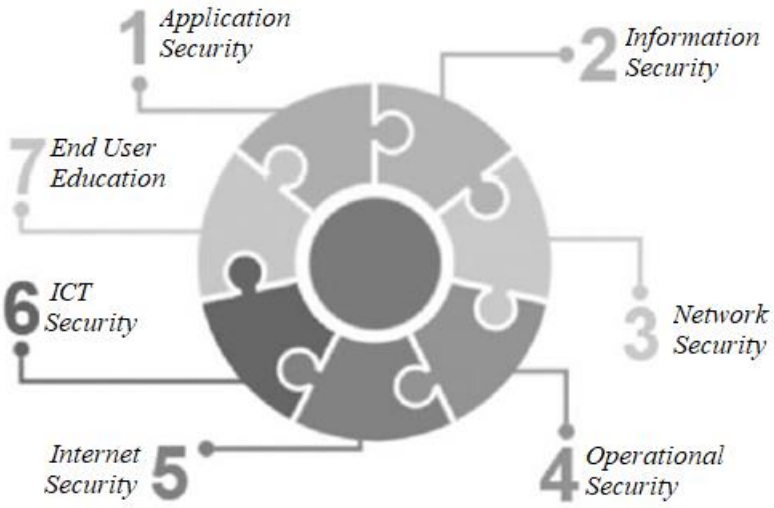


Рисунок 1.1 - Кібербезпека та її різні сфери

Важливість кібербезпеки. Кібербезпека має кілька підходів із спільними цілями – захистити та зберегти інформацію від пограбування чи нападу. Три основні цілі: конфіденційність для збереження конфіденційної інформації, цілісність для забезпечення надійності даних і доступність для забезпечення доставки інформації. Тріада конфіденційності, цілісності, доступності (CIA) є важливою та вважається базовими правилами для всіх систем безпеки, щоб керувати загальною політикою та повністю захищати дані всередині організацій [4]. Тому необхідно визначити кілька рівнів захисту даних з точки зору

забезпечення трійки CIA. Основні інструменти можна класифікувати, як перелічено в табл.1.1.

Таблиця 1.1- Найважливіші інструменти відповідно до тріади CIA

Трійка CIA	Інструмент	Мета
Конфіден-ційність	Шифрування та дешифрування	Захист конфіденційних і цінних даних, таких як банківські номери кредитів і транзакції електронної комерції.
	Управління доступом	Визначення політик доступу до систем, фізичних компонентів і ресурси віртуальних мереж, надаючи користувачам привілеї доступу та дозволи.
	Аутентифікація	Надання дозволу на підтвердження ідентифікації користувача для будь-якого процесу автентифікації.
	Авторизація	Надання користувачеві дозволу щодо його/її поведінки пов’язані з обмеженим механізмом безпеки для авторизованого доступу до системних ресурсів за допомогою попередньо визначених політик.
Цілісність	Резервні копії	Зберігання даних періодично або автоматично в системі керування базами даних.
	Контрольні суми	Забезпечення цілісності переданих даних між мережами за допомогою математичної функції для відображення вмісту файлу в числовому значенні.
	Корекція даних	Виявлення неочікуваних змін шляхом збереження автентичних даних і ідентифікації неоригінальних між собою.
Доступність	Фізичні засоби захисту	Захист компонентів інфраструктури для передачі своїх ресурсів, які зберігаються в безпечній зоні, щоб зберігати доступні дані.

1.2 Огляд термінів та технологій кібербезпеки та їх класифікація

Як правило, термін уразливості походить від виявлення слабких місць, які привертають кіберзлочинців до отримання несанкціонованого доступу або

виконання незаконних дій у системах. Він може бути використаний зловмисниками для введення шкідливого коду, доступу до пам’яті, інсталяції черв’яків, надсилання зловмисного програмного забезпечення, крадіжки особистих даних і доступу до критичних даних. Вразливі місця в Інтернеті включають слабке програмне забезпечення або збій у веб-програмі.

Крім того, уразливості, які були виявлені протягом попередніх десятиліть у багатьох компаніях, виявляються через відсутність перевірки або контролю введення полів. Неправильна конфігурація веб-сервера, його систем і компонентів є основною ціллю, яка може бути використана для атаки на захист будь-якої системи. Різноманітність цих вразливостей присутня в багатьох веб-додатках, оскільки вони виникають у кількох випадках і різного роду. Крім того, уразливості веб-безпеки слід виявляти за допомогою попередньо визначеного плану, і їх можна прийняти, дотримуючись запропонованих кіберправил, які щороку перераховуються OWASP Top 10, який є кіберстандартним документом для розробників, який охоплює найбільш критичні ризики безпеки, пов’язані з Інтернетом. безпека програми, наприклад помилка перевірки введення та доступу; Обробка помилок виняткових умов, помилка конфігурації, помилка дизайну, ін’єкція, порушення автентифікації, розкриття конфіденційних даних, порушення контролю доступу, неправильні налаштування безпеки, міжсайтовий сценарій (XSS), використання компонентів із відомими вразливими місцями, недостатнє ведення журналів і моніторинг [5].

Технології безпеки. Кібербезпека стала предметом інтересу для організацій і компаній у кіберсвіті. Насправді, технологічні інструменти, засновані на даних, мають велику обізнаність через проникнення в їх конфіденційність безпеки. Мережі організацій стають надзвичайно доступними, і з точки зору безпеки зростає проблема безпеки. Сьогодні головне усвідомлення у сфері кібербезпеки полягає в тому, щоб виключити спроби піратства над захисними компонентами з метою порушити конфіденційність. Кожна організація, що базується на інтернет-сервісах, потребує фундаментальної стратегії безпеки, такої як виявлення, аналіз та запобігання з’ясуванню кіберзлочинів. Дійсно, найважливіші методи кібербезпеки,

такі як брандмауер і VPN, система виявлення вторгнень (IDS), машинне навчання на основі IDS і система запобігання вторгненням (IPS), перераховані нижче:

1. Брандмауер і VPN. Вони створюють міцний бар'єр між мережевими компонентами для забезпечення послуг захисту. Зазвичай брандмауер встановлюється з усіх сторін мережі або підмережі, щоб захистити її від будь-якої загрози ззовні. Отже, Steven et al. [6] стверджують, що мережевий брандмауер - це сукупність елементів, розташованих між внутрішньою та зовнішньою мережею. Він покращує кібербезпеку, контролюючи весь прохідний трафік, щоб точно визначити, хто є авторизованим, щоб дозволити йому проходити між мережами. Брандмауер перевіряє кожен пакет, що надходить, і може блокувати типи, які не розпізнаються за певними критеріями кібербезпеки. З іншого боку, VPN — це віртуальний зашифрований тунель, розташований між мережею та віддаленим сервером, який використовується службами VPN. Зовнішній вхідний трафік зазвичай слід направляти через захищений тунель за допомогою ключа автентифікації, щоб отримати доступ через його сервер для доступу до потрібних ресурсів компанії. Таким чином, дані будуть у безпеці та захищені від зловмисників. Крім того, IP-адреса реального користувача стає VPN-сервером, який дозволяє приховати поточну ідентифікацію [7]. Однорідність між брандмауером і VPN можна розглядати як вимогу до підвищення кібербезпеки. Брандмауери є шлюзами для підтвердження безпеки у внутрішній мережі, тоді як VPN є правилами доступу до внутрішньої мережі. VPN рекомендується впроваджувати там, де розміщення мережевого брандмауера забезпечує безпеку мережевого трафіку. Відсутність брандмауера робить методи шифрування VPN марними. На рис.1.2 показано наявність брандмауера та VPN, тоді як сервер VPN розташований в Інтернеті попереду брандмауера.

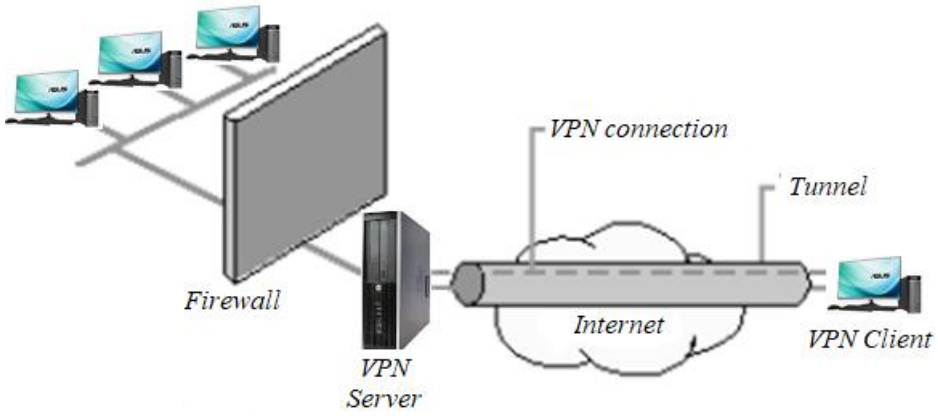


Рисунок 1.2 - Брандмауер і архітектура VPN у мережі

2. Система виявлення вторгнень (IDS). Вторгнення — це несанкціонована діяльність, яка завдає шкоди системі даних та інформації. Це вважається незаконною діяльністю, яка становить пряму загрозу для інформації ЦРУ, і тому розглядається як шкідливе втручання в мережі та системи. Таким чином, використання системи виявлення вторгнень (IDS), яка є інструментом, який ідентифікує зловмисні та шкідливі дії в комп'ютерних системах і мережах, сприяє підтримці безпеки систем. Він визначає типи зловмисного мережевого трафіку, який не міг ідентифікувати традиційний брандмауер у мережі. IDS відіграє важливу роль у будь-якому механізмі кібербезпеки. Він забезпечує динамічний моніторинг через хост і мережевий трафік, і це необхідно для досягнення кіберзахисту від атак. Система виявлення вторгнень попереджає IT-адміністратора, коли хтось намагається порушити конфіденційність мережі, а встановлений брандмауер отримує доступ до їхніх компонентів за допомогою надійних аспектів.

3. Системи виявлення вторгнень IDS на основі методів машинного навчання. Інтелектуальна система виявлення вторгнень є однією з найбільших проблем у центрах кібердосліджень. Потреба в використанні ефективної IDS з використанням машинного навчання стала надзвичайно важливою для виявлення нових і просунутих атак, які не виявляють традиційні IDS і брандмауер. Крім того, IDS ретельно перевіряє бажану мережу, щоб уникнути проникнення цінної інформації, на яку зловмисники зосереджуються під час цього проникнення. Наприклад, він створює процес моніторингу в режимі реального часу, а також керування

інцидентами, щоб створити бар’єр безпеки, пов’язаний із подіями, зібраними з мереж, пристроїв безпеки, систем і програм. Тим не менш, він забезпечує робочий процес, який відстежує та посилює інцидент. Крім того, IDS можна використовувати різними способами, такими як системи виявлення вторгнень на основі мережі або хосту, керування журналами, а також для створення звітів аналізу з метою розслідування.

4. Система запобігання вторгненням (IPS)**.** Зростання об’єднаних мереж стає все більш критичним і небезпечним питанням. Раніше звичайні типи систем захисту, брандмауер та інші типи засобів захисту здебільшого були достатніми для лікування традиційних кібер-технологій. Однак потреба в реалізації нового підходу до захисту вважається однією з найважливіших проблем дослідників безпеки, які докладають зусиль для реалізації системи запобігання вторгненню. Вважається інтеграцією між IDS, міжмережевими екранами та іншим видом бар’єрного захисту. Загалом функціональні можливості IPS такі ж, як і IDS, але потребують запобіжних заходів. Крім того, порівняно з IDS, IPS можна класифікувати на два основних типи; систему запобігання вторгненням на основі хоста (HIPS) і систему запобігання вторгнень на основі мережі (NIPS).

Кіберзлочини. Кіберзлочинність – це вид злочинної діяльності, яка відбувається в бастіоні кіберпростору. Відповідно до звіту про статистику вразливостей за 2021 рік, у їхніх установах понад 60% дослідників і техніків працюють у сфері систем безпеки. Вони докладають зусиль для перевірки частоти помилкових спрацьовувань, виявляючи повідомлені вразливості, що може зайняти більше 3 годин на день. Крім того, згідно з онлайн-опитуванням 2019 року, взятим з Info security Europe, виявляється, що не дивно, що нинішній брак навичок кібербезпеки стикається з посадовими особами служби безпеки, які відповідали на це дослідження. Було показано, що лише 32% працюють зі своїм величезним потенціалом, а 68% залишаються, оскільки фактори вимагають більшої кількості спеціалістів, які з’явилися для керування безпекою своїх кіберданих, щоб вони ефективніше працювали з випадками вразливостей. У звіті було доведено відсоток загальних прогалин, оскільки вони проаналізували та пов’язали інтенсивність

ризиків, які застосовуються для установ, щоб охопити як малі, так і середні компанії та охопити їх. Більше того, виявилося, що невеликі компанії з 11-100 співробітниками представляють середній відсоток ризиків усіх прогалин, який становить 4,1%, оскільки такі організації не мають великого цифрового простору, щоб вважатися такими, що легко піддаються атаці. Крім того, для великих організацій ризики значною мірою схожі за пропорціями. Наприклад, організації з понад 100 співробітниками можуть мати подібну щільність ризику. Крім того, на рис.1.3 показано, що існує 14% уразливостей, класифікованих як високий або критичний ризик для компаній, які охоплюють від 101 до 1000 співробітників.

Рисунок 1.3 - Відсоток розподілу ризиків за типами компаній

Однак 11,5% вразливостей класифікуються як високі або критичні ризики для компаній, які включають від 1001 до 10000 співробітників. З іншого боку, 11% вразливостей класифікуються як високий або критичний ризик для компаній з 10000+ співробітниками.

Найпопулярніші кіберзлочини. Насправді кожна країна має свої правила боротьби з кіберзлочинністю, які забороняють викрадення центру обробки даних, злом і шахрайство. Кіберзлочинці мотивуються декількома підходами, включаючи

фінансові прибутки, емоційну незахищеність, суспільні критерії, відсутність правових актів і дефіцит кібер-покарань [10]. Крім того, за останні десятиліття кіберзлочини в компаніях зросли за частотою та тяжкістю. Кіберзлочини торкнулися мільярдів і більше користувачів у багатьох секторах і змусили компанії заплатити понад сотні мільйонів доларів. За останні десятиліття були виявлені такі відомі типи кіберзлочинів, як зловмисна загроза, крадіжка особистих даних і шахрайство, хакерство, соціальна інженерія та програми-вимагачі.

За даними дослідницького відділу Statista, дослідження, представлене в 2016 році, показало, що у Франції майже 40% респондентів зазнали атаки через викрадення особистих даних через їхні профілі в соціальних мережах за допомогою кіберзлочинців соціальної інженерії. Крім того, згідно з дослідженням Центру скарг на злочини (IC3) у 2019 році звіт показує, що IC3 отримав 2047 скарг, які ідентифіковані як програми-вимагачі, і 8,9 мільйонів доларів було втрачено через атаки програм-вимагачів. У табл.1.2. наведено список найбільших корпоративних кіберзлочинів за останні роки, однак на рис.1.4 показано 20 найбільших міжнародних жертв із їх кількістю випадків для кожної країни у 2019 році.

Таблиця 1.2 - Список найбільших корпоративних кіберзлочинів

Компанія	Рік	Опис атаки	Довідка
Marriott	2018	Ця атака сталася, коли зловмисник отримав несанкціонований доступ до системи бронювання Marriott. Було викрадено інформацію про 500 мільйонів клієнтів, пов’язану з їхніми іменами, банківськими картками та паспортами. Це порушення спричинило штраф у розмірі 124 мільйонів доларів через збій захисту персональних даних клієнтів.	CNN
Facebook	2019	Згідно зі звітом UpGuardIn, дослідники безпеки виявили великий збір даних серед користувачів Facebook. Вони публічно з’являлися в хмарі серверів Amazon для завантаження без будь-якого дозволу. Ця атака спричинила штраф у розмірі 5 мільярдів доларів через втрату контролю над великими масивами даних приватних користувачів.	CNN

Продовження таблиці 1.2 - Список найбільших корпоративних кіберзлочинів

Компанія	Рік	Опис атаки	Довідка
Capital One	2019	У 2019 році сервер Capital One (COF) був зламаний зловмисником. Порушення дозволило отримати 140 000 номерів соціального страхування, а також один мільйон номерів канадського страхування, 80 000 облікових записів, пов'язаних з банківським сектором, і невідому кількість особистої інформації, такої як імена, адреси, кредитні рейтинги та іншу інформацію.	CNN
Mitsubishi	2020	Mitsubishi заявляє, що незаконне угруповання атакувало компанію Mitsubishi за допомогою масштабної кібератаки, витягнувши цінну інформацію про 8000 осіб, а також важливі дані для партнерських компаній. Більше того, злом скомпрометував чутливі державні установи, які містили секретні проекти щодо обладнання захисту.	CSIS

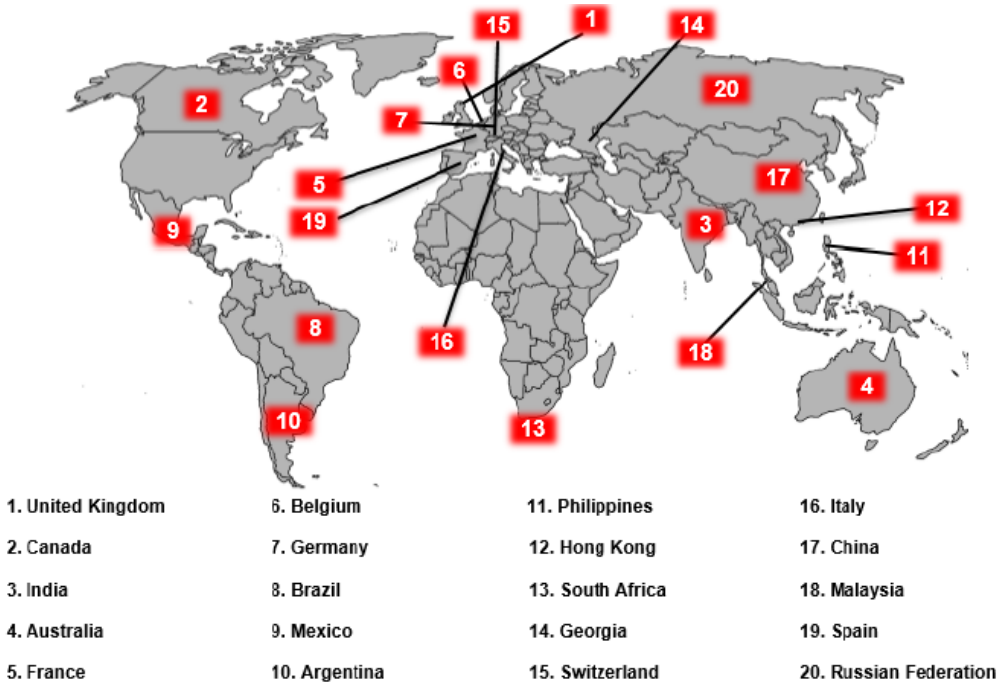


Рисунок 1.4 - 20 найбільших міжнародних постраждалих країн

Кібератаки Популярною метою кібератаки було знищення доступності мережевої інфраструктури. Це означає спробу проникнення в їх особисте життя. Він може завдати шкоди або знищити конфіденційність системи, використовуючи

попередньо визначений план, який протистоїть конфіденційності мережі. Наразі новомодні кібератаки представлені як змішані атаки, які використовують численні атаки для проникнення в бажану систему. Крім того, у звіті про загрози безпеці в Інтернеті припускається, що невеликі компанії більш імовірно будуть зламані за допомогою поширених методів атак, таких як отримання погроз електронною поштою, включаючи спам, зловмисне програмне забезпечення та фішингові електронні листи, які, ймовірно, можуть бути сильнішими, порушуючи невеликі організації, ніж великі. Більше того, за оцінками звіту, спам-атака продовжувала збільшуватися на 55 відсотків у 2018 році порівняно з 2015 роком. Протягом одного дня в середньому було заблоковано 10 573 шкідливих мобільних додатків. Таким чином, акт блокування ділився на 39% для інструментів, 15% для способу життя та 7% для розваг як найпоширеніших категорій, які залишалися шкідливими програмами. Зазвичай і відповідно до кібератаки класифікуються на 2 основні категорії, які є пасивними атаками, які складаються з прослуховування без зміни даних у системі чи мережевому корисному навантаженні. Як правило, їх неможливо виявити, хоча профілактика можлива. Активна атака полягає в тому, що зловмисник намагається обійти захищену систему, щоб проникнути в неї за допомогою шкідливих дій, таких як злом, зловмисне програмне забезпечення та відмова в обслуговуванні.

Визначено та представлено найпоширеніші кібератаки:

1. Фішинг. Фішинг — це вид забороненої діяльності, який робить зловмисників шахраями для законних дій. Наприклад, вони використовують підробку електронної пошти, щоб захопити в пастку шляхом вилучення секретних даних, таких як паролі та дані кредитних карток. Величезна кількість електронних листів підробляється зловмисниками після того, як вони намагаються перенести вразливі посилання, пов’язані з підробленими веб-сайтами. Фішингові атаки використовуються для отримання викраденої інформації, в той час як користувачі вважають цю інформацію справжньою після введення своєї особистої інформації.

2. Шкідливе програмне забезпечення. Шкідливі програми можна віднести до шкідливих інструментів. Він включає багато видів кіберзлочинів, таких як

програми-вимагачі, віруси, шпигунські програми та хробаки. Зловмисне програмне забезпечення має головну мету; він порушує мережеву інфраструктуру після виявлення вразливостей системи. Він може бути представлений, коли жертва натискає невідоме ризиковане посилання через отримане ненадійне вкладення електронної пошти, яке може інсталювати шкідливе та зловмисне програмне забезпечення. Вставивши це зловмисне програмне забезпечення в систему, зловмисне програмне забезпечення може проникнути всередину комп'ютерних мереж і таємно витягти дані, зберігаючи їх на жорсткому диску (шпигунське програмне забезпечення). Крім того, це пошкоджує окремі важливі компоненти та робить його непрактичним.

3. Розсилка спаму електронною поштою – це група незапитаних розсилок повідомлень від незвичайних компаній і асоціацій. Це можна здійснити, розсилаючи величезній масі користувачів мати справу з привабливими акціями та рекламою, що дозволяє ввести відвідувачів в оману.

4. Розподілена відмова в обслуговуванні. Відмова в обслуговуванні може переповнювати мережеві системи або сервери трафіком, щоб виснажити пропускну здатність і вплинути на ресурси. Таким чином, розподілена відмова в обслуговуванні (DDOS) тимчасово відключає мережі або сервери, які успішно працюють. Системи через ці мережі будуть повністю офлайн і потребуватимуть багато часу, щоб відповісти. Зловмисник запускає потік корисного навантаження для жертви, щоб скомпрометувати певні функції, щоб перервати доступність мережі та бути недоступною для користувачів.

5. SQL Injection. Ін'єкція мови структурованих запитів (SQL-ін'єкція) - це атака, під час якої шкідливий код потрібно вставити на веб-сервер. Зазвичай він використовує зловмисний оператор SQL, переданий у вразливому полі введення у веб-додатку. Ін'єкція буде виконана, щоб змусити жертву відкрити витягнуті дані, хоча зазвичай це не відбувається. Недостатня перевірка вхідних даних і невідповідна форма небезпечного оператора SQL у програмі веб-системи можуть наразити їх на такі види атак.

6. Тунелювання DNS. Тунелювання DNS – це атака всередині мережі всередині DNS-запитів і відповідей. Дані протоколу будуть закодовані для надсилання HTTP та інших протоколів через DNS. Переважно DNS-тунелювання включає корисні дані, які можуть бути передані на DNS-сервер таким чином, щоб зловмисник міг отримати доступ до компонентів віддаленого сервера. На практиці для цієї атаки потрібна зовнішня мережа, і її слід підключити до точки зловмисника, щоб отримати доступ до внутрішнього DNS-сервера в мережі жертви. Це допомагає кіберзлочинцям додавати шкідливі атаки всередину DNS-запитів, щоб створити таємний незаконний тунель підключення, який проникає до встановлених інструментів безпеки.

7. Міжсайтова атака (XSS). Міжсайтовий сценарій (XSS) класифікується як атака на стороні клієнта. Він вводить корисне навантаження зі зловмисним кодом JavaScript на веб-сайт, який буде виконано та запущено у браузері жертви. Зокрема, жертва запитає вразливу сторінку, яка містить введений код. Зловмисник передасть згубну сторінку з ін'єктованим корисним навантаженням до цілі, щоб виконати незаконну дію зі зміни тіла веб-сторінки.

8. Обхід шляху. Шлях до рядка вказує на розташування адреси каталогу системних файлів. Таким чином, атака обходу шляху розглядається як процес загрози для доступу до збережених файлів або каталогів, які зберігаються поза основним кореневим веб-сайтом. Змінні можуть маніпулювати посиланнями на вибрані файли за допомогою серії крапка-крапка-скісна риска (../) та її різноманітністю через кінцеві шляхи до файлів. Він може отримати доступ до випадкового компонента та файлів каталогів, зареєстрованих у системі.

1.3 Особливостей застосування методів машинного навчання в кібербезпеці

Машинне навчання (ML) — це підмножина додатків зі штучного інтелекту (AI), яка пропонує системам автоматично навчатися та розвиватися на основі минулих знань без програмування вручну.

Машинне навчання в інтелектуальному аналізі даних зазвичай використовується для реалізації процесу пошуку корисної та прихованої інформації з великої кількості даних. Крім того, це може бути ефективним у розробці комп'ютерних програм, які використовують інформацію, що має вхідні та вихідні дані, щоб створити функцію автоматичного навчання, яка виводить результати для вилучення цінних знань на основі вхідних даних. В основному методи ML часто використовуються в кількох галузях, таких як маркетинг, охорона здоров'я, Інтернет речей, виявлення вторгнень і наукові відкриття [11].

Типи машинного навчання. Як правило, машинне навчання використовує інтелектуальні алгоритми на етапі моделювання, і його зазвичай поділяють на чотири основні типи класів, які включають контрольоване навчання, неконтрольоване навчання, напівконтрольоване навчання та навчання з підкріпленням.

Контрольоване навчання. Контрольоване навчання поділяється на два основні класи; Класифікація та регресія. Класифікація зіставляє дані з деякими попередньо визначеними категоріями. Коли ми перевіряємо нові дані, цей процес забезпечує рівень точності, до яких категорій належать екземпляри даних, і передбачає окрему мітку, як-от човен і автомобіль. З іншого боку, регресія в основному використовується для прогнозування безперервних значень. Більшість додатків використовують переваги регресії для прогнозування та прогнозування, вона допомагає визначити поведінку досліджуваних змінних. Регресія прогнозує постійну кількість або значення, наприклад час або вагу.

Навчання без контролю. У неконтрольованому навчанні вхідні дані не позначаються вихідними, тому немає жодних раніше помічених шаблонів, а також надається з мінімальним контролем для людей. Як наслідок, мета тут — виявити шаблони для доступних даних. Нижче наведено різні методи навчання без контролю:

1. Асоціація. Ця техніка здебільшого шукає найпоширеніший набір елементів із великого обсягу даних. Він використовується для визначення зв'язку між змінними ознак і, таким чином, вказує на найбільш ефективні змінні цього зв'язку

як результат майбутніх значень. У величезних базах даних і за допомогою правил асоціації ми можемо створювати асоціації між об’єктами даних у великих базах даних, щоб помітити цінні зв’язки між змінними. Приклад правил асоціації в аналізі ринку, коли клієнти часто купують товари разом.

2. Кластеризація. За допомогою цієї техніки дані поділяються на групи, які не визначені з невідомими властивостями, які розпізнаються як кластери. Кластер — це група екземплярів, яка описує дані, які він містить і розкриває природну структуру досліджуваних даних. Можна сказати, що екземпляри даних в одному кластері більш схожі один на одного, ніж дані, що містяться в будь-якому іншому кластері. Деякі елементи не належать до жодного кластера, відомі як викиди. Ми можемо спостерігати структуру викиду, порівнюючи їх з даними, знайденими в інших кластерах, і це виявляє нестандартну поведінку системи [69]. Крім того, це включає навчання без нагляду. Основна мета представлена у виявленні відповідного шаблону з групою немаркованих даних. Обробка даних відбувається в алгоритмах для кластеризації, а потім визначається, чи присутні кластери в даних. Номер кластера, який потрібно визначити, можна змінити в алгоритмах, щоб маніпулювати точністю цих кластерів. Методи кластеризації можна узагальнити за цими областями; сегментація ринку, групування результатів пошуку, аналіз соціальних мереж, медична візуалізація та сегментація зображень.

Напівконтрольоване навчання. Як мічені, так і не мічені вхідні дані тренуються під час напівконтрольованого навчання, але кількість не мічених даних є набагато більшою, ніж мічених даних під час навчання. Ця комбінація покращує навчання та точність. Таким чином, напівконтрольоване навчання є сумішшю контрольованого та неконтрольованого навчання разом.

Етапи машинного навчання. Механізм машинного навчання розділений на сім етапів, описаних на рис.1.5.

Рисунок 1.5 - Етапи машинного навчання

Вся перша частина процесу машинного навчання полягає в зборі даних. Для цього потрібні три тактики: виявлення даних, збільшення даних і генерація даних. Щоразу, коли нам потрібно досліджувати нові набори даних, відкриття даних є важливим, оскільки доступні загальнодоступні набори даних через мережу Інтернет або дослідницькі установи. Після виявлення даних відбувається розширення даних, оскільки поточні бази даних зміцнюються шляхом додавання додаткових зовнішніх даних. Якщо існуючий набір даних насправді неадекватний і зовнішні дані також недоступні, генерація даних може бути корисною, навіть якщо існує можливість створювати набори даних альтернативно. Набір даних має бути попередньо оброблений після отримання, щоб мати можливість передати все в модель. Метод перетворення необроблених даних у типи даних складається з введення моделей — підготовки даних або попередньої обробки даних. У всіх техніках машинного навчання завжди є потреба у вхідних даних. Але в більшості методів ML ці вхідні дані слід спочатку переформатувати перед їх використанням. Деякі набори даних містять дані, які є неповними, ненадійними або складними для роботи алгоритмів. Потім, на третьому кроці, необхідно виконати вибір відповідної моделі, щоб точно вирішити реальну проблему з можливістю масштабування.

Подальша складна модель не гарантує, що модель краща. Метою вибору моделі є навчання набору даних. Під час навчання набору даних ми також навчаємо модель, щоб поступово вдосконалювати прогнози моделі. Упередження та ваги продовжують змінюватися в кожній ітерації. Модель навчається з позначеними даними в керованому машинному навчанні; однак неконтрольоване машинне навчання має вивчати дані, які не позначені. Фаза оцінки моделі починається після навчання; на цьому етапі тестування алгоритмів ML відбувається з використанням

частини даних, яка раніше не використовувалася на попередньому кроці навчання. Це відображає успіх моделі в реальній ситуації. Потім модель можна вдосконалити шляхом тестування параметрів. Збільшуючи кількість ітерацій навчання, можна отримати більш надійні результати. Потрібна подальша настройка, доки ви не забезпечите найефективнішу продуктивність моделі.

Алгоритми машинного навчання. Далі мною перераховано типові алгоритми машинного навчання, які можна використовувати майже для будь-якої проблеми класифікації.

Дерево рішень. Дерево рішень класифікується як навчання під наглядом. Його можна застосовувати через два типи типів машинного навчання: класифікацію та регресію, що дозволяє приймати рішення в наукових та пов'язаних із ними проблемах. Крім того, дерево рішень використовує алгоритм для прогнозування відповідної вихідної мітки на основі навченого джерела даних. Його потрібно перевірити за допомогою вектора ознак, щоб змоделювати його в правилах прийняття рішень, які слід вивчати з їхнього власного набору даних. Цей алгоритм починається як дерево, починаючи з вершини через кореневий вузол. Значення атрибута root необхідно порівнювати з атрибутом record.

Внутрішні вузли символізують особливості підмножини, а гілки представляють визначені правила, тоді як кожен листовий вузол представлятиме прогнозовані результати. Дерево рішень використовує кілька алгоритмів, таких як: ID3, C4.5, CART, CHAID і MARS [12].

Існує кілька мір домішок, ми наведемо найважливішу з двох вимірювань; ентропія та індекс Джині/домішка Джіні.

Ентропія - це набір інформації, необхідної для точного окреслення деяких випадків. Наприклад, якщо є однорідний екземпляр і елемент ідентичний, тоді ентропія дорівнює 0. В іншому випадку, якщо він розділений однаково, ентропія має бути максимум 1. Математично ентропію можна обчислити за такою формулою (1.1).

$$Entripy = -\sum_{i=1}^{n} p_i * \log(p_i) \quad (1.1)$$

Індекс Джині відноситься до вимірювання нерівності у прикладі зі значенням від 0 до 1. Якщо індекс Gini дорівнює 0, це означає, що екземпляр є однорідним, тоді як, якщо індекс Gini дорівнює 1, це стосується нерівності між елементами. математично це підсумовування квадратів ймовірностей для кожного класу, і його можна обчислити за такою формулою (1.2)

$$Gini\ index = 1 - \sum_{i=1}^{n} p_i^2 \quad (1.2)$$

Де і – кількість класів

Метод K-найближчі сусіди. Метод K-найближчих сусідів зберігає доступні вхідні дані відповідно до вимірювання подібності, він класифікує нові записи, що використовуються для регресії та ускладнень класифікації.

Інша назва методу K-найближчих сусідів — ледачий учень, оскільки фаза навчання не відбувається безпосередньо з навчального набору, однак він зберігає навчені дані та класифікує нові вхідні дані відповідно до критеріїв відповідності.

Метод машини опорного вектора (SVM). Машина опорних векторів (відома як SVM) — це контрольований метод навчання, розроблений для двійкової класифікації «позитивного або негативного класу». Він використовується з метою пошуку закономірностей із колекції даних. Загалом, класифікація шаблонів застосовує діяльність, яка має бути залучена до двох основних кроків; перший – відображення вхідних даних у простір об’єктів більшої розмірності, це робиться завдяки тому, що SVM зазвичай залежить від геометричних характеристик введених даних, а другий – це пошук найбільш зручної гіперплощини, яка класифікує відображені об’єкти в вищий вимірний простір [11]. Це корисно для підходів класифікації та регресії. Поле — це простір між гіперплощиною та сусідньою точкою даних. Основною метою є виявлення гіперплощини з найбільшим розколом бази даних; на 2 класи для отримання нових векторів із належною класифікацією.

Важливими виразами в опорній векторній машині є:

- опорні вектори: це точки даних, які є найближчими до гіперплощини. Точки даних визначають лінію розділення;

- гіперплощина: це лінія, що розділяє набори об’єктів на різні класи;

- поле: дірка між 2 лініями з найближчими точками даних для різних класів. Відступ — це простір, який відокремлює лінію від опорних векторів. Чим вище націнка, тим зручніше.

Крім того, ці роздільники гіперплощини (межі рішення) визначаються формулою:

$$h(x) = x^T\beta + \beta_o = 0 \quad (1.3)$$

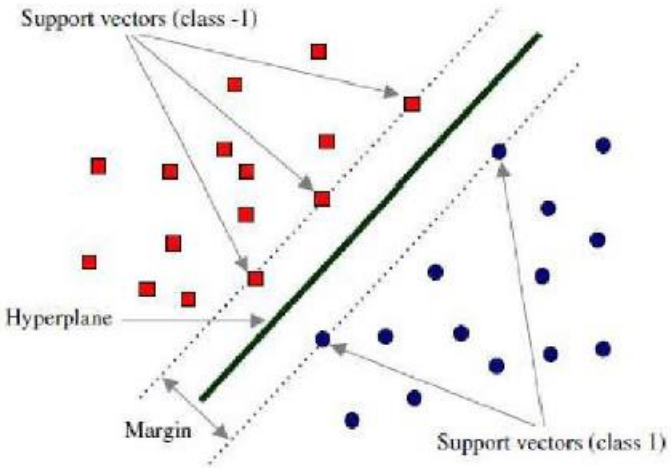
Де: x - вектор ознак; β - вектор коефіцієнтів; β_0: постійне значення.

Для будь-якої точки простору відстань від точки до гіперплощини визначається формулою (1.4)

$$d(x) = \frac{|x^T\beta+\beta_o|}{||\beta||} \quad (1.4)$$

Знаючи, що $||\beta|| = \sqrt{\beta_1^2 + \cdots + \beta_p^2}$і максимізація запасу означає мінімізацію норми вектора параметрів β.

Обрана гіперплощина є такою, яка максимізує запас, представлений відстанню між точками навчання, найближчими до гіперплощини розділювача: опорні вектори, показані на рис.1.6.



p(

Рисунок 1.6 - Техніка опорного вектора

Лінійні ядра: $k(xi\ ,\ xj\ ) = xi\ .\ xj$

Поліноміальні ядра: $k(xi\ ,\ xj\ ) = (1 + xi\ .\ xj\ )\ p$

Ядра Гауса: $k(xi\ ,\ xj\ ) = e{-}\ \|xi{-}xj\|\ 2\ \sigma 2$

Сигмоїдні ядра: $k(xi\ ,\ xj\ )$ = tanh $(\alpha xi\ .\ xj + \beta)$

Техніка ансамблю. Ансамблеві методи використовують кілька алгоритмів навчання, поєднуючи їх для отримання єдиного потужного прогнозу. Дійсно, застосування інтелектуальної моделі з використанням методів ансамблю досягає та дає точніші результати, ніж одна модель, яка була отримана раніше. Використання методів ансамблю може бути використано для кількох цілей класифікації, які виконують модель шляхом зменшення дисперсії, а також зменшення зміщення, щоб покращити прогнозування бажаного результату.

Випадковий ліс. Випадковий ліс — це керовані ансамблеві моделі навчання, які використовуються для класифікації та регресії. Випадковий ліс складається з кількох дерев прийняття рішень для класифікації нового об'єкта з вхідного вектора, тоді як вхідним вектором андомного лісу є вхідні дані кожного дерева. Вчений доводить, що цей метод дозволяє досягти точного та стабільного прогнозу з високою продуктивністю. Крім того, всупереч дереву рішень, який показаний на рис.1.7 використовує процеси пошуку кореневого вузла та розбиття вузлів ознак буде виконуватися випадковим чином.

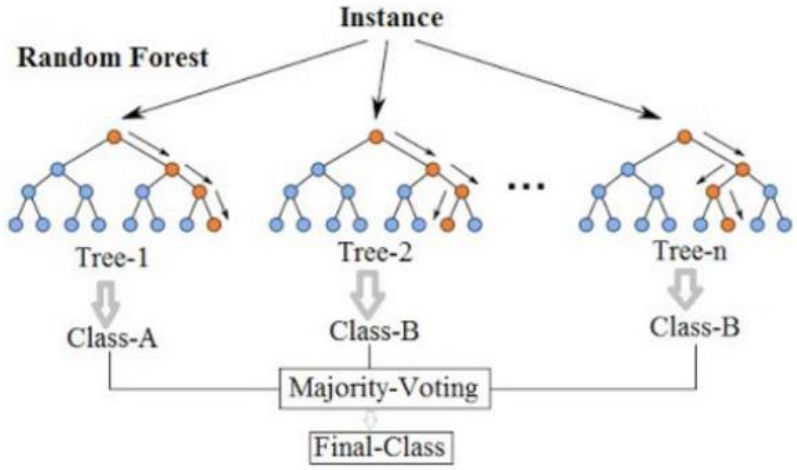


Рисунок 1.7 - Техніка випадкового лісу

На етапі прогнозування RF використовує перевірку кількох функцій, щоб використовувати основи кожного випадково згенерованого дерева для

прогнозування та збереження виходу шляхом вибору правил класифікації, які отримують максимальну кількість голосів [12].

Підсилення. Він розглядається як ансамблевий підхід, який використовується для методу машинного навчання, щоб зробити прогнозування ослабленої моделі більш ефективним шляхом послідовного навчання кожного слабкого учня, тому кожен учень коригує свого предка. Цей метод виконує кілька кроків показаних на рис.1.8.

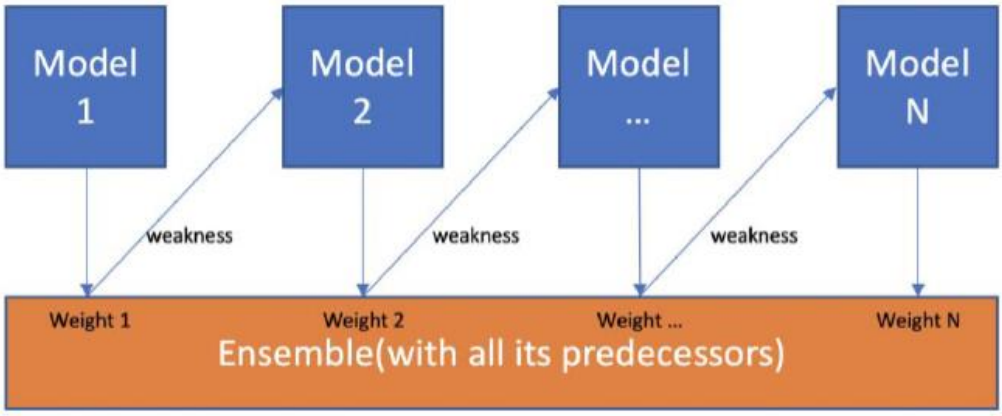


Рисунок 1.8 - Загальна архітектура посилюючого класифікатора

Існує багато алгоритмів посилення, таких як AdaBoost, GBM, XGBM, Light GBM, CatBoost і Extreme Gradient Boosting. ми перерахуємо деякі з них, як зазначено нижче:

1. ADABoost. Це ітераційна ансамблева техніка для побудови ефективного класифікатора шляхом злиття кількох недостатньо сильних класифікаторів, таких як створена потужна модель, яка може мати ефективні результати. Упродовж описаних кроків він виконує наступне:

- навчання починається з випадкової підмножини набору даних;

- вибираючи найкращий надійний набір даних навчання з попереднього прогнозу навчання, він навчається ітеративним способом;

- експериментам із помилковими прогнозами надається більша вага, але ці експерименти матимуть великі шанси на класифікацію в наступній ітерації;

- розподіл ваги виконується з точністю кожного класифікатора. Коли класифікатор буде більш точним, він набуде більшої ваги.

2. Екстремальне посилення градієнта (XGBoost). Екстремальне підсилення градієнта (XGBoost) також є ансамблевим підходом, створеним за допомогою методів дерева рішень, який покращує посилення градієнта, це корисно для регресії чи класифікації. Підвищення градієнта, представлене у формулі, наведеній нижче, послідовно об’єднує предиктори та виправляє минулі ненадійні оцінки. Виходячи з результатів попередніх прогнозів, цей підхід узгоджується з новою моделлю та зменшує втрати шляхом додавання останнього прогнозу. Щоб оптимізувати продуктивність узагальнення моделі, XGBoost використовує автоматичну регулярізацію. Він забезпечує кращу ефективність у порівнянні з Gradient Boosting, крім того, це значно регуляризована форма Gradient Boosting. Навчання є дуже швидким і може бути розподіленим/паралельним через кластери.

3. Голосування. Цю техніку можна використовувати для вирішення питань класифікації. Підхід до голосування, який використовується як ансамбль, представлено як приклад на рис.1.9. Щоб спрогнозувати кожен вхідний запис, часто використовується кілька моделей. Прогноз для кожної моделі розглядається як голосування. Останні прогнози, здається, є вихідною міткою, яка отримує голоси меншості [13].

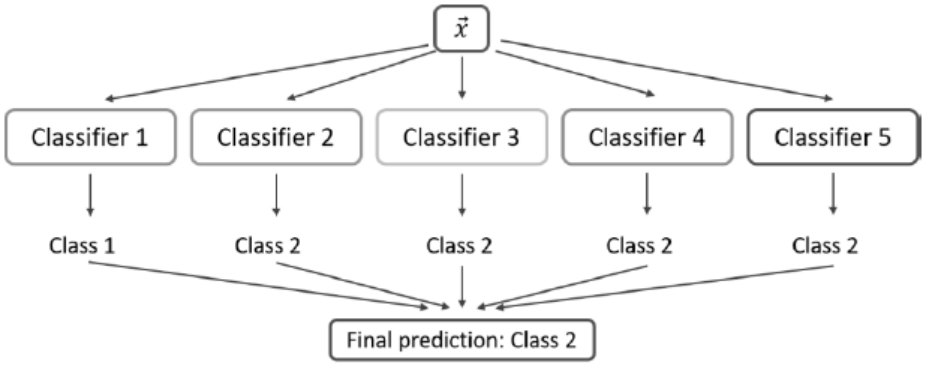


Рисунок 1.9 - Загальна архітектура класифікатора голосування

4. Укладання в мішки (початкове агрегування). Для того, щоб отримати стандартизований результат, розфасовка включає в себе результати багатьох

моделей. На основі підмножин, включених до основного набору даних, генерується та навчається кілька базових моделей. Як ви можете бачити на рис.1.10, пакетування бере N вибірок у повторенні з екземплярів навчання з величиною N, щоб навчити головний класифікатор, і він продовжує повторюватися, доки не буде досягнуто бажаного розміру ансамблю.

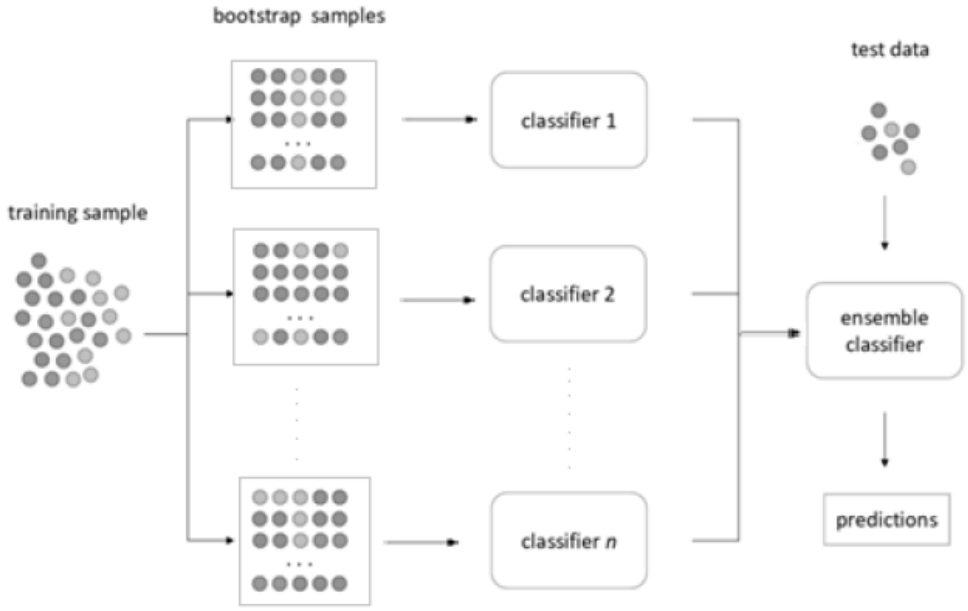


Рисунок 1.10 - Загальна архітектура класифікатора упаковки

Головним чином, метод пакетування повинен використовуватися через нестабільні класифікатори, які мають вирішальне значення для варіацій на етапі навчання, як-от дерева рішень та інші. Крім того, алгоритм Bagging можна використовувати за допомогою оцінювачів буксування, метаоцінки bagging або випадкового лісу. Щоб вибрати найкращий етап прогнозування, моделі виконуються разом і незалежно, а потім об’єднуються результати всіх моделей.

5. Укладання. Укладання розглядається як метод ансамблю, який створює нову модель, використовуючи прогнози з різних моделей, як показано на рис.1.11.

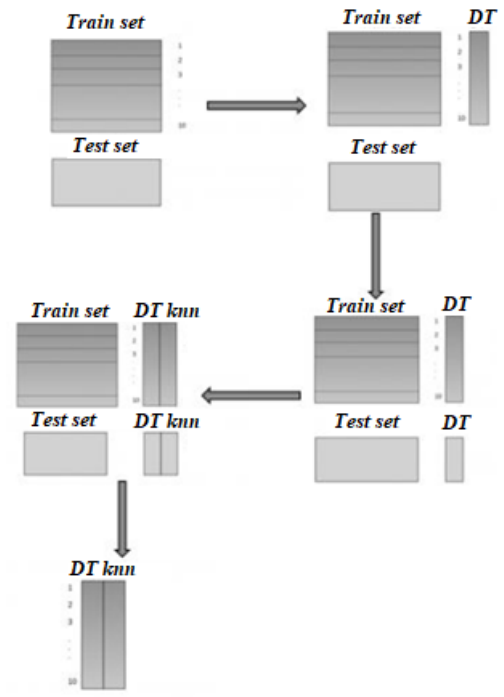


Рисунок 1.11 - Загальна архітектура класифікатора стекування

1.4 Система виявлення вторгнень на основі методів машинного навчання

Фактично, кілька IDS страждають від слабких місць через велику кількість помилкових тривог, які призводять до зростання проблем, з якими стикаються кібераналітики, з шкідливими наслідками. Це спричиняє знищення у разі непоміченої загрози. Розробка IDS привертає значну увагу кібердослідників, щоб подолати проблему збільшення рівня виявлення для мінімізації помилкових і з точки зору нездатності розрізнити незнайомі атаки, оскільки шаблони мережі швидко змінюються, і з'являється кілька видів атак. безперервно. Попереднє виявлення вторгнень виконується аналітиками-людьми (аналітиками з безпеки), які приймають рішення вручну, і розглядати величезну кількість атак вважається складною роботою. Крім того, стандартна система виявлення вторгнень потребує втручання людини шляхом створення даних підпису, що робить її далеко не інтелектуальною. Таким чином, вдосконалена IDS, що підтримується машинним навчанням, надзвичайно потрібна для забезпечення альтернативного процесу, пов'язаного з втручанням людини. Таким чином, для лікування цього підходу вчені

зосередили розробку IDS на основі методів машинного навчання для досягнення динамічної інтелектуальної форми з точним виявленням. Крім того, використання IDS з використанням машинного навчання, як показано на рис.1.12, забезпечує прогнозування атак, побудоване з надійних і попередньо оброблених джерел даних, навчених на моделі для досягнення результатів рішення. Він може бути ефективним порівняно з традиційним у спостереженні за шкідливими діями, які відбуваються в мережах і системах, у режимі реального часу з гідною швидкістю виявлення.

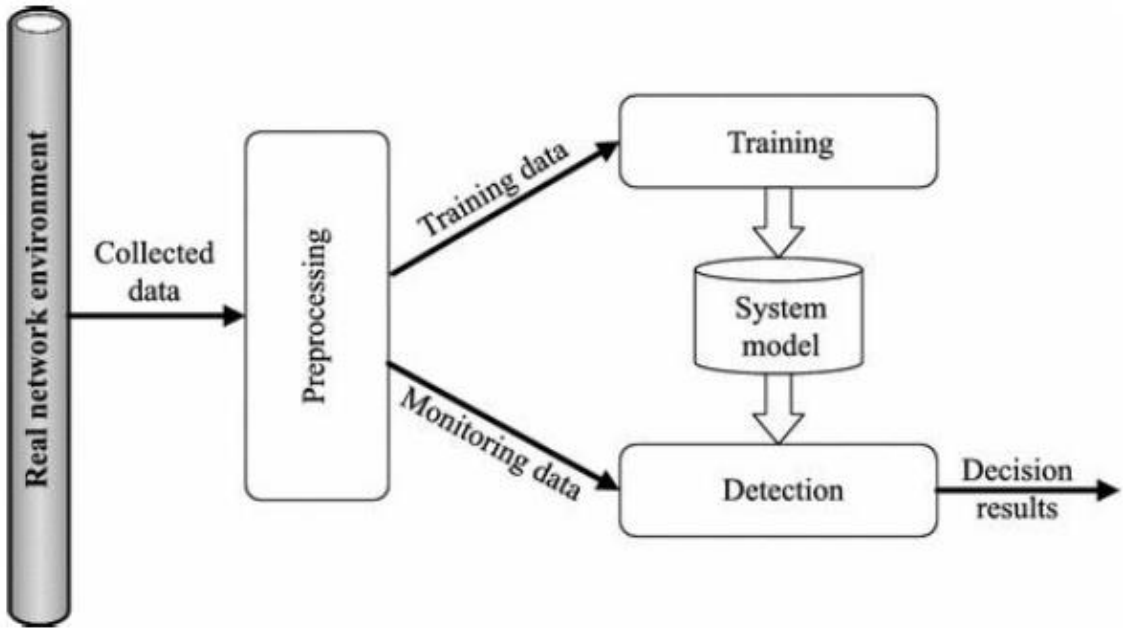


Рисунок 1.12 - Виявлення вторгнень на основі методів машинного навчання

Крім того, ми представимо огляд кількох методів машинного навчання, які були реалізовані дослідниками в їхніх моделях виявлення вторгнень. Далі ми покажемо різні підходи, які використовували надійні набори даних, як показано в табл.1.3.

Бузіда та ін. [14] використовували дані KDD, які були витягнуті з набору даних DARPA. Через відсутність нових атак у цьому наборі даних вони запропонували методологію додавання реального трафіку зі своєї лабораторії.

Тому DT і нейронна мережа використовувалися як методи навчання під наглядом для виявлення вторгнення на основі методу виявлення аномалій.

Нейронна мережа була використана для підвищення рівня точності, але вона досягла низької продуктивності в процесі виявлення нової атаки. Дерево рішень показало кращі результати як щодо точності, так і виявлення нових атак. Таким чином, дослідники досягли цікавих результатів, порівнюючи ефективність застосованих моделей зі старими підходами.

Нарешті, внесок дослідників довів застосовність системи виявлення аномалій як для нормальної, так і для відомої атаки після додавання нових записів про атаки на етапі навчання.

McElwee та ін. [15] запропонував техніку фільтрації сповіщень за допомогою DNN. По-перше, вони збирають журнал, створений McAfee. Після цього виконується навчання моделі DNN для виявлення основних шаблонів безпеки в журналах. Потім виявлені важливі дії, перевірені фахівцями з безпеки, і результати перевірки використовуються для тренувальних даних для вдосконалення моделі DNN для встановлення взаємодії та циклу просування та полегшення процесу виявлення.

Наджафабаді та ін. [12] розробили моделі прогнозу, щоб відрізнити регулярний мережевий трафік від аномального мережевого трафіку за допомогою методів машинного навчання. Запропонована архітектура інтегрує підхід до скорочення даних (наприклад, вибір функцій), щоб усунути непотрібні та дубльовані функції, що призведе до скорочення часу обробки. Вони використовували 4 різні підходи до вибору ознак для вивчення 3 моделей класифікації (Нав Байєса, 5-найближчий сусід і C4.5 DT. Моделі навчені та перевірені з використанням набору даних Kyoto2006+. Результати продемонстрували, що коли вибір функції зменшує ознаки, він зберігає подібні результати або не сильно зменшує результати. Вони визначили, що в області застосування IDS етап вибору функції є критичним етапом попередньої обробки, який не можна ігнорувати.

Таблиця 1.3 - Пов’язані роботи, які використовували надійний набір даних у розробці IDS

Дослід-ник	Тип IDS	Фаза попередньої обробки	Nb. Заняття	Використа-ний набір даних	Найкраща модель
Sharafaldin I et al	IDS для DDOS	Вилучення функцій	13	CICDDoS 2019	Випадковий ліс - Точність: 0,77 - Згадайте 0,65 - Оцінка F1 0,69
Thabtah et al.	Модель захисту від фішингу	Вибір функції	2	Фішингові веб-сайти	Епоха нейронної мережі (500) - Acc = 93,06% - Оцінка F1 = 92,30% - Відкликання = 91,12% - Точність = 93,71%
Elsayed MS et al.	Виявлення мережевих атак	-Особливості зменшення - Очищення даних - Нормалізація даних	2	CICDDoS 2019	Глибоке навчання - Точність: 0,99
Mamun MSI et al.	Виявлення шкідливих URL-адрес	Вибір функції	5	ISCX-URL-2016	Випадковий ліс - Точність: 0,97 - Відкликання: 0,97
Kapil Det al.	Виявлення шкідливих URL-адрес	Вибір і скорочення функцій	5	ISCX-URL-2016	Weka: Випадковий ліс - TPR:0,961;FPR:0,032 - Точність: 0,961 - Згадайте 0,961 - Оцінка F1 0,961

Систему виявлення вторгнень на основі машинного навчання можна проілюструвати наступними важливими кроками; отримання даних за допомогою різних інструментів, таких як (перегляд, реєстрація, датчики апаратних засобів), інженерія даних і функцій (наприклад, регулярізація, очищення та нормалізація даних), фаза моделювання (наприклад, класифікація, кластеризація) і показники продуктивності:

1. Збір даних. Надання інформативного набору даних (вхідних даних) зазвичай використовується з метою аналізу. Вхідні дані — це набір екземплярів, таких як вектори зразків, відповідні шаблони та необроблені дані, які можуть продемонструвати, чи є вибраний шаблон нормальним чи ні. Ця фаза в системі

виявлення вторгнень складається зі збору даних із систем, які піддаються атакам. Останній інтерес дослідницьких центрів зосереджується на ідеї знання відповідних функцій для вибору джерела даних, яке буде взято як еталонний для інтелектуального алгоритму. Існує обмеження дослідницьких центрів у забезпеченні надійного джерела даних, яке має велику кількість і якість даних. Обмеження пов'язане з принципом конфіденційності, прийнятим компаніями, який вважається одним із головних пріоритетів збереження конфіденційності даних.

У підрозділі описано набір даних, який найчастіше використовується для реалізації інтелектуальної IDS.

Існує два джерела для отримання даних: веб-сервер і дані на основі мереж. Кожен з них має свої особливості та має свої основні переваги та недоліки:

- дані веб-сервера.Він складається із записів, зібраних з даних журналу, які містять дії користувача, які підтримуються та створюються динамічно (журнали веб-сервера, база даних, журнали брандмауера). Ці файли журналів можна розглядати як джерела даних, тому всі дії користувачів реєструються утилітою журналу. Цей тип даних висвітлює кілька індикаторів серйозних проблем, що виникають у системі. Однак файли журналів містять тисячі й мільйони дій, представлених у формі двійкових форматів, простих текстів або комбінації обох. Журнал веб-сервера оцінюється в багатьох дослідженнях як кінцевий пристрій запиту HTTP, наприклад IIS для платформи Microsoft і apache для Linux. Він реєструє дії відвідувача, які відбуваються на веб-сервері, на основі специфікацій загального формату журналу (CLF). Крім того, файл журналу містить цінну інформацію, включаючи поведінку відвідувача щодо ідентифікатора користувача, запитуваної URL-адреси, URL-адреси переходу, IP-адреси, коду стану та багатьох параметрів, включаючи звернення, які записані як параметр GET у цьому файлі. Він розглядається як набір вхідних даних, який є важливою частиною виявлення шкідливих дій, що проходять через веб-сервер. Таким чином, багато вчених із кібербезпеки використовують це корисне джерело даних для виявлення хакерських і аномальних дій;

- дані мереж. Мережні дані складаються з джерел, які вже розроблені центрами кібербезпеки, зібрані з мережевого трафіку, в якому реєструються потенційні атаки. Таким чином, мережевий трафік буде вибрано для експорту як набору даних. Нижче ми наведемо найбільш використовувані джерела даних у розробці IDS, як показано в таблиці [6,11].

2. Розробка даних і функцій. Згідно з доступною літературою, пов’язаною з побудовою інтелектуальної IDS [11], дані та інженерія функцій та їх представлення відіграють важливу роль у процесі виявлення та стають найважливішими факторами, які впливають на адекватність IDS. Цей процес залежить від очищення та сегментації даних, а також від зменшення функцій, які не впливають на процес класифікації IDS, щоб покращити IDS зі збільшенням як швидкості обчислення pf, так і точності виявлення. Існує кілька методів функцій, які допомагають витягувати та вибирати корисні предиктори із зібраних даних, наприклад латентне семантичне індексування (LSI), що використовується для отримання знань із неструктурованих даних, і аналіз головних компонентів (PCA) як техніка зменшення ознак для зменшення кількості вибрану функцію шляхом розробки нових попередньо оброблених даних, які містять остаточні дані з невеликою кількістю ефективних предикторів. Крім того, метод вибору ознак може бути використаний для покращення ефективності результатів, пропонуючи рекурсивне усунення ознак на основі ранжирування функцій, використовуючи метод перехресної перевірки «feature_selection.RFECV», який дозволяє отримати найкращі накопичені функції, які будуть використовуватися на етапі навчання.

3. Фаза моделювання. Процес моделювання класифікується за кількома методами, щоб запропонувати класифікатор аномалій, який слід використовувати в процесах навчання та виявлення. Він може використовувати методи машинного навчання, такі як контрольоване, неконтрольоване та напівконтрольоване навчання. Крім того, слід перевірити вимірювання наближення, яке повертає числові показники для оцінки щодо остаточного рішення щодо виявлення, зазначеного в наступній частині.

4. Показники продуктивності IDS. IDS на основі машинного навчання оцінюється за допомогою показників продуктивності для оцінки ефективності виявлення. Таким чином, кібердослідники застосовують різні вимірювання для оцінки ефективності моделі. Деякі з цих вимірювань перераховані нижче:

- коефіцієнт виявлення атак (ADR): співвідношення загальної кількості атак, виявлених системою, до загальної кількості атак у наборі даних;

- рівень виявлення атак = Загальна кількість виявлених атак/загальна кількість атак * 100

- частота помилкових тривог (FAR): співвідношення між неправильно класифікованою загальною кількістю випадків до загальної кількості звичайних випадків.

- частота помилкових тривог = Загальна кількість неправильно класифікованих випадків/загальна кількість нормальних випадків * 100

Таблиця 1.4 пояснює кожну фазу частоти тривоги; нижня права клітинка вказує на кількість з’єднань, класифікованих як атака, якщо вони дійсно є атакою (TN), а верхня ліва клітинка вказує на кількість з’єднань, класифікованих як звичайні, і вони дійсно були нормальними (TP). Інші комірки позначені кількістю неправильно класифікованих з'єднань. Верхня права комірка вказує на кількість з’єднань, класифікованих як атака, але вони справді нормальні (FP), тоді як нижня ліва вказує на кількість з’єднань, класифікованих як нормальні, але вони дійсно були атакованими (FN) [11].

Таблиця 1.4 - Частота тривоги виявлення для вимірювання IDS на основі методів машинного навчання

	Класифікується як нормальний	Класифікується як атака
Нормальний	TP	FP
Атака	FN	TN

РОЗДІЛ 2 ДОСЛІДЖЕННЯ СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ ХОСТА АНАЛІЗУ ТЕКСТУ ТА МАШИННОГО НАВЧАННЯ

У даному розділі було виконано оцінку продуктивності досліджуваного HIDS (Host-based Intrusion Detection System) за допомогою чотирьох методів машинного навчання. Досліджено систему виявлення вторгнень на хост за допомогою методу інтелектуального аналізу тексту.

Методи захисту мереж, систем і конфіденційної інформації відомі як служби кібербезпеки, які можуть бути побудовані на протоколі HTTP/HTTPS. Дослідники показують, що серед різноманітних веб-атак ін’єкція коду на веб-сторінках зростає з кожним роком і проводить до 96,15% веб-атак за останні кілька років [16]. Крім того, згідно з WAAR, який зосереджується на найпоширеніших типах ін’єкційних атак [6], атака міжсайтового сценарію (XSS) налічує 49,09% усіх веб-атак, атака SQLI налічує 28,32% усіх веб-атак і нарешті, Path Traversal нараховує 9,82% усіх веб-атак (DT) [7]. Крім того, введені користувачем дані діяли як метод ін’єкції для атак веб-додатків. Таким чином, ці вхідні дані з’являються в рядку запиту запиту HTTP GET. Залежно від цього припущення, шкідливий запит можна розглядати як одну з основних атак веб-ін’єкцій [8].

Тому кібернауковці представляють різні методи для виявлення шкідливих дій. Таким чином, ці дії виявляються як запити у веб-запитах з використанням систем виявлення сигнатур і аномалій [9].

Набір даних вибрано таким чином, щоб він відповідав файлу журналу веб-сервера. Він містить 6000 текстових записів SQLi, XSS і проходження шляху, пов’язаних із рядком запиту URL-адрес HTTP GET. Ми також обговоримо труднощі, з якими можуть зіткнутися ці джерела даних, наприклад, страждання від методу складних функцій. Ми застосували чотири моделі за допомогою техніки машинного навчання; дерево рішень, багатошаровий перцептрон (MLP), опорна векторна машина та KNeighbors. Нарешті, ми перевіримо застосовані моделі, ці методи запропонують класифікацію для виявлення атак SQL-ін’єкцій, XSS і

проходження шляху, а також експортуватимуть вимірювання продуктивності для кожної моделі.

2.1 Аналіз архітектури Host-based Intrusion Detection System

В даному підрозділі виконано аналіз процесу реалізації системи виявлення вторгнень на основі хосту. Дані вибрані таким чином, щоб вони відповідали файлу журналу, отриманому з веб-сервера. Крім того, виконано техніку попередньої обробки із досліджуваним методом представлення ознак.

Кожну необроблену частину тексту буде перетворено на вектор ознак шляхом додавання всіх унікальних слів до словника.

Таким чином, кожне слово у словнику стає ознакою у векторі для представлення вхідного тексту. Тому досліджено систему IDS, щоб реалізувати атаки як процес виявлення за допомогою кількох методів машинного навчання, показаних на рис.2.1.

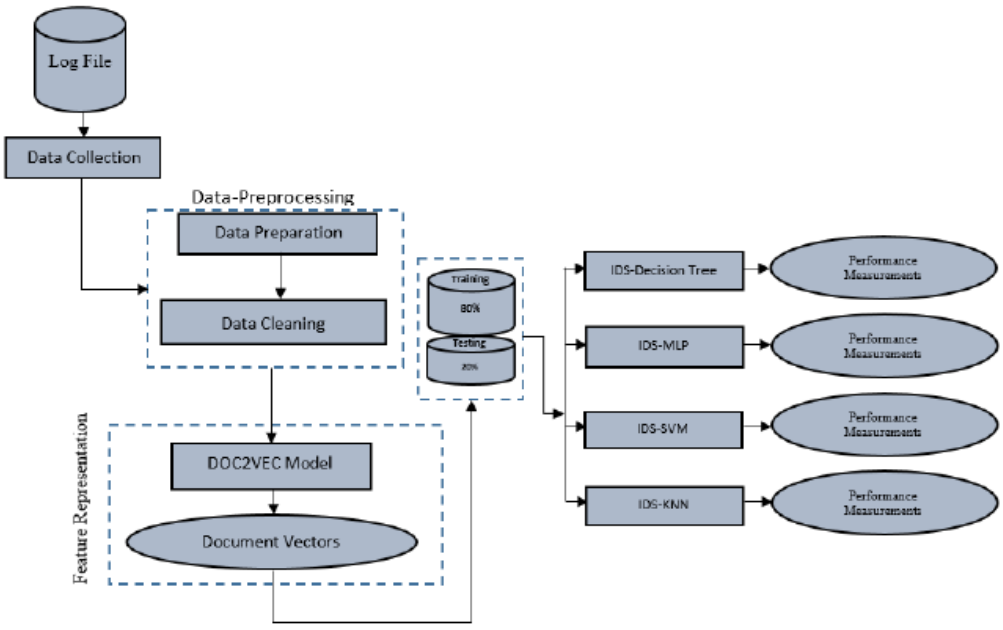
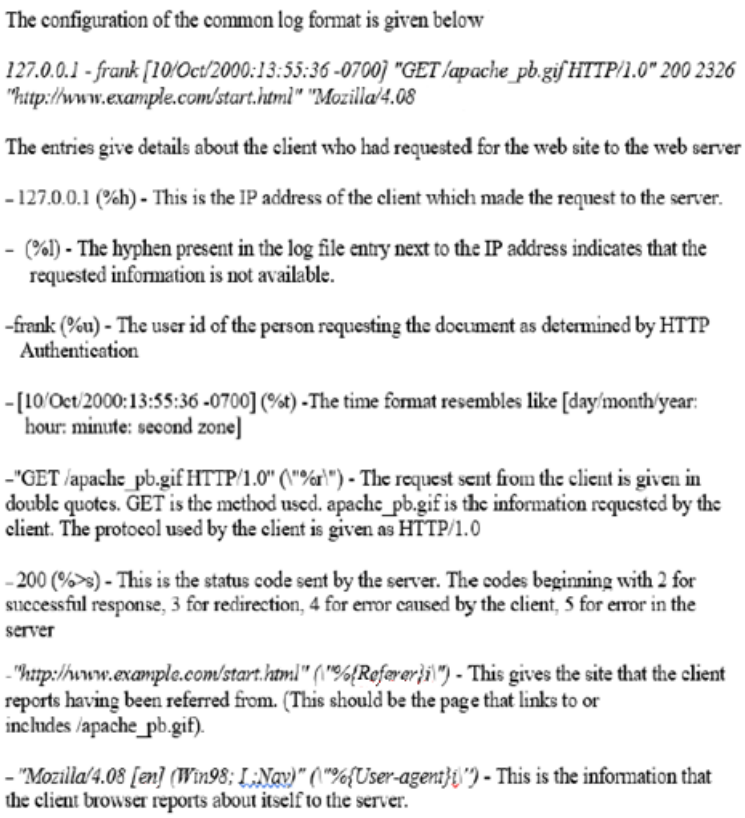


Рисунок 2.1 - Запропонована архітектура HIDS із фазою впровадження

Збір даних. Файли журналів розширюються майже всіма реальними веб-серверами. Таким чином, наявність цих файлів журналу дає нам велику перевагу

для нашої інтерпретації. Таким чином, у нашому дослідженні використано файл журналу apache, представлений у форматі CLF, як показано на рис.2.2.



The configuration of the common log format is given below

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08

The entries give details about the client who had requested for the web site to the web server

- 127.0.0.1 (%h) - This is the IP address of the client which made the request to the server.

- (%l) - The hyphen present in the log file entry next to the IP address indicates that the requested information is not available.

-frank (%u) - The user id of the person requesting the document as determined by HTTP Authentication

- [10/Oct/2000:13:55:36 -0700] (%t) -The time format resembles like [day/month/year: hour: minute: second zone]

-"GET /apache_pb.gif HTTP/1.0" (\"%r\") - The request sent from the client is given in double quotes. GET is the method used. apache_pb.gif is the information requested by the client. The protocol used by the client is given as HTTP/1.0

- 200 (%>s) - This is the status code sent by the server. The codes beginning with 2 for successful response, 3 for redirection, 4 for error caused by the client, 5 for error in the server

-*"http://www.example.com/start.html" (\"%{Referer}i\")* - This gives the site that the client reports having been referred from. (This should be the page that links to or includes /apache_pb.gif).

- *"Mozilla/4.08 [en] (Win98; I ;Nav)" (\"%{User-agent}i\")* - This is the information that the client browser reports about itself to the server.

Рисунок 2.2 - Файл журналу у форматі CLF

Це джерело даних пропонує інформацію, пов’язану з безпекою. Кожен рядок журналу є комбінацією статичної та динамічної інформації, яка складається з інформації про дату та час, інформації про користувача, інформації про події та інформації про програму. Використані файли журналів були зібрані з уразливих сайтів, які включають кілька ресурсів і складаються з динамічних і статичних веб-сторінок. Ці сайти були розміщені на веб-платформі Server-Linux.

Тому протягом певного часу здійснювалась велика кількість різноманітних атак на веб-сервер за допомогою веб-системи за допомогою методів ручного та динамічного впровадження. Спочатку виконано деякі завдання, просканувавши вразливості за допомогою інструменту OpenVas. Крім того, виконано потенційні атаки за допомогою досліджених сценаріїв оболонки, використовуючи інструмент тестування на проникнення для веб-додатків і онлайн-звіти про вразливості, такі як

US-CERT і CERT/CC Advisories. Крім того, виявлено цінну інформацію з кількох списків розсилки, пов’язаних із безпекою.

Попередня обробка даних. Навігація користувача на веб-сайті представлена у вигляді рядків у файлах журналу веб-сервера, деякі з цих рядків не розкривають жодної корисної інформації. Вони непотрібні для процесу аналізу та можуть викликати шум на цій стадії, тому це впливає на продуктивність під час виявлення атаки. Таким чином, етап попередньої обробки повинен бути застосований перед перенесенням будь-яких алгоритмів навчання на джерело даних. На жаль, багато досліджень у цій галузі не згадували етапи попередньої обробки [17]. Вони лише неявно представляють процес аналізу журналу, який відповідає за перетворення файлу журналу в певний формат. У дослідженні магістерської роботи описуєтьсяце питання. Рис. 2.3 і табл.2.1. ілюструють кроки, які були задіяні в процесі попередньої обробки. Він включає сегментацію журналу та очищення даних у згенерованому форматі. Наступний підрозділ буде згадано та уточнено під час наступних кроків.

Таблиця 2.1 - Застосування процесу попередньої обробки

Попередня обробка даних	
Очищення даних	- Запити розширення звуків і зображень слід видалити - Зберігання запиту щодо статусу серії помилок - Видалення запиту, який не має параметра з успішним статусом
Після попередньої обробки	Атака буде згенерована та використана як вхідні дані для алгоритмів навчання

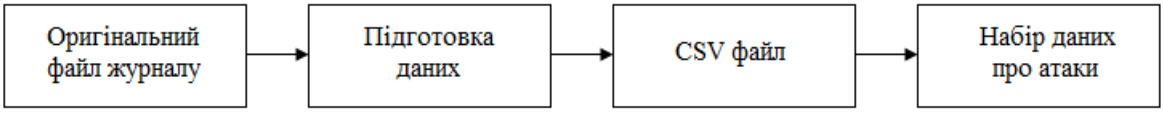


Рисунок 2.3 - Файл журналу з етапами попередньої обробки

Підготовка даних. Через те, що файл журналу є неструктурованим текстом, нам потрібно розібрати записи журналу в структуроване представлення.

Отримання шаблону навігації користувача стає можливим для навчання моделі навчання на цьому формальному представленні [13]. Попередні методи аналізу журналу нехтували значеннями записів журналу з часовими мітками та використовували статичний ключ журналу для виявлення значень аномалії. Під час нашого дослідження всі записи, знайдені у файлі журналу, зберігатимуться з метою застосування наступного:

- створення файлу CSV під назвою набір даних, де номери та назви стовпців у цьому файлі відповідають записам журналу, представленим у зібраних файлах журналу;

- застосування регулярних виразів до зібраних файлів журналу, щоб згрупувати значення кожного запису журналу в окрему групу. Назва кожної групи збігається з заголовком, отриманим із файлів журналу;

- зіставлення виведеної групи після застосування регулярних виразів до відповідних стовпців у файлі CSV.

Після цього етапу застосовано додаткові методи попередньої обробки, представлені пізніше у виведеному файлі CSV, як показано на рис.2.4, щоб порівняти різні алгоритми навчання.

host	time	request	status	bytes	user_agent
92.184.100.14	31/Dec/2017:05:13:57	GET /css/fonts/slick.woff HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:13:57	GET /css/fonts/slick.ttf HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:13:57	GET /css/ajax-loader.gif HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:13:58	GET /favicon.ico HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:14:34	GET /css/ajax-loader.gif HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
54.36.150.183	31/Dec/2017:05:14:35	GET /robots.txt HTTP/1.1	404	-	Mozilla/5.0 (compatible; AhrefsBot/5.2; +http://ahrefs.com/robot/)
92.184.100.14	31/Dec/2017:05:14:47	GET /css/fonts/slick.woff HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
92.184.100.14	31/Dec/2017:05:14:47	GET /css/fonts/slick.ttf HTTP/1.1	404	-	Mozilla/5.0 (Linux; Android 6.0.1; SAMSUNG SM-G935F Build/MMB29K) AppleWebKit/537.36 (KHTML, like Gecko)
207.46.13.205	31/Dec/2017:05:17:18	GET /robots.txt HTTP/1.1	404	-	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)

Рисунок 2.4 - Приклад даних на етапі попередньої обробки

Очищення даних. Процес очищення даних полягає у зміні ресурсу зберігання, у якому знаходиться створений файл CSV з попереднього етапу, таким чином ми гарантуємо, що збираємо лише необхідні дані. На цьому етапі спочатку ми почали з декодування запиту HTTP GET у символи ASCII, щоб перетворити

його на нижній регістр і видалити числові значення. Крім того, була запропонована нова техніка для відхилення HTTP-запитів GET, показана таким чином:

- повернення необробленого коду статусу, який містить число 200, без рядка запиту;

- очищення статичних запитів, таких як (наприклад, 'jpg', 'png', 'gif', 'webp', 'cr2', 'tif' та інші).

Далі решту запитів HTTP GET було проаналізовано, щоб отримати запит із URL-адреси, яка з'являється після «?» позначене, наприклад, «GET /ccne.php?id=2 HTTP/1». Нарешті, дубльований текст має відображатися як один запис у нашому наборі даних, оскільки різні запити можуть виглядати ідентичними після застосування попередніх методів очищення. Дані, зібрані на цьому етапі, а також ресурси використані для створення набору даних, показаного на рис.2.5, який складається з трьох типів атак {XSS, SQLI, Path-Traversal}.

payload	attack_type
/../../../../../../../../../../etc/passwd	path-traversal
/etc/passwd	path-traversal
file:/etc/passwd	path-traversal
.....etcpasswd	path-traversal
file:.....etcpasswd	path-traversal
....//....//....//....//....//....//....//....//....//.	path-traversal
/....//....//....//....//....//....//....//....//....//	path-traversal
//../....//....//....//....//....//....//....//....//.	path-traversal
1' where 6406=6406;select count(*) from rdb$f	sqli
1) and 8514=(select count(*) from domain.dom	sqli
-3136%') or 3400=6002	sqli
1) where 7956=7956 or sleep(5)#	sqli
-7387'))) order by 1--	sqli
1')) as gfzb where 7904=7904;begin dbms_lock.	sqli
1))) union all select null,null,null#	sqli
confirm(2)>/	xss
open()>	xss
location=?javascript:alert(1)>click	xss
<svg><script>alert(/1/)</script>	xss
</script><script>alert(1)</script>	xss
5rt(0);'>rhainfosec	xss

Рисунок 2.5 - Частина згенерованого набору даних, включаючи три типи атак

2.2 Огляд методу представлення ознак

Проаналізовано процес вивчення вектора слова та документа з тексту, це відоме як вилучення ознак. Представлення функцій у текстових даних є активною дослідницькою сферою, у якій пропонується багато різних моделей архітектури в

обробці природної мови (NLP) для вилучення інформації з контексту та представлення вилучених знань вектором. Ці функції використовуються набором класифікаторів для навчання, після чого вводиться метод метанавчання, щоб визначити, який із попередньо навчених класифікаторів є надійним під час процесу навчання.

Використано техніку NLP, щоб застосувати метод функції, який базується на векторі абзаців, використовуючи підходи вектора слів для запиту HTTP GET. Крім того, цей підхід був реалізований, коли вектори слів проходили через завдання для передбачення наступного слова в реченні (шкідливі URL-адреси). Потім, хоча вектори слів спочатку знаходяться в довільному стані, вектори слів можуть реалізовувати семантику у формі непрямого результату під час завдання прогнозування.

На рис.2.6 можемо спостерігати структуру для вектора абзацу в прикладі впровадження SQL. Кожен абзац має інший вектор, що характеризується матрицею «D» як стовпець, і кожне слово (з рядка запиту) має інший вектор, що характеризується матрицею «W» як стовпець. Після цього для вектора слів і абзаців було виконано процес усереднення або конкатенації, щоб застосувати передбачення слова, пов’язаного з контекстом. Символ абзацу можна визначити як інше слово. Ми можемо пов’язати це зі спогадом, який залежить від теми абзацу або поточного контексту запам’ятовування пропущених [18].

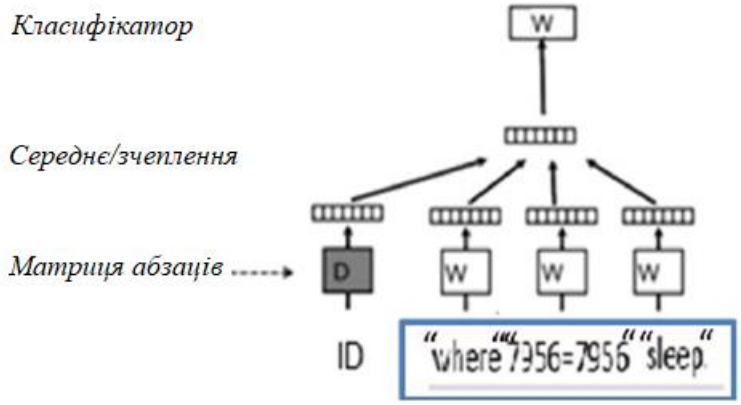


Рисунок 2.6 - Приклад досліджуваної моделі Doc2vec

Стохастичний градієнтний спуск є одним із основних факторів, який використовується на етапі навчання як для векторів абзаців, так і для векторів слів. Градієнт буде отримано за допомогою методу зворотного поширення. На кожній фазі навчання стохастичного градієнта контекст фіксованої довжини буде відбиратися через випадковий абзац для обчислення помилки градієнта, яка може виникнути у вибраній мережі. Таким чином, його можна використовувати для динамічного оновлення параметрів у моделі прогнозування. Дійсно, це є необхідністю в будь-якому процесі текстового передбачення.

Важливий крок, відомий як висновок, який використовується для обчислення вектора абзацу для нового абзацу, його можна виконати за допомогою коефіцієнта градієнтного спуску. Крім того, вектори слів (W) і ваги Soft Max, які використовуються в задачі прогнозування, повинні бути зафіксовані як параметри в моделі.

Припустімо, що ми маємо N як загальну кількість абзаців «витягнутий URL» і M як кількість слів «шкідливий код» у словнику. Наша мета полягає в тому, щоб вивчити вектори абзаців таким чином, щоб кожен абзац зіставлявся з розмірами P, а кожне слово – з розмірами Q. Після цього модель міститиме загальну кількість (N × P + M × Q) параметрів (за винятком параметрів SoftMax). Хоча при збільшенні N число параметра також збільшиться. Поки ми проводили оновлення на етапі навчання, результат буде ефективним.

Коли етап навчання завершено, ми можемо представити об’єкти в абзаці за допомогою векторів абзаців. Останнім часом функції можна додати до методів машинного навчання, таких як дерево рішень, логістична регресія, опорні векторні машини або KNN тощо.

Цей метод повинен складатися з двох основних етапів:

- етап отримання векторів слів W фазою навчання, а також вагових коефіцієнтів SoftMax і векторів абзаців D на основі вже відомих абзаців;

- етап отримання векторів абзаців D на етапі виведення, щоб передбачити нові абзаци, які ніколи раніше не бачили. Цей етап слід виконати шляхом додавання багатьох стовпців і градієнтного спуску в D.

Щоб завершити процес прогнозування, ми використовуємо D для кількох конкретних міток на основі класифікатора прогнозування машинного навчання. Перевага застосування цього методу полягає в тому, що модель може навчатися, використовуючи дані без міток, крім того, це може забезпечувати завдання прогнозування, якщо немає достатнього міченого типу даних.

Ми застосовуємо цей метод до згенерованого набору даних атаки, який обговорювався раніше, щоб перетворити URL-атаку з тексту на числові вектори. Через те, що машинне навчання можна застосовувати до числових даних, цей крок є важливим для класифікації запитів HTTP GET на різні типи атак за допомогою алгоритмів машинного навчання.

Модель DOC2VEC. Представлено методологію створення doc2vec для виявлення атак на запит HTTP GET. Наша мета — вивчити вбудовування, яке отримує властивості про порядок, у якому слова з'являються в запиті. Щоб досягти цього, ми спочатку визначаємо гіперпараметри doc2vec, які слід передати в цю модель, а саме:

- Dm=1: щоб визначити алгоритм навчання, наприклад «розподілену пам'ять» (PV -DM);

- size=300: Розмір векторів ознак;

- вікно=10: максимальна відстань між поточним і прогнозованим словом у реченні.

- альфа=0,025: початковий рівень навчання;

- min_alpha=0,025: Швидкість навчання буде лінійно падати до min_alpha у міру навчання;

- min_count =5: для ігнорування всіх слів із загальною частотою, нижчою за цю;

- вибірка=10e-5: Порогове значення для налаштування того, які слова з вищою частотою випадково знижуються, корисний діапазон становить (0, 1e-5);

- Negative=5: використовуватиметься негативна вибірка, значення для negative визначає, скільки «шумових слів» має бути намальовано (зазвичай між 5-20);

- dm_concat=1: використання конкатенації контекстних векторів замість суми/середнього значення;

- документ = набір навчальних даних, створений раніше.

Після того, як ми застосували цю модель до нашого набору даних, ми отримали 300-вимірне векторне представлення для кожного документа разом зі словником (вбудовування слів), який містить усі унікальні слова, знайдені в навчальному корпусі набору даних. Таблиця 2.2 показує кількість векторів документа та унікальне слово, які були отримані з цієї моделі.

Таблиця 2.2 - Вбудовані слова та числа векторів документів, отримані за допомогою моделі doc2vec

Вбудовування слів	Кількість векторів документа
1172	6000

Крім того, у цьому підрозділі виконано візуалізацію документа вбудовування запиту HTTP GET, витягнуті зі згенерованого набору даних. Ми використали весь набір даних із 6000 запитів, взятих із нашого набору даних, які складаються зі збалансованих класів розподілу, які були вибрані з наших даних, як представлено в табл.2.3. Ми витягли вектори ознак цих запитів із матриці D, яка була згенерована під час навчання моделі doc2Vec (навчання на 6000 запитах). Вибрані вектори ознак були отримані шляхом конкатенації векторів слів, виведених із слів, які з’явилися в одному документі. Ми отримали 300-вимірний вектор для кожного документа в навчальному корпусі та для кожного унікального слова, зібраного з набору даних під час процесу навчання. Для базових функцій — розподілена пам’ять у doc2vec із використанням CBOW під час генерації векторів слів із набору даних. Крім того, з виділених векторів ознак ми застосували t-SNE [17], щоб зменшити розміри ознак і побудувати запит HTTP GET у двовимірному просторі вбудовування. Представлення вбудованих запитів можна побачити на рис.2.7.

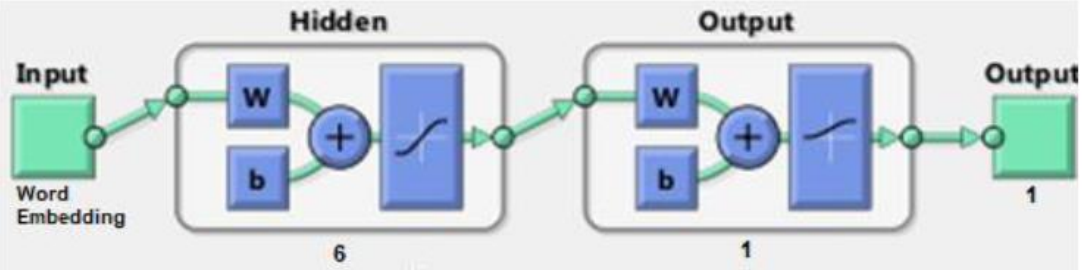


Рисунок 2.7 - Штучна нейронна мережа: модель MLP

Таблиця 2.3 - Кількість векторів документа для кожного класу

XSS	SQLI	Path-Traversal
1999	2003	1998

Як можна побачити на рис.2.8, для запитів запити XSS, SQLI та Path-Traversal чітко розділені на три групи запитів. Більшість запитів SQLI розташовано в правій частині графіка, тоді як запит XSS – у центральній частині, а Path-Traversal – у лівій частині графіка.

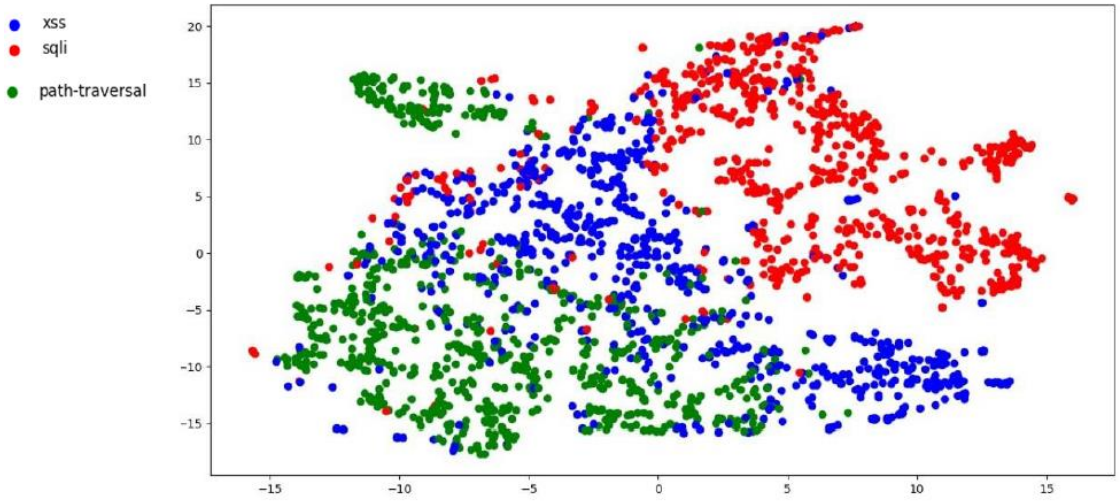


Рис.2.8 - Векторний простір документа

Дуже небагато точок даних різних запитів на атаку накладаються одна на одну. Крім того, ми помітили, що на сюжеті з’явилося кілька кластерів. Ми додатково проаналізували деякі з цих кластерів, щоб визначити потенційні шаблони в бажаному рядку, які, можливо, можуть вказувати на природу цієї атаки. Аналіз таких шаблонів може бути корисним для глибшого розуміння властивостей

запиту HTTP GET. Наприклад, запити, які містять такі фрази, як «вибрати *», «де» в рядку запиту або «від», були згруповані разом. Для фраз, які виглядали як «<script>», їх було згруповано в інший кластер. Ми все ще змогли розрізнити два кластери запитів; один для «<script>» у рядку запиту, а інший для «select *» у рядку запиту. Тому відмінність між різними текстами під час навчання була чіткою. З іншого боку, використовуючи наш метод функцій, doc2vec підтверджує та отримує значуще представлення та вбудовує ці запити в розріджену матрицю, де подібні векторні запити об'єднуються разом. Нарешті, з вибраною кількістю вимірів і без необхідності отримання експертних ознак вектор ознак є ефективним для представлення запитів. Для даних розміром 6000 запитів модель doc2vec не страждає від обмежень пам'яті для обробки та зберігання векторів, які можна використовувати для подальших завдань.

2.3 Прикладні методи машинного навчання та попередні відомості про класифікацію

Проведено пошук бажаних методів машинного навчання для застосування до нашого набору даних, щоб реалізувати IDS для виявлення найпопулярніших атак на веб-сервер.

У даному підході застосовується техніка вбудовування документів (модель doc2Vec), витягнувши 6000 векторів документів, навчених doc2vec. Крім того, застосовується чотири різні алгоритми (дерево рішень, багатошаровий персептрон (MLP), опорна векторна машина та класифікатор KNeighbors для класифікації витягнутих URL-адрес, представлених у запиті HTTP GET, який дозволяє нам виявляти ці зловмисні атаки. Тому, щоб оцінити застосовані моделі, ми почали з поділу набору даних на дві частини: 80% для навчання та 20% для тестування, як представлено в табл.2.4. Крім того, інтелектуальні моделі, показані згодом, були протестовані з численними показниками оцінки, такими як точність, специфічність точності , нагадаємо, площа під кривою (ROC) і матриця плутанини Метрики були

згенеровані для кожного класифікатора машинного навчання, які включають важливі докази про існуючі та виявлені класи атак.

Таблиця 2.4 - Запропонована пропорція розподілу для етапу навчання та тестування

	Навчальний набір даних	Тестування набору даних
Створено набір даних	80%	20%

Крім того, під час етапу навчання та в кожній моделі застосовується кілька параметрів продуктивності для класифікатора машинного навчання, зазначених нижче:

1. Модель дерева рішень. Дерево рішень схоже на структуру дерева, що складається з вузлів, які формують кореневе дерево. Основною метою алгоритму ID3 є побудова дерева рішень за допомогою блок-схеми, починаючи зверху вниз. Він перевіряє кожен параметр у будь-якому вузлі, наприклад ентропію для дослідницького аналізу. Ця властивість відокремить фазу навчання з урахуванням їх цільової класифікації. Щоб досягти цього, ентропія була перевірена за формулою (2.1).

$$\text{Entropy} = -\sum_{i=1}^{n} p_i * \log(p_i) \quad (2.1)$$

де і — кількість усіх класів, а p_i — ймовірність класу I. У цій моделі ми вибрали такі параметри:

- Criterion= Entropy;
- Splitter=Best;
- Max_depth=None;
- Class_weight= Balanced;
- Max_features= None.

2 Модель MLP. Штучна нейронна мережа (ШНМ) вважається однією з новітніх технологій у сфері штучного інтелекту (ШІ). Нейронна мережа може обробляти будь-яку інформацію так само, як людський мозок. Мережа збирається

з величезної кількості оброблених взаємопов'язаних нейронів, які працюють паралельно. Це може вирішити багато наукових проблем, щоб прийняти штучні рішення. Таким чином, ми використали багаторівневий перцептрон (MLP) у ШНМ, яка є мережею з численними рівнями. Ці шари виконують кілька ролей і функцій. Мережа включає вхідний рівень, прихований рівень і мережевий вихід, який називається вихідним рівнем. Крім того, математично класифікатор MLP, показаний на малюнку [47], обчислюється відповідно до функції активації Rectified Linear Unit (2.2).

$$"Relu(x) = \max(x, 0)" \tag{2.2}$$

Де *x* — елемент, а *Relu(x)* — функція, яка досягає максимуму цього елемента з нулем.

Таким чином, наша модель була перевірена з векторами слів як вхідним шаром, прихованим шаром і вихідним шаром відповідно до таких параметрів:

- активація=Relu як функція випрямленої лінійної одиниці;

- Solver=adam як розв'язувач для оптимізації ваги;

- Hidden_layer_sizes =6, тобто кількість нейронів у прихованому шарі;

- швидкість навчання = 0,001 розклад для оновлення ваги;

- random_state= пакетна вибірка для точного визначення випадкового числа для ваг і ініціалізації зсуву;

- імпульс =0,9 як оновлення градієнтного спуску;

- альфа=0,0001 для уникнення переобладнання шляхом штрафних ваг;

- Max_iter =1000, що є максимальною кількістю ітерацій; розв'язувач виконує ітерації до збіжності на цю кількість ітерацій.

3. Модель SVM. Алгоритм навчання під керівництвом. Він працює на основі розділення межею лінійно роздільних класів, а потім узагальнює це в нелінійні межі, змінюючи простір.

У цій моделі обчислюються розділювачі гіперплощин (межі прийняття рішення) за формулою $h(x) = x^T\beta + \beta_o = 0$. Потім ми обчислили відстань від точки до гіперплощини за такою формулою $d(x) = \frac{|x^T\beta+\beta_o|}{||\beta||}$знаючи, що $\left|\left|\beta\right|\right| =$

$\sqrt{\beta_1^2 + \cdots + \beta_p^2}$ максимізація суми запасу для мінімізації норми вектора параметрів β. SVM можна використовувати у двох підходах, які є проблемами класифікації та регресії. У нашій моделі під час етапів навчання та тестування ми використовували такі параметри:

- Ядро = лінійне: з рівнянням: $k(x_i , x_j) = x_i . x_j$

- Tol=0,001 Критерій допуску для зупинки: 1e-3

- gamma='scale', який використовує 1 / (n_features * X.var()) як значення гами

- cache_size=200 як розмір кешу ядра

- max_iter=- 1 без обмежень.

4. Модель KNN. У цій моделі ми застосували контрольоване навчання, яке є технікою KNN, яка використовується для проблем класифікації. Він працює на основі припущення точок даних ідентичних класів, які розташовані ближче одна до одної. Ми застосували наступні кроки для виявлення класів атак:

1. Крок 1: слід визначити значення K.

2. Крок 2: ми розрахували відстані між точками даних тестування та тренування за допомогою метрики Мінковського.

3. Крок 3: ми відсортували відстані, щоб встановити k найближчих сусідів відповідно до найменших значень відстані.

4. Крок 4: ми проаналізували сусідів, щоб призначити відповідну категорію для нових даних (тест) за допомогою більшості голосів.

5. Крок 5: ми досягли визначеного класу.

У даному підході ми протестували нашу навчену модель із такими параметрами:

- Вага = рівномірна функція, яка використовується в процесі виявлення

- Метрика= Мінковського з p=2, яка є стандартною евклідовою метрикою

- n_neighbors=2 Кількість сусідів для використання

- algorithm='auto' Алгоритм, який використовується для обчислення найближчих сусідів

- n_jobs=1 кількість паралельних завдань, що дозволяє шукати сусідів

У наступному розділі ми почнемо обговорення результатів кожної моделі, щоб провести порівняльне дослідження

Таким чином, ми доведемо найкращу точну та ефективну модель серед методів hosen ML.

2.4 Експериментальні результати та їх обговорення

Виконано оцінку продуктивності досліджуваного HIDS за допомогою чотирьох методів машинного навчання: дерева рішень, багатошарового персептрона (MLP), опорної векторної машини та класифікатора K-сусідів. Наша мета — класифікувати найбільш шкідливі атаки, з якими може зіткнутися веб-сервер.

Тому, щоб оцінити застосовані HIDS, перевіпено різні показники оцінки, такі як рівень виявлення, точність, точність, запам’ятовування, площа під кривою (ROC) і матриця плутанини. Визначено, що кожного разу, коли використовується досліджуваний алгоритм, згенеровані результати кожної моделі явно відрізняються.

Таким чином, табл.2.6. ілюструє рівень виявлення для кожної атаки, яка була протестована з використанням запропонованих IDS.

Таблиця 2.6 - Найвищий рівень виявлення, досягнутий класифікаторами для кожної атаки

		Класифікатор			
		Decision Tree	MLP	SVM	KNN
Тип атаки	Path-Traversal	0.8214	0.9260	0.9541	0.9184
	SQLI	0.9215	0.9469	0.8776	0.9330
	XSS	0.8320	0.8400	0.6347	0.7840

Можемо зробити висновок, що класифікатор SVM досяг найвищого показника 95,41% у виявленні атаки Path-Traversal. Класифікатор MLP досяг найвищого рівня виявлення як для SQLI, так і для XSS, відповідно, 94,69%, 84%. З

іншої точки зору, SVM досяг найвищого рівня виявлення для обходу шляху, де він досяг найнижчого показника виявлення XSS з 63,47%.

На рис. 2.9-2.12 показано площу під робочою характеристикою приймача (ROC) для кожної застосованої моделі, яка була розрахована на основі істинно позитивних і хибно позитивних результатів. Велике значення ROC показує здатність моделі виявляти зловмисники, тоді як нижче значення демонструє слабкість цієї моделі.

Що стосується області під ROC кожної моделі, ми можемо зробити висновок, що класифікатори MLP зареєстрували найвищий показник 93%, тоді як KNN зафіксував 91%, дерево рішень (90%) і SVM показали найнижче значення (87%).

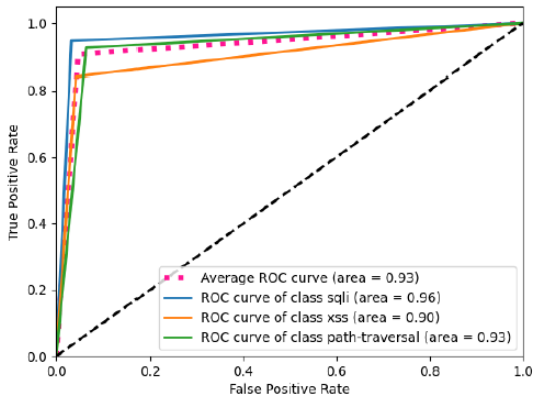


Рисунок 2.9 - Крива MLP ROC

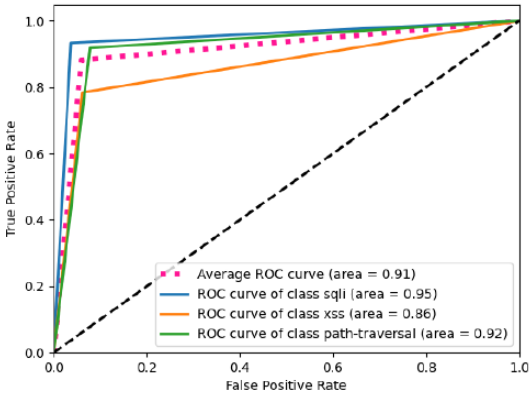


Рисунок 2.10 - Крива KNN ROC

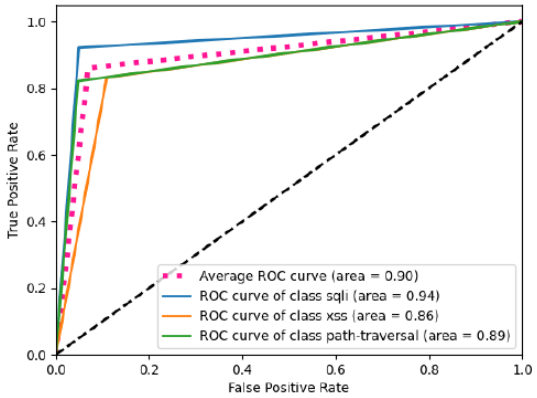


Рисунок 2.11 - Крива ROC дерева рішень

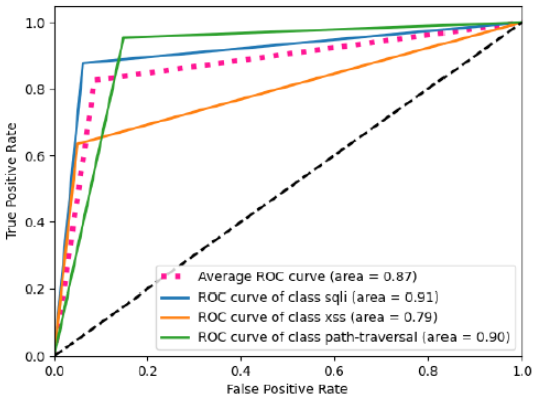


Рисунок 2.12 - Крива SVM ROC

Серед перевірки 6000 записів, існуючих у нашому наборі даних. Середній показник точності для кожного обраного класифікатора наведено в табл.2.6. Ми

можемо зробити висновок, що класифікатор MLP є найточнішою моделлю з точністю 90,67%, тоді як SVM є найнижчою з точністю 82,67%.

Таблиця 2.6 - Результати точності різних методів ML

Класифікатори машинного навчання	Точність
MLP	0.9067
KNN	0.8817
Decision Tree	0.8608
SVM	0.8267

Використовуючи модель дерева рішень та проаналізувавши табл.2.7. і

На рис.2.13 виявлено, що модель дерева рішень записала середні значення в даних тестування.

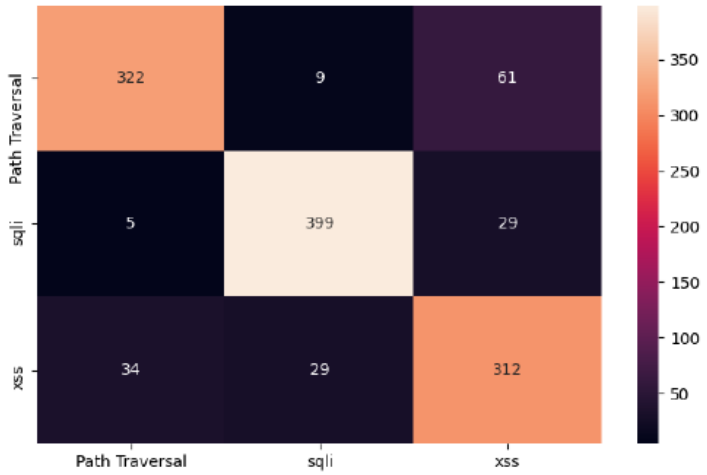


Рисунок 2.13 - Матриця плутанини для моделі дерева рішень

Таблиця 2.7 - Звіт про класифікацію для моделі дерева рішень

	Точність	Відклик	
Path Traversal	0,8920	0,8214	
SQLi	0,9130	0,9215	
XSS	0,7761	0,8320	
Accuracy			0,8608
Macro avg	0,8604	0,8583	
Weighted avg	0,8634	0,8608	

Атака обходу шляху мала рівень точності 89,20%, 82,14% як рівень відкликання, SQLI отримує найвищий бал точності з 91,30%, 92,15% для рівня

відкликання, і, нарешті, XSS досягає найнижчої точності з 77,61% під час відкликання досягає 83,20%. Крім того, модель дерева рішень зафіксувала середнє значення точності 86,04%, а площа ROC досягає середнього значення 90%.

Крім того, згідно з моделлю MLP і після аналізу табл.2.8. і рис.2.14. можливо зробити висновок, що ця модель досягла значень покращення на етапі тестування, де атака обходу шляху мала низький рівень точності (87,47%), SQLI отримує найвищий показник точності (94,25%), і нарешті XSS досягає рівня 90%. Результати відкликання для атак проходження шляху, SQLI та XSS становили 92,60%, 94,69% та 84% послідовно. Ця модель зафіксувала збільшення середнього значення точності на етапі виявлення на 90,57%, а також площа під ROC отримує середнє значення на 93%.

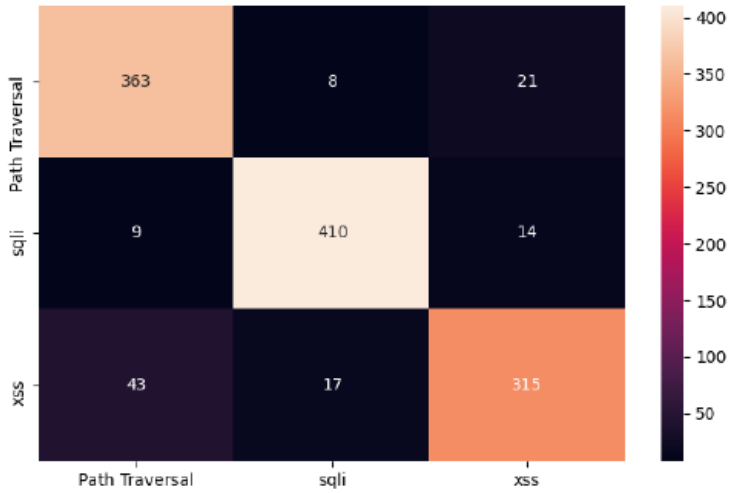


Рисунок 2.14 - Матриця плутанини для моделі MLP

Таблиця 2.8 - Звіт про класифікацію для моделі MLP

	Точність	Відклик	
Path Traversal	0,8747	0,9260	
SQLi	0,9425	0,9496	
XSS	0,9000	0,8400	
Accuracy			0,9067
Macro avg	0,9057	0,9043	
Weighted avg	0,9071	0,9067	

Крім того, виявлено, що модель SVM була розширена в деяких класах на етапі тестування після вивчення табл.2.9. і рис.2.15.

Таблиця 2.9 - Звіт про класифікацію для моделі SVM

	Точність	Відклик	
Path Traversal	0,7571	0,9541	
SQLi	0,8899	0,8776	
XSS	0,8530	0,6347	
Accuracy			0,8267
Macro avg	0,8334	0,8221	
Weighted avg	0,8350	0,8267	

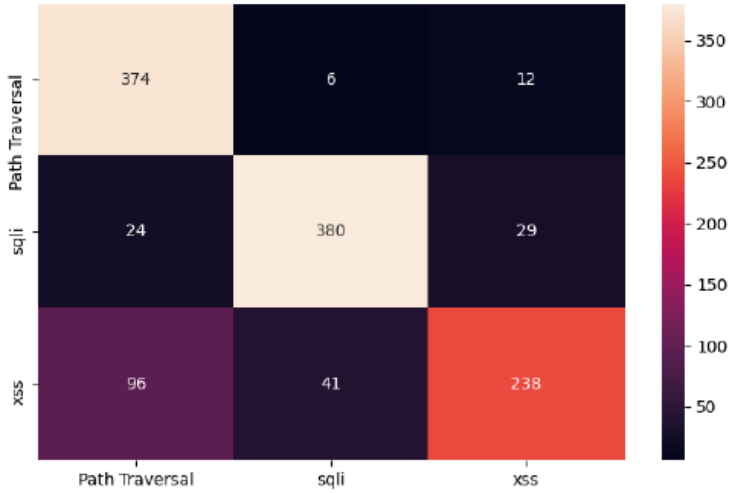


Рисунок 2.15 - Матриця помилок для моделі SVM

Наприклад, атака з обходом шляху мала низький рівень точності 75,71%, SQLI досягає показника точності 88,99%, а XSS становить приблизно 85,30%. Результати відкликання для атак проходження шляху, SQLI та XSS становили 95,41%, 87,76% та 63,47% відповідно. Крім того, ця модель зафіксувала 83,34 % як середнє значення точності та 87 % як середнє значення площі під показником ROC.

Нарешті, проаналізувавши табл.2.10 і рис.2.16, зроблено висновк, що модель KNN досягла нормальних значень у даних тестування, атака обходу шляху мала низький рівень точності (85,11%).

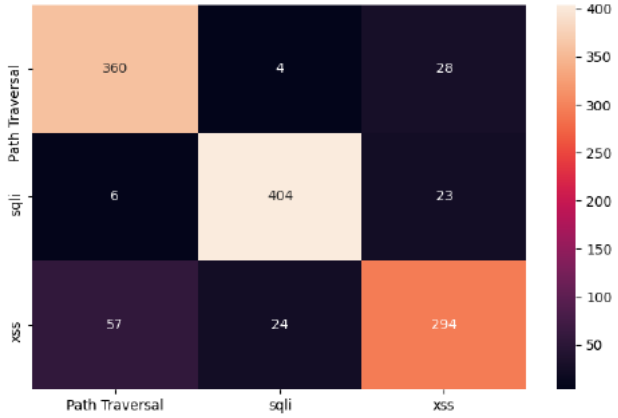


Рисунок 2.16 - Матриця плутанини для моделі KNN

Таблиця 2.10 - Звіт про класифікацію для моделі KNN

	Точність	Відклик	
Path Traversal	0,8511	0,9184	
SQLi	0,9352	0,9330	
XSS	0,8522	0,7840	
Accuracy			0,8817
Macro avg	0,8795	0,8785	
Weighted avg	0,8818	0,8817	

SQLI досягає високої точності в 93,52%, тоді як атака XSS займає 85,22%. Результати відкликання для атак проходження шляху, SQLI та XSS склали 91,84%, 93,30% та 78,40% послідовно. Ця модель зафіксувала середнє значення 87,95% як значення точності на етапі виявлення, а площа під ROC отримує середнє значення 91%.

РОЗДІЛ 3 ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ БЕЗПЕКИ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ

3.1 Загальний огляд та вимоги до інтелектуальних систем безпеки виявлення вторгнень

Системи виявлення вторгнень (IDS) і системи запобігання вторгненням (IPS) вважаються одними з найважливіших міркувань кібердослідників через їх ефективність у виявленні та запобіганні будь-яким новим кібератакам у реальній мережі проти різних атак і аномальна діяльність. Оцінка вхідних наборів кіберданих має важливе значення для ефективності будь-якого підходу до системи виявлення вторгнень. Це дозволяє нам оцінити запропоновані методи, які є кваліфікованими для виявлення та запобігання кібератакам. Підготовку наборів даних для мережевої IDS може бути нелегко отримати через питання конфіденційності та конфіденційності. Крім того, найбільшою проблемою є відсутність публічних, доступних і позначених наборів даних, які використовуватимуться на етапі навчання, що базується на техніках машинного навчання. У цій частині зупинилися на виборі надійних наборів даних, які було вирішено використовувати для реалізації SIS-ID. У магістерській роботі система SIS-ID була досліджена шляхом посилання на набори даних, витягнуті з організації CIC, вибраної та названої таким чином: DB-MALCURL з використанням набору даних ISCX-URL-2016 [19] і DBDDOS з використанням набору даних DDOS2019.

Таблиця 3.1 - Розподіл класів для DB-MALCURL

DB-MALCURL	
Тип атаки	Кількість рядів
Benign	7530
Spam	7735
Phishing	7945
Malware	6670
Defacement	6820

Підготовка набору даних DB-DDOS. Атака розподіленої відмови в обслуговуванні (DDoS) вважається однією з найбільш загрозливих атак для сфери безпеки мережі. Він спрямований на виснаження мереж із великою кількістю шкідливого трафіку. Хоча для виявлення DDoS-атаки було призначено кілька інструментів, низька обчислювальна продуктивність наразі вважається основною проблемою для вчених із кібербезпеки для розробки систем виявлення вторгнень на основі методів машинного навчання, які значною мірою покладаються на надійний набір даних. Таким чином, DB-DDOS підготовлено з використанням набору даних CICDDoS2019 для реалізації моделей SIS-ID для виявлення атак DDOS. Цей набір даних охоплює колекцію останніх атак мережевих потоків DDOS, перелічених у табл.3.2. з 80 функціями, отриманими з фреймворку CICFlowMeter-V3, який містить 13 класів атак DDOS, запропонованих у розподілі, як показано в наступній таблиці.

Таблиця 3.2 - Розподіл класів для DB-DDOS

DB-DDOS	
Тип атаки	Кількість рядів
BENIGN	55665
DrDoS_DNS	55975
DrDoS_LDAP	55875
DrDoS_MSSQL	56050
DrDoS_NetBIOS	55635
DrDoS_NTP	56325
DrDoS_SNMP	56310
DrDoS_SSDP	56625
DrDoS_UDP	55930
Syn	55910
TFTP	55890
WebDDoS	55590

3.2 Аналіз реалізації даних і функцій

У даному підрозділі представлено інженерні методи даних і проектованих функцій, які використовуються при реалізації системи SIS-ID. Дійсно, кілька факторів можуть впливати на результати алгоритму машинного навчання.

Тому на етапі реалізації використано дані і функції для покращення продуктивності ML на основі інтелектуальної моделі.

Попередня обробка даних. Метод попередньої обробки даних є критичним етапом у машинному навчанні має на меті підвищити якість даних, щоб покращити вилучення корисної інформації з бажаних даних. Ця фаза готує екземпляри, обробляючи їх у належній формі. Таким чином, у цій частині етап попередньої обробки розглядався за допомогою кількох методів, що стосуються перетворення підмножин у читабельний, зрозумілий і чистий формат, оскільки дані зібрані з джерел і можуть містити шумні, порожні або нерелевантні дані, які потенційно можуть знизити продуктивність системи SIS-ID. Етапи попередньої обробки даних вказані на рис.3.1 і перераховані таким чином:

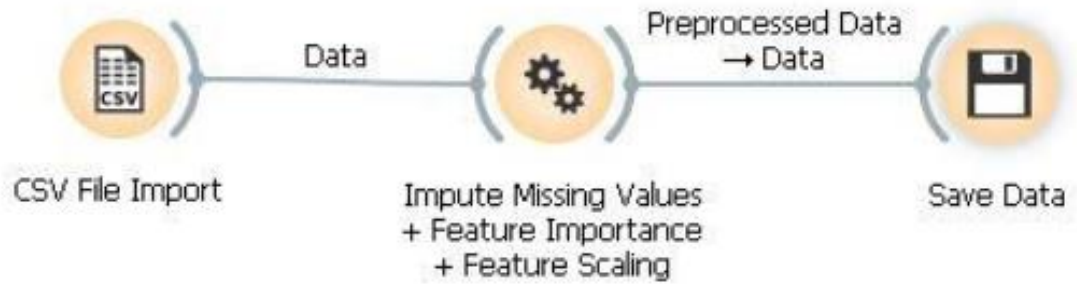


Рисунок 3.1 - Процес попередньої обробки даних

1. Етап очищення даних для видалення та обробки відсутніх значень шляхом заміни відсутніх значень значущими.

2. Збалансування підмножини за допомогою методів недостатньої та надмірної вибірки, пояснених нижче. Незбалансовані дані зазвичай стосуються набору даних, у якому класи мають велику різницю в розмірі.

Кілька наборів даних мають незбалансовану кількість екземплярів у своїх класах. Щоб подолати цю проблему, був застосований метод вибірки з використанням бібліотеки scikit-learn [22] із двома основними алгоритмами, визначеними таким чином:

- надмірна вибірка: дублювання зразків із недостатньо представлених класів за допомогою «imblearn.over_sampling import RandomOverSampler»;

- недостатня вибірка: видалення зразків із надмірно представлених класів за допомогою «imblearn.under_sampling import RandomUnderSampler».

3. Трансформація даних шляхом кодування категоріальних ознак, а також їх масштабування. Як правило, більшість набору даних IDS включає функції з різним обсягом і діапазоном. Дійсно, набір даних включає функції з кількома вимірами та діапазоном. Особливості з високими величинами можна виміряти за допомогою евклідової відстані між двома точками даних. Тому для подолання цієї проблеми слід застосувати методи масштабування серед даних, і вони повинні бути доведені до однієї шкали величин [23].

Розглянутий метод масштабування було застосовано для системи SIS-ID за допомогою методу «StandardScaler».

Він стандартизує ознаки шляхом виключення середнього значення та масштабування його до одиничної дисперсії, де стандартна оцінка зразка x обчислюється за цією формулою «Z= (X - U) /S»; де U є середнім значенням навченого екземпляра (x), а S вважається стандартним відхиленням $\frac{xi-mean(x)}{stdev(x)}$, як виведено, наприклад, у табл.3.3.

Таблиця 3.3 - Приклад результату методу масштабування

	Querylength	domain_token_count	path_token_count	avgdomaintokenlen	avgpathtokenlen	charcompvowels
1	-0.232407712	0.158097788	-0.223720996	2.148138384	0.28908439	0.151258571
2	-0.030905003	0.158097788	0.010092389	-0.537537372	-0.0203342	0.076355921
3	-0.232407712	-0.946569866	0.243905774	1.308864569	0.651546168	0.151258571
4	0.000911215	1.262765441	0.010092389	-0.705392185	-0.668639807	0.151258571
5	-0.211196901	1.262765441	-0.691347767	-0.453610091	-0.815982004	-0.672670581
6	-0.232407712	0.158097788	-1.392787923	-0.537537372	-1.14013479	-1.047183831
7	-0.179380684	1.262765441	-0.457534382	-0.831283232	0.156476423	-0.223254679
8	-0.168775278	-0.946569866	1.880599472	0.301736192	0.321499683	1.574408925
9	-0.232407712	-0.946569866	-1.392787923	1.560646663	2.749698778	-0.897378531

Техніка функцій. У цій частині розглянуто метод вибору функцій, який використано в системі SIS-ID для покращення ефективності результатів. Тому запропоновано рекурсивне усунення функцій на основі рейтингу функцій, яке використовує метод перехресної перевірки «feature_selection.RFECV».

Вибрані функції. Техніка рекомендованих функцій призначає оцінку важливості або вагу кожній вибраній функції в системі SIS-ID. Таким чином,

функції з найнижчою оцінкою важливості та дисперсією виключені з даних, а модель навчено відповідно до цієї методики: 1. Набір даних DB-MALCURL; 2. Набір даних DB-DDOS.

3.3 Методологія навчання для системи SIS-ID

Досліджувана SIS-ID є інтелектуальною системою, розробленою з використанням мови Python. Розгорнута для виявлення останніх кібератак за допомогою BDD-MALCURL і DB-DDOS, які використовуються на етапі навчання системи, яка представлена у роботі. Метод навчання викладено в наступній частині з використанням 80% кожного джерела даних для етапу навчання, який, безумовно, займає найбільшу частину, яку можна використати для навчання або підгонки моделі, щоб знайти оптимальні параметри, що впливають на неї. Однак 20% даних, що залишилися, використані для етапу тестування, необхідного для перевірки SIS-ID для неупередженої оцінки, тоді як цей набір буде назавжди виключено з екземплярів навчання для порівняння їх результатів із фактичними класами. Дійсно, щоб досягти високого рівня виявлення, запропонований SIS-ID використовує набори для навчання та тестування, як зазначено в наступній методології, проілюстрованій на рис.3.2.

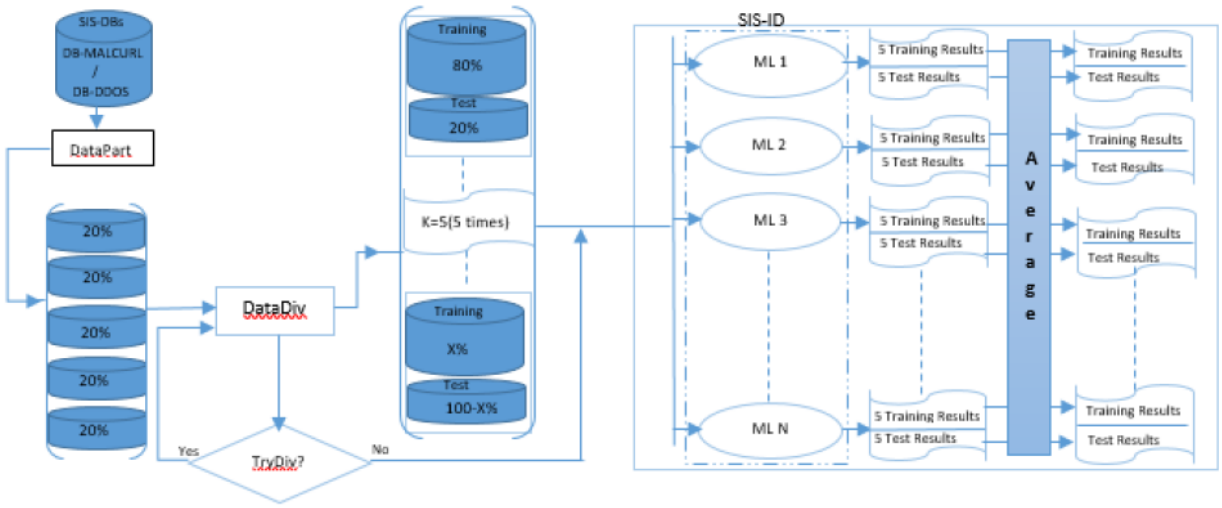


Рисунок 3.2 - Загальна архітектура методології навчання SIS-ID на основі прикладних методів машинного навчання

Прикладні методи машинного навчання. Нижче опишемо методи навчання в пропозиції системи SIS-ID з використанням кількох методів машинного навчання, які були перевірені на наборах даних, описаних вище, як-от BDD-MALCURL і DBDDOS.

Таким чином, моделі машинного навчання вибрані на основі контрольованого навчання з використанням k-найближчих сусідів (KNN), дерева рішень і двох багатокласових методів, які є моделями OneVsRest і OneVsOne. Ми також розвинули методи ансамблю, використовуючи п’ять моделей, таких як Voting, Stacking, Bagging, XGBoost, Random Forest і Adaboost на основі визначеної методології, яка була використана для оптимізації системи, що застосовується до кожного набору даних, як зазначено нижче:

1 Контрольоване навчання.

- дерево рішень;
- алгоритм k-найближчого сусіда;
- багатокласні прийоми.
- класифікатор один проти одного
- Класифікатор один проти решти.

2. Ансамблеві прийоми. Ця техніка поєднує деякі слабкі базові моделі, щоб створити потужну та ефективну модель. У цьому експерименті було розгорнуто шість моделей ET, які включають екстремальне посилення градієнта (XGBoost), випадковий ліс, adaboost, пакетування, голосування та стекування:

- XGBoost
- випадковий ліс
- класифікатор Adaboost

3. Розвиток техніки ансамблю:

- класифікатор мішковий;
- класифікатор голосування.

4. Навчання без контролю

- Фактор локального виключення (LOF).

Коефіцієнт локального відхилення (LOF) вважається однією з найбільш використовуваних моделей у методі виявлення аномалій через важливість подолання проблеми невідомого надходження трафіку до мережі.

Тому інтерес полягав у тому, щоб запропонувати правильно налаштувати SIS-ID, щоб уникнути будь-якої атаки, з якою може зіткнутися сервер. Тому витягнуто з DB-DDOS 60000 записів, пов'язаних із доброякісними екземплярами, щоб навчити модель нормального трафіку. Таким чином, система навчилася, використовуючи безпечний трафік (inlier), і її можна перевірити в реальному часі на майбутні аномальні дії (outlier). З іншого боку, витягнуто 40 000 записів із кількох екземплярів DDOS, щоб довести його ефективність на етапі тестування для виявлення будь-якої наступної атаки. Оскільки LOF є технікою неконтрольованого навчання, локальне відхилення щільності слід вимірювати відповідно до точки даних, обчисленої з її сусідами. Отже, зразок із кінцевою нижчою інтенсивністю, ніж його сусіди, вважатиметься викидом (атакою), де LOF є оцінкою кожного зразка аномалії. Модель вибрала місцевість для точки даних методом k-найближчих сусідів, щоб обчислити відстань і оцінити локальну інтенсивність.

Реалізація навчання. У досліджуваній системі SIS-ID застосовується кілька моделей класифікації машинного навчання на етапі навчання, як зазначено у вищезазначеній частині. Щоб подолати тривалий час під час перевірки кожного блоку, застосовано методологію навчання відповідно до запропонованого псевдокоду, представленого на рис.3.3.

1. Define set of hyper-parameter combinations, C, for current model. If model has no hyper-parameters, C is the empty set.
2. Divide data into K folds with approximately equal distribution
3. (outer loop) For fold ki. in the K folds:
    1. Set fold ki as the test set
    2. For parameter combination c in C:
        1. (inner loop) For fold ki in the remaining K-1 folds:
            1. Set fold ki as the validation set
            2. Train model on remaining K-2 folds
            3. Evaluate model performance on fold ki.
        2. Calculate average performance over K-2 folds for parameter combination c
    3. Train model on K-1 folds using hyper-parameter combination that yielded best average performance over all steps of the inner loop
    4. Evaluate model performance on fold ki.
4. Calculate average performance over K folds

Рисунок 3.3 - Застосований псевдокод реалізації навчання SIS-ID

Таким чином, проаналізована реалізація навчання, процес навчання та оцінки для кожної моделі з використанням техніки перехресної перевірки. Він складається з двох циклів для фаз навчання та оцінки [24].

На початку досліджуваної системи навчання і DB-MALCURL, і DB-DDOS були розділені на k згорток (k = 5 у нашому дослідженні), які мали приблизно однакові розміри. Потім у зовнішньому циклі для кожної ітерації ми взяли 20% як одне згортання даних, яке потрібно зарезервувати для етапу тестування. Таким чином, згортки, що залишилися (k-1), потрапляють у внутрішній цикл, щоб виконати налаштування гіперпараметрів як автоматизовану модель, і їх слід систематично вибирати окремо від екземплярів оцінки. Для кожної моделі внутрішній цикл включає пошук параметрів у сітці, і їх значення необхідно оцінити, тоді як кожен параметр оцінюється за допомогою кроку перехресної перевірки (k-1). Крім того, обрані гіперпараметри, які досягли вищої середньої перехресної перевірки. Отже, SIS-ID буде навчено на основі вибраних даних у межах (k-1) складок на основі найкращих досягнутих параметрів, які будуть оцінені відповідно до його продуктивності виявлення для стійкої складки у зовнішній контур. Зрештою цю процедуру слід повторити k разів (k=5), щоб кожна згортка даних зовнішнього циклу використовувалася один раз, що призводить до k оцінки продуктивності нашої системи.

Метод оптимізації навчання. Оскільки досліджувана інтелектуальна система безпеки базується на техніці машинного навчання, ми зіткнулися з серйозним завданням у реалізації системи SIS-ID, яка полягає у виборі техніки оптимізації моделі. Пріоритетом було знайти найкращу теорію для техніки оптимізації моделювання, яка буде присвячена на етапі навчання. Таким чином, техніка оптимізації гіперпараметрів була призначена на нашому етапі навчання, що є завданням метаоптимізації, кожне випробування конкретного налаштування гіперпараметра включатиме навчання моделі, і це може бути процес внутрішньої оптимізації. У результаті цього методу було досягнуто найкращого набору комбінацій параметрів, який впливає на обрані нами алгоритми для забезпечення найкращої продуктивності в наборі перевірки, зазначеному нижче:

Гіперпараметр миттєво впливає на фазу навчання системи. Таким чином, використано GridSearchCV, який проходить через попередньо визначений словник гіперпараметрів, щоб відповідати моделі для досягнення потужних змінених параметрів і вказувати кількість разів для параметра перехресної перевірки для кожного набору гіперпараметрів. Вибрані параметри для об'єкта GridSearchCV перераховані, як показано нижче:

- Оцінювач: обрана модель.
- Params_grid: параметри моделі словника, яка містить гіперпараметр.
- Оцінка: метрика оцінювання.
- N_jobs: кількість процесів, які будуть виконуватися паралельно.
- cv: номер методу перехресної перевірки, який у нашій системі становить 5.
- Verbose: це виправлено на 1, щоб отримати детальний вихід, поки ми підганяємо дані до об'єкта GridSearchCV.

3.4 Експерементальні дослідження та оцінка продуктивності SIS-ID

Застосування системи SIS-ID на DB-MALCURL.

У даному підрозділи наводяться результати отримані системою SIS-ID, застосованою до DB-MALCURL з використанням контрольованого навчання та методів ансамблю.

Контрольоване навчання. Моделі були досліджені за допомогою кількох контрольованих алгоритмів навчання, як: дерево рішень, k-найближчі сусіди (KNN) і багатокласові методи за допомогою моделей OneVsRest і OneVsOne.

Як показано в табл.3.4, яка представляє підсумок вимірювань продуктивності, витягнутих із кожного класифікаційного звіту із прикладних моделей. Ми виявили, що багатокласні методи досягли найвищої продуктивності, OneVSRest досяг точності 98,28%, запам'ятовування (98,21%), показник F1 (98,24%) і точність 98,20%.

Таблиця 3.4 - Результати застосованих методів навчання під наглядом, перевірених за допомогою DB-MALCURL

Модель	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Точність
OneVsRest	0.982861	0.982111	0.982456	0.982016
OneVsOne	0.981371	0.980555	0.980926	0.980518
KNN	0.964221	0.965574	0.964551	0.964033
Decision Tree	0.955805	0.956841	0.956249	0.955586

Модель OneVsOne досягла точності (98,13%), запам'ятовування (98,05%), показника F1 (98,09%) і точності 98,05%. Крім того, ми помітили, що модель KNN досягла точності 96,42%, запам'ятовування (96,55%), оцінки F1 (96,45%) і точності 96,40%. Однак модель дерева рішень зафіксувала точність (95,58%), запам'ятовування (95,68%), оцінку F1 (95,62%) і точність 95,55%.

Крім того, табл.3.5 відображає виявлення для класу, а також коефіцієнт вимірювання з використанням моделей навчання під наглядом.

Таблиця 3.5 - Результати виявлення кожного класу за допомогою коефіцієнта вимірювання за допомогою моделей навчання під наглядом

Model	**Nb. Instance**	**Coefficient**	**One Vs Rest**		**One Vs One**		**KNN**		**Decision Tree**	
			Rate	**Detect**	**Rate**	**Detect**	**Rate**	**Detect**	**Rate**	**Detect**
Defacement	1547	Precision	98.96%	**1527**	98.96%	1526	95.50%	1527	95.50%	1502
		Recall	98.71%		98.71%		Recall		98.64%	
		F1-score	98.84%		97.08%		F1-score		98.80%	
Benign	1506	Precision	97.83%	1485	97.76%	**1487**	96.96%	1469	96.96%	1454
		Recall	98.61%		97.54%		Recall		98.74%	
		F1-score	98.21%		97.25%		F1-score		98.25%	
Malware	1334	Precision	99.39%	1302	99.23%	1297	96.13%	**1315**	96.13%	1292
		Recall	97.60%		98.58%		Recall		97.23%	
		F1-score	98.49%		97.34%		F1-score		98.22%	
Phishing	1589	Precision	95.84%	**1544**	95.53%	1540	96.27%	1421	96.27%	1431
		Recall	97.17%		89.43%		Recall		96.92%	
		F1-score	96.50%		92.72%		F1-score		96.22%	
Spam	1364	Precision	99.41%	**1350**	99.19%	1347	97.25%	1344	97.25%	1335
		Recall	98.97%		98.53%		Recall		98.75%	
		F1-score	99.19%		97.89%		F1-score		98.97%	

Проаналізувавши табл.3.5, виявлено, що одна проти інших досягла найточнішої моделі, а також найкращого класифікатора для виявлення класів пошкодження, фішингу та спаму. Результати були отримані з матриці плутанини, показаної на рис.3.4. Отже, серед 1547 записів, класифікованих як пошкодження,

ми виявили 1527 пошкодження, 1 доброякісне, 0 зловмисних програм, 17 фішингових і 2 спаму з рівнем точності 98,96%, відкликання (98,71%) і показником f1 (98,84%). Крім того, серед 1506 записів, класифікованих як доброякісні, ми отримали 2 пошкодження, 1485 доброякісних, 4 зловмисних програм, 15 фішингових і 0 спамів із рівнем точності 97,83%, відкликання (98,61%) і показником f1 (98,21%). Серед 1334 записів, класифікованих як зловмисне програмне забезпечення, ми виявили 0 пошкоджень, 6 доброякісних програм, 1302 зловмисних програм, 25 фішингових програм і 1 спам із коефіцієнтом точності (99,39%), відкликанням (97,60%) і показником f1 (98,49%), а серед 1589 записів ми класифікували як фішинг, ми отримали 10 пошкоджень, 26 доброякісних, 4 зловмисних програм, 1544 фішингові та 5 спамів із рівнем точності 95,84%, відкликання (97,17%) та показником f1 (96,50%). Нарешті, серед 1364, пов’язаних із спам-атакою, ми досягли 4 пошкоджень, 0 доброякісних, 0 зловмисних програм, 10 фішингових і 1350 спамів із вимірюванням високого коефіцієнта; коефіцієнт точності 99,41%, запам'ятовування (98,97%) і показник f1 (99,19%).

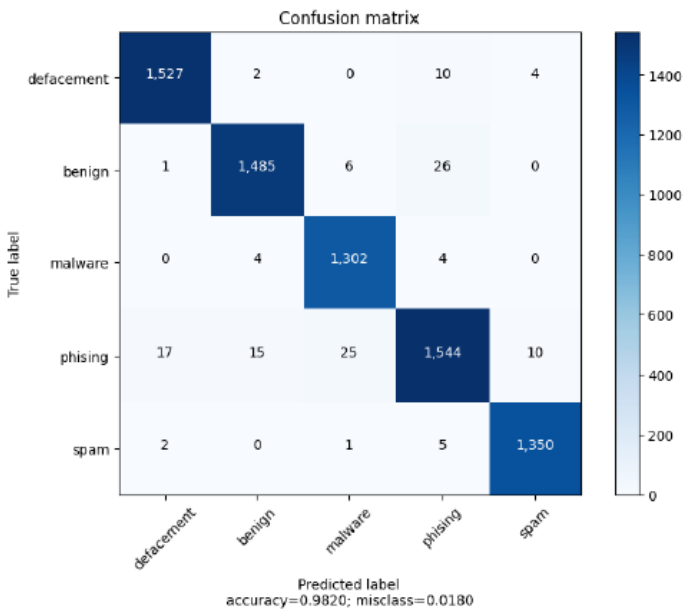


Рисунок 3.4 - Матриця помилок для моделі OVR на DB-MALCURL

Крім того, один проти одного було класифіковано як найкращий класифікатор у виявленні доброякісного класу. Отримані результати були

витягнуті з матриці плутанини, показаної на рис.3.5. Таким чином, серед 1506 записів, класифікованих як доброякісні, ми виявили 1 пошкодження, 1487 доброякісних, 4 зловмисних програм, 13 фішингових і 1 спам із точністю 97,76%, відкликання (98,74%) і показник f1 (98,25%). Крім того, серед 1547 записів, класифікованих як пошкодження, ми отримали 1526 пошкоджень, 1 нешкідливий, 0 шкідливих програм, 18 фішингових і 2 спаму.

Серед 1334 записів, класифікованих як зловмисне програмне забезпечення, виявлено 0 пошкоджень, 8 доброякісних, 1297 шкідливих програм, 28 фішингових і 1 спам. Однак серед 1589 записів, класифікованих як фішинг, отримали 11 пошкоджень, 25 нешкідливих, 6 шкідливих програм, 1540 фішингових і 7 спаму. Нарешті, серед 1364, пов'язаних зі спам-атакою, система зафіксувала 4 пошкодження, 0 доброякісних, 0 шкідливих програм, 13 фішингових і 1347 спамів.

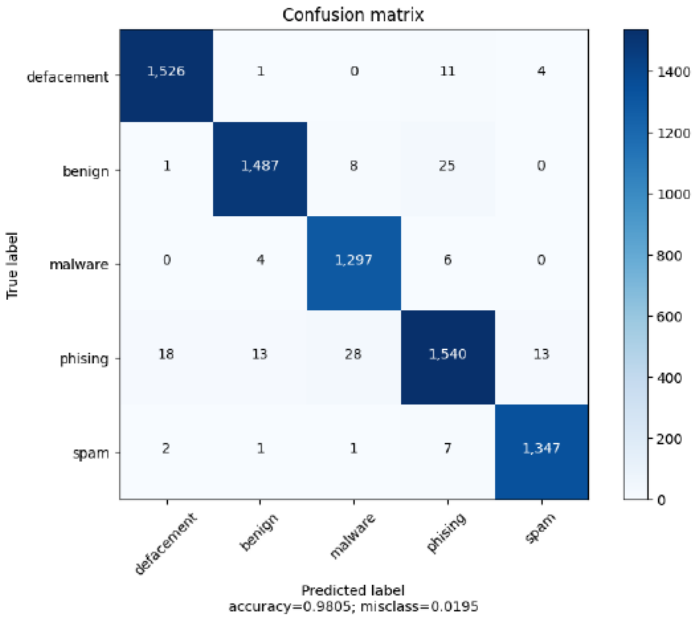


Рисунок 3.5 - Матриця помилок для моделі OVO на DB-MALCURL

KNN було класифіковано як найкращий класифікатор у виявленні класу шкідливих програм. Отримані результати були витягнуті з матриці плутанини, показаної на рис.3.6. Серед 1334 записів, класифікованих як зловмисне програмне забезпечення, ми виявили 0 пошкоджень, 5 доброякісних, 1315 зловмисних програм, 13 фішингових і 1 спам із показником точності 96,13%, відкликання

(98,58%) і показником f1 (97,34%). Крім того, серед 1547 записів, класифікованих як пошкодження, ми отримали 1527 пошкоджень, 1 нешкідливий, 2 зловмисних програм, 12 фішингових і 5 спамових. Серед 1506 записів, класифікованих як безпечні, ми досягли 8 пошкоджень, 1469 доброякісних програм, 11 шкідливих програм, 17 фішингових і 1 спам, а серед 1589 записів, класифікованих як фішинг, ми досягли 58 пошкоджень, 40 доброякісних, 39 шкідливих програм, 1421 фішинг і 31 спам. Нарешті, серед 1364, пов'язаних зі спам-атакою, ми отримали 6 пошкоджень, 0 доброякісних, 1 зловмисне програмне забезпечення, 13 фішингових і 1344 спаму.

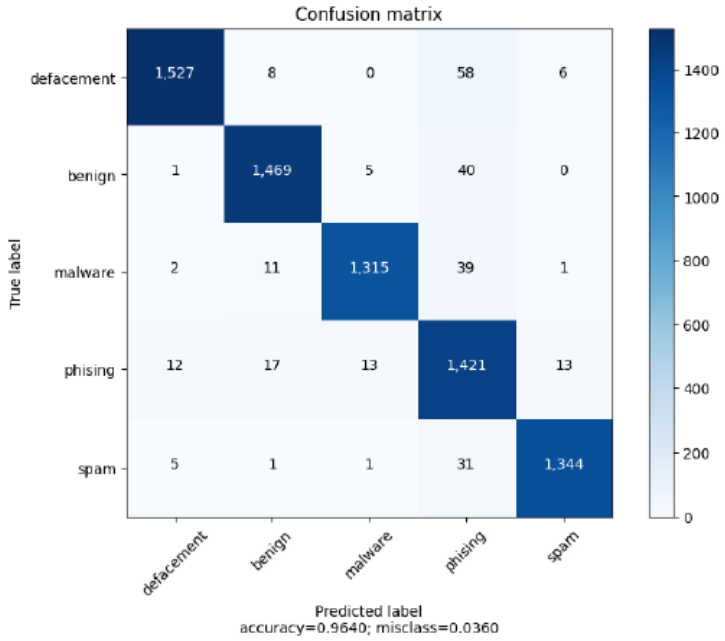


Рисунок 3.6 - Матриця плутанини для моделі KNN на DB-MALCURL

Результати класифікатора дерева рішень показали найнижчу продуктивність у виявленні всіх класів. Як показано в матриці помилок на рис.3.7, серед 1547 записів, класифікованих як пошкодження, ми отримали 1502 пошкодження, 3 доброякісних, 7 шкідливих програм, 28 фішингових і 7 спамових. Крім того, серед 1506 записів, класифікованих як доброякісні, ми отримали 9 пошкоджень, 1454 доброякісних, 10 шкідливих програм, 32 фішингові та 1 спам.

Крім того, серед 1334 записів, класифікованих як зловмисне програмне забезпечення, ми отримали 1 пошкодження, 4 доброякісних, 1292 зловмисних

програм, 35 фішингових і 2 спаму, тоді як серед 1589 записів, класифікованих як фішинг, ми отримали 44 пошкодження, 36 доброякісних, 47 шкідливих програм, 1431 фішинг і 31 спам. Нарешті, серед 1364 випадків, пов'язаних зі спам-атакою, ми отримали 7 пошкоджень, 1 нешкідливий, 6 зловмисних програм, 15 фішингових і 1335 спам.

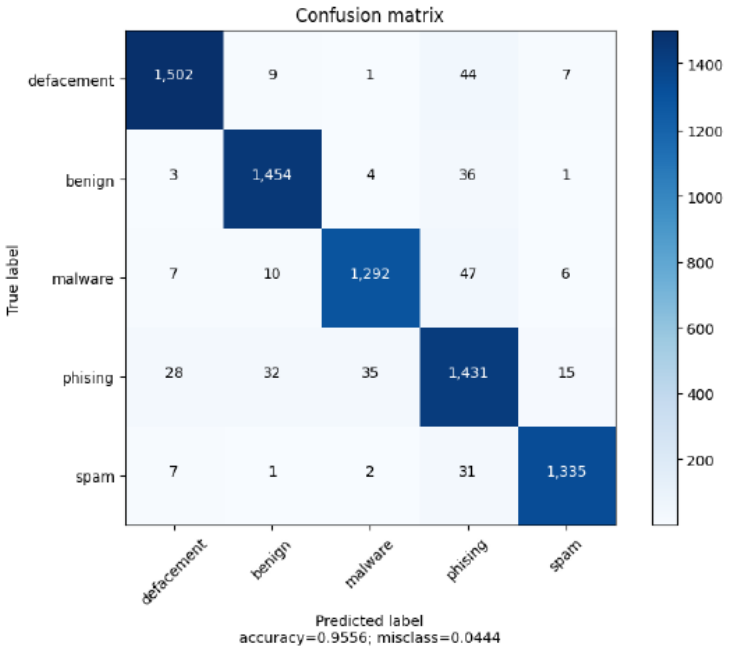


Рисунок 3.7 - Матриця помилок для моделі дерева рішень на DB-MALCURL

Застосування системи SIS-ID на DB-DDOS. Далі будуть наведені результати, отримані досліджуваною системою SIS-ID, яка була використана в DB-DDOS з використанням контрольованого навчання.

Контрольоване навчання. Моделі були перевірені з використанням кількох контрольованих алгоритмів навчання, які: дерево рішень, найближчі сусіди (KNN) і багатокласові методи за допомогою моделей OneVsRest і OneVsOne. Як показано в табл.3.6, яка представляє підсумок вимірювань продуктивності, отриманих із кожного класифікаційного звіту цих моделей. Ми виявили, що багатокласні методи досягли найвищої ефективності, OneVSRest досяг точності 79,60%, запам'ятовування (76,85%), показник F1 (76,03%) і точність 76,82%, тоді як модель OneVsOne зафіксувала точність (79,43%)), запам'ятовування (76,81%), F1-оцінка (76,02%) і точність 76,78 %. Крім того, ми помітили, що модель «Дерево рішень»

досягла точності (79,29%), запам’ятовування (76,74%), оцінки F1 (75,97%) і точності 76,71%, і, нарешті, модель KNN досягла точності (73,51%), запам’ятовування (72,52%), F1-бал (71,98%) і точність 72,48 %.

Таблиця 3.6 - Результати застосованих методів навчання під наглядом, перевірених через DB-DDOS

Модель	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Точність
OneVsRest	0.7961	0.7685	0.7604	0.7682
OneVsOne	0.7944	0.7682	0.7603	0.7679
Decision Tree	0.793	0.7675	0.7597	0.7672
KNN	0.7352	0.7252	0.7199	0.7248

Таблиця 3.7 відображає етап виявлення для кожного класу, а також вимірювання коефіцієнтів за допомогою моделей навчання під наглядом:

Таблиця 3.7 - Результати виявлення кожного класу за допомогою моделей навчання під наглядом, які перевірялися через DB-DDOS

Attack	**Nb. Instance**	**Coefficient**	**One Vs Rest**		**One Vs One**		**Decision Tree**		**KNN**	
			Rate	**Detect**	**Rate**	**Detect**	**Rate**	**Detect**	**Rate**	**Detect**
BENIGN	11133	Precision	99.54%	**10998**	99.56%	10995	99.49%	10995	93.65%	10337
		Recall	98.79%		98.67%		98.79%		98.67%	
		F1-score	99.16%		99.08%		99.16%		99.08%	
DrDoS_ DNS	11195	Precision	81.99%	5493	81.85%	5500	79.97%	5479	47.37%	**8214**
		Recall	49.07%		48.94%		Recall		49.13%	
		F1-score	61.39%		60.72%		F1-score		61.40%	
DrDoS_ LDAP	11175	Precision	52.02%	8355	52.02%	**8362**	51.94%	8285	50.86%	4763
		Recall	74.77%		74.14%		Recall		74.83%	
		F1-score	61.35%		61.09%		F1-score		61.37%	
DrDoS_ MSSQL	11210	Precision	71.39%	4905	71.58%	4923	71.90%	4928	58.26%	**5011**
		Recall	43.76%		43.96%		Recall		43.92%	
		F1-score	54.26%		54.56%		F1-score		54.43%	
DrDoS_ NetBIOS	11127	Precision	97.52%	10550	97.57%	10551	97.28%	**10549**	95.57%	10185
		Recall	94.81%		94.81%		Recall		94.82%	
		F1-score	96.15%		96.03%		F1-score		96.18%	
DrDoS_ NTP	11265	Precision	79.04%	8076	78.86%	8071	79.15%	**8091**	76.50%	7745
		Recall	71.69%		71.82%		Recall		71.65%	
		F1-score	75.19%		75.31%		F1-score		75.08%	
DrDoS_ SNMP	11262	Precision	89.72%	**11156**	89.73%	11155	89.68%	11151	89.67%	11114
		Recall	99.06%		99.01%		Recall		99.05%	
		F1-score	94.16%		94.12%		F1-score		94.16%	
DrDoS_ SSDP	11325	Precision	61.65%	**9072**	62.55%	8759	61.76%	8828	68.44%	6435
		Recall	80.11%		77.95%		Recall		77.34%	
		F1-score	69.68%		68.92%		F1-score		69.16%	
DrDoS_ UDP	11186	Precision	70.04%	10085	69.92%	10078	70.52%	**11186**	68.84%	9209
		Recall	90.16%		89.04%		Recall		90.09%	
		F1-score	78.84%		78.71%		F1-score		78.73%	

Продовження таблиці 3.7 - Результати виявлення кожного класу за допомогою моделей навчання під наглядом, які перевірялися через DB-DDOS

Attack	**Nb. Instance**	**Coefficient**	**One Vs Rest**		**One Vs One**		**Decision Tree**		**KNN**	
			Rate	**Detect**	**Rate**	**Detect**	**Rate**	**Detect**	**Rate**	**Detect**
Syn	11182	Precision	96.05%	4714	95.83%	4709	95.52%	4729	79.44%	**5158**
		Recall	42.16%		42.29%		Recall		42.11%	
		F1-score	58.60%		58.63%		F1-score		58.51%	
TFTP	11178	Precision	61.98%	**10953**	61.95%	10942	61.99%	10928	61.90%	10076
		Recall	97.99%		97.76%		Recall		97.89%	
		F1-score	75.93%		75.87%		F1-score		75.88%	
UDP-lag	11244	Precision	77.13%	6400	74.31%	6657	74.63%	**6702**	68.96%	6255
		Recall	56.92%		59.61%		Recall		59.20%	
		F1-score	65.50%		66.28%		F1-score		65.90%	
WebDDoS	11118	Precision	96.85%	11099	96.95%	**11101**	97.04%	11088	96.28%	11036
		Recall	99.83%		99.73%		Recall		99.85%	
		F1-score	98.32%		98.37%		F1-score		98.38%	

Проаналізувавши табл.3.7, видно, що один проти решти є найкращим класифікатором для визначення класів BENIGN, DrDoS_SNMP, DrDoS_SSDP і TFTP. Результати були отримані з матриці плутанини, показаної на рис.3.8. Таким чином, серед 11133 записів, класифікованих як доброякісні, ми виявили 10998 доброякісних і 135 випадків для решти класів атак як хибнопозитивні з рівнем точності 99,54%, запам’ятовування (98,79%) і показник f1 (99,16%). Крім того, серед 11262 записів, класифікованих як DrDoS_SNMP, ми виявили 11156 DrDoS_SNMP і 106 випадків для решти класів як хибнопозитивні з рівнем точності 89,72%, запам’ятовування (99,06%) і показник f1 (94,16%). Крім того, серед 11325 записів, класифікованих як DrDoS_SSDP, ми помітили 9072 DrDoS_SSDP і 2253 випадки для решти класів як хибнопозитивні з рівнем точності 61,65%, запам’ятовування (80,11%) і f1-бал (69,68%). Нарешті, серед 11178 записів, класифікованих як TFTP, ми знайшли 10953 TFTP і 225 випадків для решти класів як хибнопозитивні з рівнем точності 61,98%, запам’ятовування (97,99%) і f1-оцінки (75,93%).

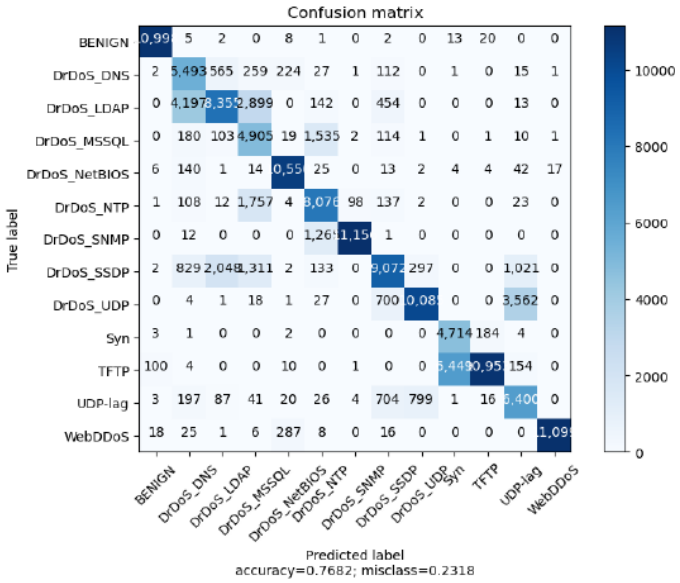


Рисунок 3.8 - Матриця помилок для моделі OVR на DB-DDOS

Один проти одного є найкращим класифікатором у виявленні класів DrDoS_LDAP і WebDDoS. Результати були отримані з матриці плутанини, показаної на рис.3.9. Таким чином, серед 11175 записів, класифікованих як DrDoS_LDAP, ми виявили 8362 DrDoS_LDAP і 2813 випадків для решти класів як хибнопозитивні з рівнем точності 52,02%, запам’ятовування (74,83%) і оцінкою f1 (61,37%). Нарешті, серед 11118 записів, класифікованих як WebDDoS, ми виявили 11101 WebDDoS і 17 випадків для решти класів як хибнопозитивні з рівнем точності 96,95%, запам’ятовування (99,85%) і f1-бал (98,38%).

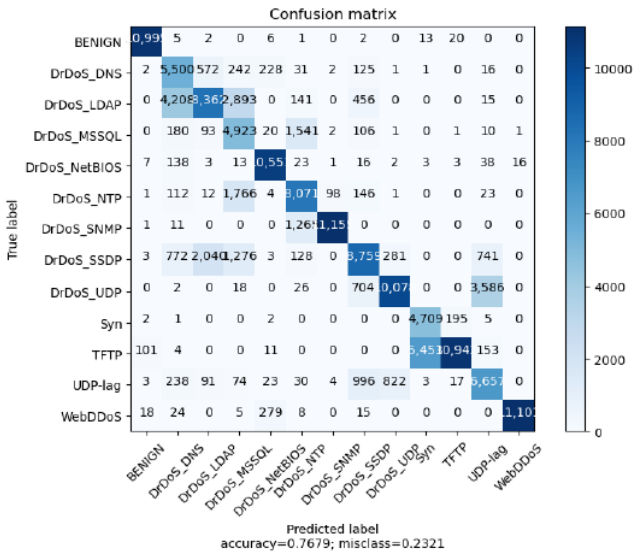


Рис.3.9 - Матриця помилок для моделі OVO в DB-DDOS

Крім того, модель дерева рішень є найкращою для виявлення класів DrDoS_NetBIOS, DrDoS_NTP, DrDoS_UDP і UDP-lag. Отримані результати були витягнуті з матриці плутанини, показаної на рис.3.10. Таким чином, серед 11127 записів, класифікованих як DrDoS_NetBIOS, ми отримали 10549 DrDoS_NetBIOS і 578 випадків для решти класів як хибнопозитивні з рівнем точності 97,28%, запам'ятовування (94,81%) і оцінкою f1 (96,03%). Крім того, серед 11265 записів, класифікованих як DrDoS_NTP, ми отримали 8091 DrDoS_NTP і 3174 випадки для решти класів як хибнопозитивні з рівнем точності 79,15%, запам'ятовування (71,82%) і f1-бал (75,31%). Крім того, серед 11186 записів, класифікованих як DrDoS_UDP, ми отримали 9960 DrDoS_UDP і 1226 випадків для решти класів як хибнопозитивні з рівнем точності 70,52%, запам'ятовування (89,04%) і показник f1 (78,71%). Нарешті, серед 11244 записів, класифікованих як UDP-лаг, ми отримали 6702 UDP-лаг і 4542 випадки для решти класів як хибнопозитивні з рівнем точності 74,63%, запам'ятовування (59,61%) і f1-бал (66,28%).

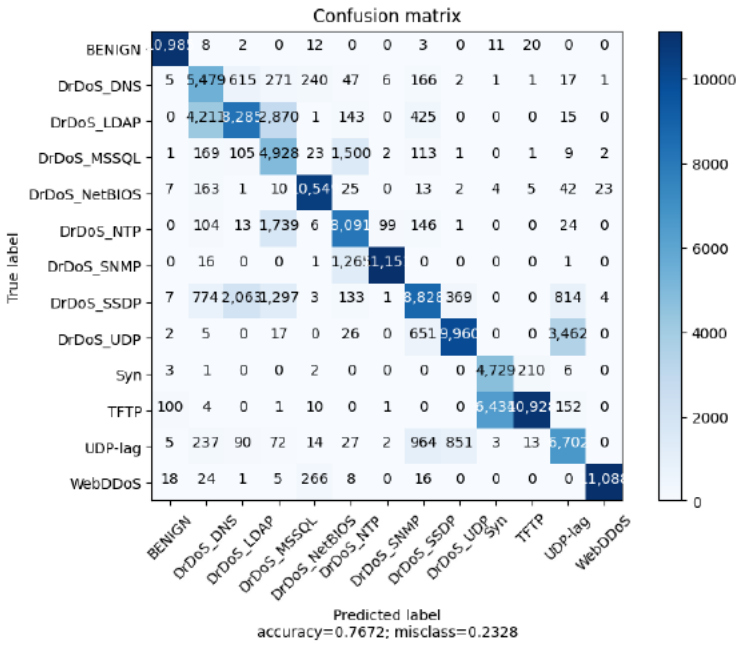


Рисунок 3.10 - Матриця помилок для моделі дерева рішень у DB-DDOS

Зрештою, модель KNN заслужила бути найкращим класифікатором у ідентифікації класів DrDoS_DNS, DrDoS_MSSQL і Syn. Отримані результати були витягнуті з матриці плутанини, показаної на рис.3.11. Таким чином, серед 11195

записів, класифікованих як DrDoS_DNS, ми отримали 8214 DrDoS_DNS і 2981 екземпляр для решти класів як хибнопозитивні з показником точності 47,37%, запам'ятовування (73,37%) і оцінкою f1 (57,57%). Крім того, серед 11210 записів, класифікованих як DrDoS_MSSQL, ми отримали 5011 DrDoS_MSSQL і виявили 6199 випадків для решти класів як хибнопозитивні з показником точності (58,26%), запам'ятовування (44,70%) і оцінкою f1 (50,59%). Нарешті, серед 11182 записів, класифікованих як Syn, ми отримали 5158 Syn і 6024 випадки для решти класів як хибнопозитивні з рівнем точності (79,44%), запам'ятовування (46,13%) і показником f1 (58,36%).

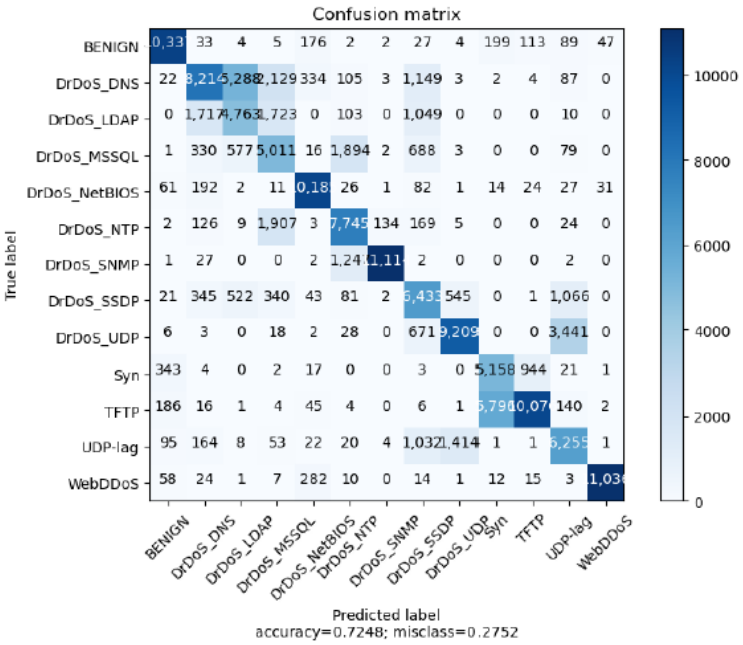


Рис.3.11 -Матриця плутанини для моделі KNN на DB-DDOS

Навчання без вчителя. Далі наведено результати моделі факторів локальних викидів, яка була розгорнута в дослідженому апаратному забезпеченні для виявлення невідомих прихідних трафіків (виявлення новизни). Як показано на рис.3.12, який представляє визначення продуктивності на етапі тестування з використанням DB-DDOS, результати показують, що серед 40000 записів, класифікованих як наступні атаки (викиди), ми виявили 38778 як атаки та 1222 як доброякісні (внутрішні). Крім того, модель досягла ефективного рівня виявлення в 96,94%.

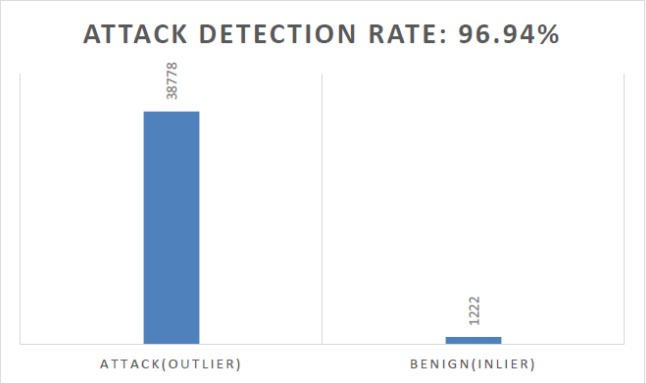


Рисунок 3.12 - Вимірювання продуктивності моделі LOF, яка розгорнута в пропонованому обладнанні

Загальне обговорення та оцінка результатів. Досліджена інтелектуальна система безпеки показує високі показники виявлення при прогнозуванні як шкідливих URL-адрес, так і атак DDOS на основі десяти моделей машинного навчання. Дійсно, щоб підтвердити наш підхід і подолати обмеження у виявленні шкідливих URL-атак, ми представляємо продуктивність нашої системи на етапі тестування в табл.3.8 та 3.9. Ці таблицях відображають найкращі моделі для виявлення кожної атаки з досягнутим рівнем виявлення. Таким чином, проаналізувавши ці таблиці, ми робимо висновок, що модель XGBosst досягла найкращої продуктивності у виявленні атаки дефейсингу у відсотках 99,41%, голосування за доброякісність 99,13%, KNN для зловмисного програмного забезпечення 98,57%, OneVsRest для фішингу 97,16% і, нарешті, стек для спаму з 99,41%.

Таблиця 3.8. - Результати моделей для виявлення кожної атаки за допомогою DB-MALCURL

Тип атаки	Кільк. екземп.	Voting	Stacking	XG-Boost	One Vs Rest	KNN
Defacement	1547	1535	1536	**1538**	1527	1527
Benign	1506	**1493**	1482	1489	1485	1469
Malware	1334	1312	1311	1313	1302	**1315**
Phishing	1589	1539	1541	1532	**1544**	1421
Spam	1364	1353	**1356**	1353	1350	1344

Таблиця 3.9 - Рівень виявлення, досягнутий найкращими моделями для кожної атаки за допомогою DB-MALCURL

Тип атаки	Модель	Виявлення екземплярів	Швидкість виявлення
Defacement	XGBoost	1538	99.41%
Benign	Voting	1493	99.13%
Malware	KNN	1315	98.57%
Phishing	OneVsRest	1544	97.16%
Spam	Stacking	1356	99.41%

Крім того, система довела покращення з гідними результатами в порівнянні з моделлю дерева рішень, яка зафіксувала найнижчий результат за допомогою навчання під наглядом, як показано в табл.3.10. Таким чином, ми розвинули продуктивність нашої системи, використовуючи голосування як пропоновану модель ансамблю, яка покращила на 3% точність, 2,86% (запам’ятовування), 2,93% (оцінка F1) і 2,97% (точність), а потім стек, який збільшився на 2,90% (точність), 2,78% (відкликання), 2,85% (оцінка F1) і 2,88% (точність).

Таблиця 3.10 - Результати різних методів ML, перевірених за допомогою DB-MALCURL

Модель	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Точність
Voting	0.98571	0.98552	0.9856	0.98529
Stacking	0.984885	0.98473	0.9848	0.98447
XGBoost	0.98465	0.98464	0.98463	0.98433
OneVsRest	0.982861	0.982111	0.982456	0.982016
Random Forest	0.982644	0.9818	0.98219	0.98174
Adaboost	0.981808	0.98101	0.98138	0.98093
OneVsOne	0.981371	0.980555	0.980926	0.980518
Bagging	0.979931	0.97903	0.97944	0.97902
KNN	0.964221	0.965574	0.964551	0.964033
Decision Tree	0.955805	0.956841	0.956249	0.955586

Крім того, після порівняння ефективності виявлення шкідливих URL-адрес у нашому SIS-ID з результатами, наданими лабораторією CIC [21], як показано в табл.3.11, очевидно, що наш підхід із моделлю KNN виявився кращим, точність і

перетнула понад 1,4%, зросла на 2,42% точності та (2,55%) для запам’ятовування, поки модель дерева рішень розвивалася; 0,55% точності, 1,58% точності та 1,68% запам'ятовування. Нарешті, випадковий ліс досяг кращої продуктивності, точність на 3,17%, точність на 1,26% і відкликання на 1,18%.

Таблиця 3.11 - Порівняльне дослідження SIS-ID, протестованого за допомогою DB-MALCURL і лабораторії CIC

	CIC Laboratory			SIS-ID		
Модель	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Random Forest	>0.95	0.97	0.97	0.98174	0.9826	0.9818
Decision Tree	>0.95	0.94	0.94	0.9555	0.9558	0.9568
KNN	>0.95	0.94	0.94	0.9640	0.9642	0.9655

З іншого боку, ми обговоримо вимірювання продуктивності нашої системи за допомогою DB-DDOS для виявлення атак DDOS. Проаналізувавши таблиці 3.12 та 3.12, які представляє найкращі моделі для виявлення кожного класу, ми робимо висновок, що модель стекування досягла найкращої продуктивності у виявленні п’яти зазначених класів відповідно; DrDoS_LDAP: 75,45%, DrDoS_NetBIOS: 95,46%, DrDoS_NTP: 72,98%, DrDoS_SNMP: 99,18% і DrDoS_UDP: 90,45%. Таким чином, ми підтвердили нашу методологію, застосувавши ансамблеві моделі. Крім того, модель OneVsRest досягла найкращих результатів у виявленні BENIGN: 98,79% і DrDoS_SSDP: 80,11%, тоді як модель KNN була кращою у виявленні DrDoS_DNS: 73,37% і Syn: 46,13%. Крім того, модель пакетування для TFTP: 98,41%, WebDDoS: 99,87% і Adaboost для DrDoS_MSSQL 44,87%. Нарешті, XG-Boost досягає 59,90% у виявленні атаки із затримкою UDP.

Таблиця 3.12 - Результати найкращих моделей для виявлення кожної атаки за допомогою DB-DDOS

Модель	Stacking	Baggin g	XG-Boost	One Vs Rest	Adaboost	KNN
BENIGN	10992	10997	10997	10998	10967	10337

Продовження таблиці 3.12 - Результати найкращих моделей для виявлення кожної атаки за допомогою DB-DDOS

Модель	Stacking	Baggin g	XG-Boost	One Vs Rest	Adaboost	KNN
DrDoS_DNS	5588	5469	5545	5493	5527	8214
DrDoS_LDAP	8431	8347	8357	8355	8328	4763
DrDoS_MSSQL	4847	4888	5015	4905	5030	5011
DrDoS_NetBIOS	10622	10536	10584	10550	10536	10185
DrDoS_NTP	8221	8180	8032	8076	7944	7745
DrDoS_SNMP	11170	11164	11162	11156	11150	11114
DrDoS_SSDP	8812	9042	8734	9072	8730	6435
DrDoS_UDP	10118	10095	10003	10085	10005	9209
Syn	4716	4676	4729	4714	4728	5158
TFTP	10912	11000	10924	10953	10963	10076
UDP-lag	6657	6424	6735	6400	6732	6255
WebDDoS	11094	11103	11095	11099	11093	11036

Таблиця 3.13 - Показники виявлення, досягнуті найкращими моделями для кожної атаки з використанням DB-DDOS

Attack Type	**Model**	**Detection of instances**	**Detection Rate**
BENIGN	OneVSRest	10998	98.79%
DrDoS_DNS	KNN	8214	73.37%
DrDoS_LDAP	Stacking	8431	75.45%
DrDoS_MSSQL	Adaboost	5030	44.87%
DrDoS_NetBIOS	Stacking	10622	95.46%
DrDoS_NTP	Stacking	8221	72.98%
DrDoS_SNMP	Stacking	11170	99.18%
DrDoS_SSDP	OneVSRest	9072	80.11%
DrDoS_UDP	Stacking	10118	90.45%
Syn	KNN	5158	46.13%
TFTP	Bagging	11000	98.41%
UDP-lag	XG-Boost	6735	59.90%
WebDDoS	Bagging	11103	99.87%

Система продемонструвала покращення вимірювань продуктивності з гідними результатами порівняно з KNN, який зафіксував найнижчий результат за допомогою навчання під керівництвом, як показано в табл.3.14. Тому систему розвинуто, використовуючи модель стекування, яку було вдосконалено наступним чином; 6,26% (точність), 4,55% (відкликання), 4,29% (оцінка F1) і 4,56% (точність).

Таблиця 3.14 - Результати різних методів ML, перевірених за допомогою DB-DDOS

Модель	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Точність
Stacking	0.79775	0.77077	0.7628	0.77047
Voting	0.79639	0.76899	0.76091	0.76869
Bagging	0.79762	0.76898	0.76072	0.76869
XGBoost	0.79455	0.76894	0.76135	0.76863
Random Forest	0.79649	0.76865	0.76051	0.76837
OneVsRest	0.79609	0.76853	0.7604	0.76824
OneVsOne	0.79436	0.76819	0.76027	0.76788
Adaboost	0.79343	0.76771	0.7603	0.7674
Decision Tree	0.79298	0.76749	0.75975	0.76719
KNN	0.73518	0.72524	0.71987	0.72484

Порівнюючи досліджувані підходи до виявлення атаки DDOS із продуктивністю IDS, пов’язаною з лабораторією CIC [21], наведеною в табл.3.15, чітко показано, що за допомогою моделі випадкового лісу досліджувана система показала кращі вимірювання; точність до 2,64%, запам'ятовування 11,87% і f1-Score (7,05%).

Покращення за допомогою моделі дерева рішень може бути показано відповідно; точність на 1,30%, запам'ятовування (11,75%) і F1-Score (6,97%).

Таблиця 3.15 - Порівняльне дослідження SIS-ID, протестованого через DB DDOS і лабораторію CIC

	CIC Laboratory			SIS-ID		
Модель	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Random Forest	0.77	0.65	0.69	0.796485	0.768653	0.760505
Decision Tree	0.78	0.65	0.69	0.792984	0.76749	0.759745

3.5 Апаратне моделювання в реальному часі

Далі наведено результати етапу перевірки досліджуваної системи SIS-ID, яка була перевірена за допомогою DB-DDOS на основі налаштованого обладнання на

етапі реального часу. Через проблематику системи запобігання вторгненням, пов’язану з труднощами роботи з невідомими майбутніми трафіками, які можуть стримувати кібератаки в мережі, застосовано в цьому експерименті мету впровадження механізму кібербезпеки, щоб уникнути будь-якої атаки, яка може загрожує серверу, як показано на рис.3.13.

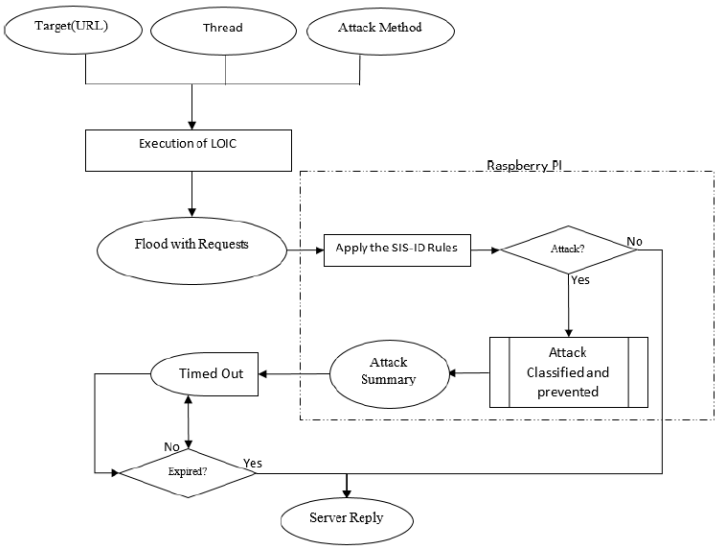


Рисунок 3.13 - Загальна архітектура апаратного моделювання в реальному часі SIS-ID

Перевірка. Проведено аналіз симуляції атаки та процес перевірки обладнання, щоб уникнути атаки відмови в обслуговуванні на етапі реального часу. Як показано на рис.3.13, програмне забезпечення LOIC було взято для виконання атаки на відмову в обслуговуванні (DOS), а потім для перевірки на основі raspberrypi як інтелектуального апаратного забезпечення безпеки з використанням моделі локального викиду. Отже, жертвою ми обрали доменне ім’я. HTTP-запит — це метод атаки, для імітації цієї атаки було п’ять потоків. Після цього здійснено невелику DOS-атаку. З іншого боку, апаратне забезпечення було налаштоване для захоплення пакетів, що надходять, за допомогою CICFLOWMETER, щоб отримати відповідні функції, пов’язані з потоками. Таким чином, ці пакети були створені та узгоджені з нашою системою навчання SIS-ID відповідно до її правил. Крім того, як показано на рис.3.14, апаратне забезпечення довело ефективність у виявленні

наступної атаки, а також у запобіганні їй. Спостережувана атака тривала 60 секунд з 12:40:08 до 12:41:08 і засипала жертву кількома незвичайними запитами.

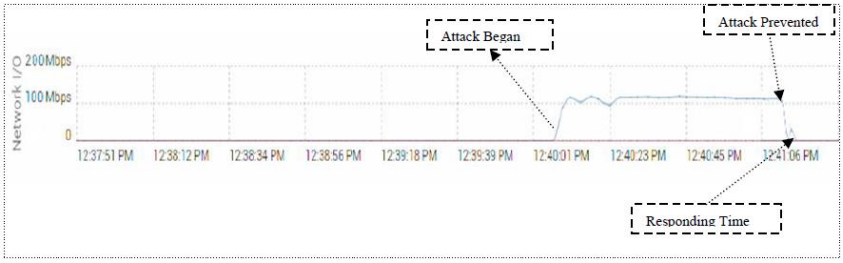


Рисунок 3.14 - Ефективність виявлення наступної атаки на етапі реального часу

Після цього, як показано на рис.3.15 представлено деякі результати нашого raspberry pi за цей період. Крім того, було досягнуто запобігання, використовуючи налаштований брандмауер у нашому обладнанні, щоб уникнути виявлених атак. Таким чином, наша розгорнута система зафіксувала всі потоки, які містять кілька аномальних пакетів, із можливістю ідентифікації IP-адрес, пов'язаних із джерелом загрози. Таким чином, у цьому експерименті інтелектуальне апаратне забезпечення безпеки запобігло п'яти потокам наслідків, які надходили з IP-адреси «10.3.141.106» на веб-сервер, що підтвердило динамічний захист за правилами. Зрештою, атаку було ідентифіковано та заблоковано, а також затримано її на 5 секунд, оскільки запропонований запит минув, щоб вказати, що сервер не отримав жодного незвичайного запиту протягом визначеного періоду часу.

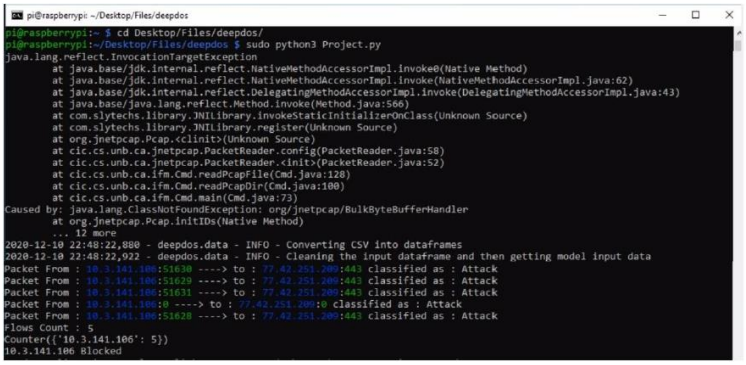


Рис.3.15 - Результат роботи нашого апаратного забезпечення для уникнення майбутньої атаки DOS

ВИСНОВКИ

Виконуючи поставлені завдання у роботі проаналізовано основні поняття кібербезпеки. Виконано огляд термінів та наведено їх класифікацію. Досліджено основні технології кібербезпеки. Для розуміння небезпеки яку несуть кібернетичні атаки і їх результат у першому розділі приведено приклад маштабних кібератак. Визначено та представлено найпоширеніші кібератаки

Виконано дослідження особливостей застосування штучного інтелекту в кібербезпеці, проаналізоввно прикладні методи машинного навчання.

Досліджено функціонування системи виявлення вторгнень на основі хосту (HIDS), використовуючи техніку аналізу тексту. наведено результати експериментальних досліджень застосування методів машинного навчання в кібербезпеці. Виконуючи аналіз було використано чотири різні методи машинного навчання (KNN, SVM, Decision Tree і MLP), щоб обрати найкращу ефективну модель класифікації. HIDS досяг здатності виявляти SQLi, XSS і атаки обходу каталогу. Таким чином, MLP зафіксував найкращу точність у 90,67 %. Згодом KNN отримав другий показник точності з 88,17 %, а потім дерево рішень — 86,08 %. Нарешті, SVM отримав найнижчий рівень точності 82,67%.

У третьому розділі досліджено виявлення вторгнень на основі інтелектуальної системи безпеки за допомогою методів машинного навчання, яка призначена для виявлення останніх зловмисних URL-адрес і розширену для атак розподіленої відмови в обслуговуванні (DDoS). Проведено експерементальні дослідження та оцінка продуктивності досліджуваної системи SIS-ID.

Крім того, було обговорено кілька вимірювань щодо досягнутої швидкості виявлення для кожної моделі з точки зору кожної розглянутої атаки з використанням SIS-ID. Досліджувана система SIS-ID бла перевірений як апаратне забезпечення для запобігання вторгненню з ефективністю запобігання атакам відмови в обслуговуванні (DOS) на етапі реального часу.

ПЕРЕЛІК ПОСИЛАНЬ

1. R. Toshniwal, K. G. Dastidar, A. Nath . “Big Data Security Issues and Challenges”, International Journal of Innovative Research in Advanced Engineering (IJIRAE), 2020, ISSN: 2349-2163 Issue 2, Volume 2.

2. J. Jang-Jaccard, S .Nepal. “A Survey of Emerging Threats in Cybersecurity”, Journal of Computer and System Sciences, 2018. pp. 973-993, Issue 5, Volume 80.

3. A. Saravanan and S. S. Bama. “A Review on Cyber Security and the Fifth Generation Cyberattacks”, Oriental Journal of Computer Science and Technology, ISSN: 0974-6471, No. (2), 2019. pp. 50- 56, Volume 12.

4. M. M. Alhassana, A. Adjei-Quayeb.. “Information Security in an Organization”, International Journal of Computer (IJC), 2017. ISSN 2307-4523.

5. R. S. Limaye. “The importance of Information Integrity, Security, Networking and Data Protection”, International Journal of Innovations in Engineering and Technology (IJIET), ISSN: 2319-1058, 2018. Issue 3, Volume 2.

6. W. R. Cheswick, S. M. Bellovin and A. D. Rubin. “Firewalls and Internet Security: Repelling the Wilyhacke”. Addison-Wesley Longman Publishing Co., Inc.2018.

7. S. Jingyao, S. Chandel, Y. Yunnan, Z. Jingji, Z. Zhipeng. (2020). “Securing a Network: How Effective Using Firewalls and VPNs. In: Arai K., Bhatia R. (eds) Advances in Information and Communication. FICC 2019. Lecture Notes in Networks and Systems. Springer, Cham, Volume 70.

8. S. Jingyao, S. Chandel, Y. Yunnan, Z. Jingji, Z. Zhipeng. “Securing a Network: How Effective Using Firewalls and VPNs Are?”. In: Arai K., Bhatia R. (eds) Advances in Information and Communication. FICC 2019. Lecture Notes in Networks and Systems. Springer, Cham, Volume 70.

9. A. Salah, M. Shouman, and H. Faheem. “Surviving Cyber Warfare with A Hybrid Multiagent- Base Intrusion Prevention System”, Potentials, IEEE, no. 1, 2020. pp. 32–40, Volume 29.

10. MMH. Alansari, ZM. Aljazzaf, M. Sarfraz. “On Cyber Crimes and Cyber Security”. In M. Sarfraz (Ed.), Developments in Information Security and Cybernetic Wars, 2018., pp. 1-41.

11. A. Chauhan, G. Mishra, G. Kumar. . “Survey on Data Mining Techniques in Intrusion Detection”, International Journal of Scientific & Engineering Research 2(7) 2019.

12. A. Navada, A. N. Ansari, S. Patil, B. A. Sonkamble. "Overview of Use of Decision Tree Algorithms In Machine Learning", 2018 IEEE Control and System Graduate Research Colloquium, Shah Alam, pp. 37-42, doi: 10.1109/ICSGRC.2011.5991826.

13. S. Wan, H. Yang. "Comparison among Methods of Ensemble Learning", 2013 International Symposium on Biometrics and Security Technologies, Chengdu, 2017. pp. 286-290.

14. Y. Bouzida, F. Cuppens. “Neural Networks Vs. Decision Trees for Intrusion Detection”, in IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM),2016. pp. 81–88.

15. S. McElwee, J. Heaton, J. Fraley, J. Cannady. “Deep Learning for Prioritizing and Responding to Intrusion Detection Alerts”. In Proceedings of the MILCOM 2017—2017 IEEE Military Communications Conference (MILCOM), Baltimore,2017 MD, USA, 23–25, pp. 1–5.

16. J. Fonseca, M. Vieira, H. Madeira. “Evaluation of Web Security Mechanisms using Vulnerability & Attack Injection”, IEEE Transactions on Dependable & Secure Computing, 2018. 440–453, Volume 11.

17. S. Singh, S Silakari. “An Ensemble Approach for Cyber Attack Detection System: A Generic Framework”, 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2018. 79–84.

18. Q. Le, T. Mikolov. “Distributed Representations of Sentences and Documents”, The 31st International Conference on International Conference on Machine Learning,2019. Volume 32, 88–96.

19. MSI. Mamun, MA Rathore, AH. Lashkari, N. Stakhanova, AA. Ghorbani. 2016. “Detecting Malicious URLs Using Lexical Analysis”, Network and System Security. doi:10.1007/978-3-319-46298-1_30.

20. Sharafaldin, AH. Lashkari, S. Hakak, AA. Ghorbani. (2019). “Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy”, 2019 International Carnahan Conference on Security Technology (ICCST). doi:10.1109/ccst.2019.8888419.

21. AH. Lashkari, G. Draper-Gil, MSI Mamun, AA. Ghorbani. "Characterization of Tor Traffic Using Time Based Features", In the proceeding of the 3rd International Conference on Information System Security and Privacy, SCITEPRESS, Porto, Portugal. 2017.

22. Pedregosa. (2011). “Scikit-learn: Machine Learning in Python”,2021. JMLR 12, pp. 2825-2830.

23. R. Mohammed, J. Rawashdeh, M. Abdullah. (2020). "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results", 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, pp. 243-248, doi: 10.1109/ICICS49469.2020.239556.

24. S. Mani, A. Ozdas, C. Aliferis, HA.Varol, Q. Chen, R. Carnevale.“Medical Decision Support Using Machine Learning for Early Detection of Late-Onset Neonatal Sepsis”, J Am Med Informatics Assoc. 2017. 21(2):326–36.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра Комп'ютерної інженерії

ПРЕЗЕНТАЦІЯ ДО МАГІСТЕРСЬКОЇ РОБОТИ

на тему:

«ПРОТИДІЯ КІБЕРЗАГРОЗАМ В УМОВАХ СТРІМКОГО РОЗВИТКУ ШТУЧНОГО ІНТЕЛЕКТУ»

Студент: Поліщук А.Р.
Керівник: Антоненко А.В.

Київ 2023р.

Мета та завдання

Слайд 2

Метою даної роботи є дослідження шляхів покращення механізму безпеки із застосуванням методів машинного навчання для виявлення кібер-атак.

Об'єкт дослідження - процес виявлення кібер-атак.

Предмет дослідження – методи машинного навчання для виявлення кібер-вторгнень.

Для виконання поставленої мети, у магістерській роботі розроблено та виконано наступні завдання:

- огляд основних понять кібербезпеки та особливості застосування штучного інтелекту в кібербезпеці;
- дослідження систем виявлення вторгнень на основі хоста аналізу тексту та машинного навчання;
- експерементальні дослідження виявлення вторгнень на основі інтелектуальної системи безпеки.

Аналіз основних понять, термінів, технологій кібербезпеки та їх класифікація

Слайд 3

Таблиця 1- Найважливіші інструменти відповідно до тріади CIA

Трійка CIA	Інструмент	Мета
Конфіден-ційність	Шифрування та дешифрування	Захист конфіденційних і цінних даних, таких як банківські номери кредитів і транзакцій електронної комерції.
	Управління доступом	Визначення політик доступу до систем, фізичних компонентів і ресурси віртуальних мереж, надаючи користувачам привілеї доступу та дозволи.
	Аутентифікація	Надання дозволу на підтвердження ідентифікації користувача для будь-якого процесу автентифікації.
	Авторизація	Надання користувачеві дозволу щодо його/її поведінки пов'язані з обмеженим механізмом безпеки для авторизованого доступу до системних ресурсів за допомогою попередньо визначених політик.
Цілісність	Резервні копії	Зберігання даних періодично або автоматично в системі керування базами даних.
	Контрольні суми	Забезпечення цілісності переданих даних між мережами за допомогою математичної функції для відображення вмісту файлу в числовому значенні.
	Корекція даних	Виявлення неочікуваних змін шляхом збереження автентичних даних і ідентифікації неоригінальних між собою.
Доступність	Фізичні засоби захисту	Захист компонентів інфраструктури для передачі своїх ресурсів, які зберігаються в безпечній зоні, щоб зберігати доступні дані.

1. Брандмауер і VPN. Створюють міцний бар'єр між мережевими компонентами для забезпечення послуг захисту.

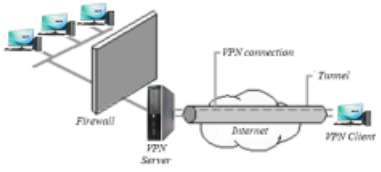


Рисунок 1- Брандмауер і архітектура VPN у мережі

2. Система виявлення вторгнень (IDS). Інструментом, який ідентифікує зловмисні та шкідливі дії в комп'ютерних системах і мережах, сприяє підтримці безпеки систем.

3. Системи виявлення вторгнень IDS на основі методів машинного навчання. Потреба в використанні ефективної IDS з використанням машинного навчання стала надзвичайно важливою для виявлення нових і просунутих атак, які не виявляють традиційні IDS і брандмауер.

4. Система запобігання вторгненням (IPS). Функціональні можливості IPS такі ж, як і IDS, але потребують запобіжних заходів. Крім того, порівняно з IDS, IPS можна класифікувати на два основних типи; систему запобігання вторгненням на основі хоста (HIPS) і систему запобігання вторгнень на основі мережі (NIPS).

Огляд світових кіберзлочинів

Слайд 4

Найпоширеніші кібератаки:

1. Фішинг.
2. Шкідливе програмне забезпечення.
3. Розсилка спаму електронною поштою
4. Розподілена відмова в обслуговуванні.
5. SQL Injection.
6. Тунелювання DNS.
7. Міжсайтова атака (XSS).
8. Атака обходу шляху. Процес загрози для доступу до збережених файлів або каталогів

Рисунок 2- 20 найбільших міжнародних постраждалих країн

Таблиця 2 - Список найбільших корпоративних кіберзлочинів

Компанія	Рік	Опис атаки	Довідка
Marriott	2018	Ця атака сталася, коли зловмисник отримав несанціонований доступ до системи бронювання Marriott. Було викрадено інформацію про 500 мільйонів клієнтів, пов'язану з їхніми іменами, банківськими картками та паспортами. Це порушення спричинило штраф у розмірі 124 мільйонів доларів через збій захисту персональних даних клієнтів.	CNN
Facebook	2019	Згідно зі звітом UpGuardIn, дослідники безпеки виявили великий збір даних серед користувачів Facebook. Вони публічно з'являлися в хмарі серверів Amazon для завантаження без будь-якого дозволу. Ця атака спричинила штраф у розмірі 5 мільярдів доларів через втрату контролю над великими масивами даних приватних користувачів.	CNN
Capital One	2019	У 2019 році сервер Capital One (COF) був зламаний зловмисником. Порушення дозволило отримати 140 000 номерів соціального страхування, а також один мільйон номерів канадського страхування, 80 000 облікових записів, пов'язаних з банківським сектором, і невідому кількість особистої інформації, такої як імена, адреси, кредитні рейтинги та іншу інформацію.	CNN
Mitsubishi	2020	Mitsubishi заявляє, що незаконне угруповання атакувало компанію Mitsubishi за допомогою масштабної кібератаки, витягнувши цінну інформацію про 8000 осіб, а також важливі дані для партнерських компаній. Більше того, злом скомпрометував чутливі державні установи, які містили секретні проекти щодо обладнання захисту.	CSIS

Особливостей застосування методів машинного навчання в кібербезпеці

Слайд 5

Рисунок 3- Етапи машинного навчання

Методи машинного навчання.

1.Дерево рішень.
2.Метод К-найближчі сусіди.
3.Метод машини опорного вектора (SVM).
4.Техніка ансамблю:
- *випадковий ліс;*
- *підсилення.*

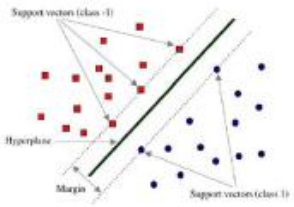


Рисунок 4- Техніка опорного вектора

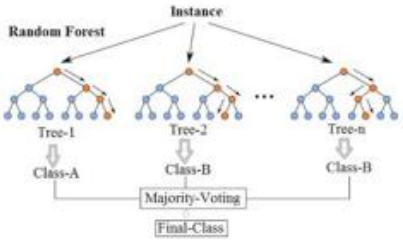


Рисунок 5– Техніка ансамблю випадкового лісу

Система виявлення вторгнень на основі методів машинного навчання

Слайд 6

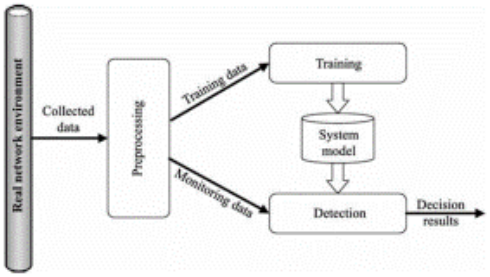


Рисунок 6 - Виявлення вторгнень на основі методів машинного навчання

Таблиця 3 - Пов'язані роботи, які використовували надійний набір даних у розробці IDS

Дослід-ник	Тип IDS	Фаза попередньої обробки	No. Занять	Використа-ний набір даних	Найкраща модель
Sharafal din I et al	IDS для DDOS	Вилучення функцій	13	CICDDoS 2019	Випадковий ліс - Точність: 0,77 - Згадайте 0,65 - Оцінка F1 0,69
Thabtah et al.	Модель захисту від фішингу	Вибір функції	2	Фішингові веб-сайти	Етап нейронної мережі (500) - Acc = 95,06% - Оцінка F1 = 92,30% - Відкликання = 91,12% - Точність = 93,71%
Elsayed MS et al.	Виявлення мережевих атак	-Особливості зменшення - Очищення даних - Нормалізація даних	2	CICDDoS 2019	Глибоке навчання - Точність: 0,99
Mamun MSI et al.	Виявлення шкідливих URL-адрес	Вибір функції	5	ISCX-URL-2016	Випадковий ліс - Точність: 0,97 - Відкликання: 0,97
Kapil Det al.	Виявлення шкідливих URL-адрес	Вибір і скорочення функцій	5	ISCX-URL-2016	Weka: Випадковий ліс - TPR:0,961;FPR:0,032 - Точність: 0,961 - Згадайте 0,961 - Оцінка F1 0,961

Аналіз архітектури Host-based Intrusion Detection System

Слайд 7

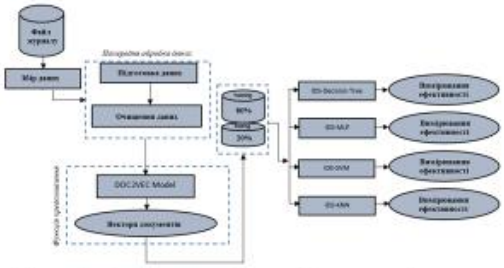


Рисунок 7 - Досліджувана архітектура із фазою впровадження

Таблиця 4 - Застосування процесу попередньої обробки

Попередня обробка даних	
Очищення даних	- Запити розширення звуків і зображень слід видалити - Зберігання запиту щодо статусу серії помилок - Видалення запиту, який не має параметра з успішним статусом
Після попередньої обробки	Атака буде згенерована та використана як вхідні дані для алгоритмів навчання

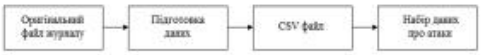


Рисунок 8 - Файл журналу з етапами попередньої обробки

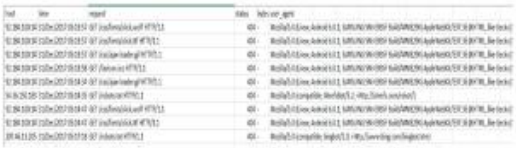


Рисунок 9 - Приклад даних на етапі попередньої обробки

Експериментальні результати та їх обговорення

Слайд 8

Таблиця 5 - Найвищий рівень виявлення, досягнутий класифікаторами для кожної атаки

		Класифікатор			
		Decision Tree	MLP	SVM	KNN
Тип атаки	Path-Traversal	0.8214	0.9260	0.9541	0.9184
	SQLI	0.9215	0.9469	0.8776	0.9330
	XSS	0.8320	0.8400	0.6347	0.7840

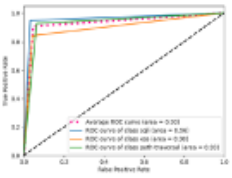


Рисунок 10 - Крива MLP ROC

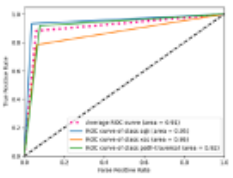


Рисунок 11 - Крива KNN ROC

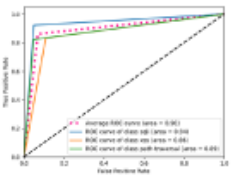


Рисунок 12 - Крива ROC дерева рішень

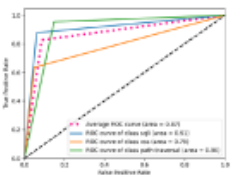


Рисунок 13 - Крива SVM ROC

На рис. 10-13 показано площу під робочою характеристикою приймача для кожної застосованої моделі, яка була розрахована на основі істинно позитивних і хибно позитивних результатів. Велике значення ROC показує здатність моделі виявляти зловмисники, тоді як низьке значення демонструє слабкість цієї моделі. Можемо зробити висновок, що класифікатори MLP зареєстрували найвищий показник 93%, тоді як KNN зафіксував 91%, дерево рішень (90%) і SVM показали найнижче значення (87%).

Експериментальні результати та їх обговорення

Слайд 9

Таблиця 6-Результати точності різних методів ML

Класифікатори машинного навчання	Точність
MLP	0.9067
KNN	0.8817
Decision Tree	0.8608
SVM	0.8267

Таблиця 7-Звіт про класифікацію для моделі дерева рішень

	Точність	Відклик	
Path Traversal	0,8920	0,8214	
SQLi	0,9130	0,9215	
XSS	0,7761	0,8320	
Accuracy			0,8608
Macro avg	0,8604	0,8583	
Weighted avg	0,8634	0,8608	

Таблиця 8-Звіт про класифікацію для моделі MLP

	Точність	Відклик	
Path Traversal	0,8747	0,9260	
SQLi	0,9425	0,9496	
XSS	0,9000	0,8400	
Accuracy			0,9067
Macro avg	0,9057	0,9043	
Weighted avg	0,9071	0,9067	

Таблиця 9-Звіт про класифікацію для моделі SVM

	Точність	Відклик	
Path Traversal	0,7571	0,9541	
SQLi	0,8899	0,8776	
XSS	0,8530	0,6347	
Accuracy			0,8267
Macro avg	0,8334	0,8221	
Weighted avg	0,8358	0,8267	

Таблиця 10-Звіт про класифікацію для моделі KNN

	Точність	Відклик	
Path Traversal	0,8511	0,9184	
SQLi	0,9352	0,9330	
XSS	0,8522	0,7840	
Accuracy			0,8817
Macro avg	0,8795	0,8785	
Weighted avg	0,8818	0,8817	

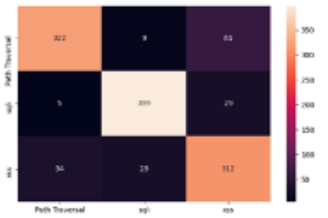


Рисунок 14 - Матриця плутанини для моделі дерева рішень

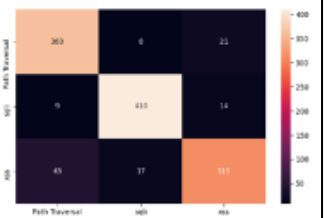


Рисунок 15 - Матриця плутанини для моделі MLP

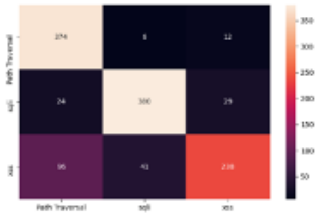


Рисунок 16 - Матриця помилок для моделі SVM

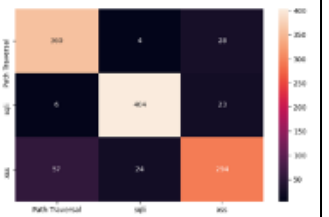


Рисунок 17 - Матриця плутанини для моделі KNN

МЕТА ТА ЗАВДАННЯ

Слайд 10

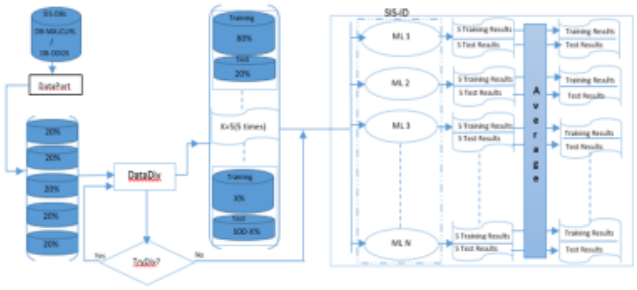


Рисунок 18 - Загальна архітектура методології навчання SIS-ID на основі прикладних методів машинного навчання

Прикладні методи машинного навчання.

1 Контрольоване навчання.
- дерево рішень;
- алгоритм k-найближчого сусіда;
- багатокласні прийоми.
- класифікатор один проти одного
- Класифікатор один проти решти.

2. Ансамблеві прийоми.
- XGBoost
- випадковий ліс
- класифікатор Adaboost

3. Розвиток техніки ансамблю:
- класифікатор мішковий;
- класифікатор голосування.

4. Навчання без контролю
- Фактор локального виключення (LOF).

Застосування системи SIS-ID на DB-MALCURL

Слайд 11

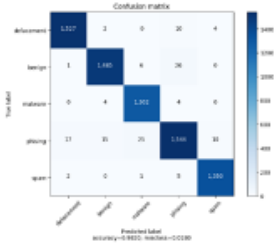


Рисунок 18- Матриця помилок для моделі OVR на DB-MALCURL

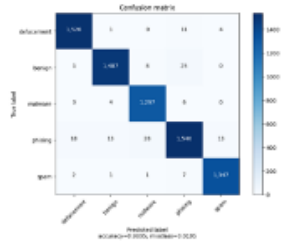


Рисунок 19- Матриця помилок для моделі OVO на DB-MALCURL

Таблиця 11 - Результати застосованих методів навчання під наглядом, перевірених за допомогою DB-MALCURL

Модель	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Точність
OneVsRest	0.982861	0.982111	0.982456	0.982016
OneVsOne	0.981371	0.980555	0.980926	0.980518
KNN	0.964221	0.965574	0.964551	0.964033
Decision Tree	0.955805	0.956841	0.956249	0.955586

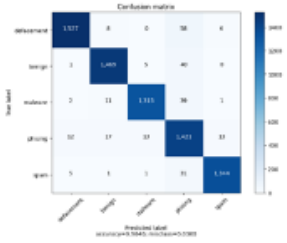


Рисунок 20- Матриця плутанини для моделі KNN на DB-MALCURL

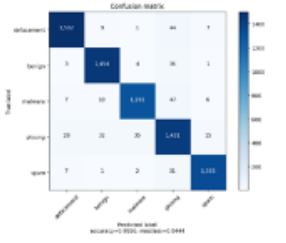


Рисунок 21 - Матриця помилок для моделі дерева рішень на DB-MALCURL

Застосування системи SIS-ID на DB-DDOS

Слайд 12

Таблиця 12 - Результати застосованих методів навчання під наглядом, перевірених через DB-DDOS

Модель	Precision Macro Average	Recall Macro Average	F1-Score Macro Average	Точність
OneVsRest	0.7961	0.7685	0.7604	0.7682
OneVsOne	0.7944	0.7682	0.7603	0.7679
Decision Tree	0.793	0.7675	0.7597	0.7672
KNN	0.7352	0.7252	0.7199	0.7248

Таблиця 13 - Результати виявлення кожного класу за допомогою моделей навчання під наглядом, які перевірялися через DB-DDOS

Attack	Nb. Instance	Coefficient	One Vs Rest		One Vs One		Decision Tree		KNN	
			Rate	Detect	Rate	Detect	Rate	Detect	Rate	Detect
BENIGN	11133	Precision	99.54%	10998	99.56%	10995	99.49%	10993	93.65%	10337
		Recall	98.78%		98.67%		98.79%		98.67%	
		F1-score	99.16%		99.08%		99.16%		99.08%	
DrDoS_ DNS	11195	Precision	81.89%	5493	81.83%	5500	79.97%	5479	47.37%	8214
		Recall	49.07%		48.94%		Recall		49.12%	
		F1-score	61.39%		60.72%		F1-score		61.40%	
DrDoS_ LDAP	11175	Precision	52.03%	8355	52.02%	8362	51.94%	8285	50.86%	4763
		Recall	74.77%		74.14%		Recall		74.93%	
		F1-score	61.35%		61.09%		F1-score		61.37%	
DrDoS_ MSSQL	11210	Precision	71.39%	4905	71.58%	4925	71.96%	4928	48.20%	5011
		Recall	43.76%		43.96%		Recall		43.92%	
		F1-score	54.26%		54.36%		F1-score		54.43%	
DrDoS_ NetBIOS	11127	Precision	97.47%	10550	97.57%	10551	97.28%	10549	95.47%	10185
		Recall	94.81%		94.81%		Recall		94.92%	
		F1-score	96.13%		96.03%		F1-score		96.18%	
DrDoS_ NTP	11265	Precision	79.64%	8076	78.86%	8071	79.13%	8091	76.50%	7745
		Recall	71.69%		71.82%		Recall		71.65%	
		F1-score	75.18%		75.31%		F1-score		75.08%	
DrDoS_ SNMP	11262	Precision	89.72%	11156	89.75%	11155	89.68%	11151	89.67%	11114
		Recall	99.06%		99.01%		Recall		99.05%	
		F1-score	94.16%		94.12%		F1-score		94.16%	
DrDoS_ SSDP	11525	Precision	61.85%	9072	62.55%	8759	61.78%	8828	68.44%	6435
		Recall	80.11%		77.99%		Recall		77.58%	
		F1-score	69.68%		68.92%		F1-score		69.16%	
DrDoS_ UDP	11186	Precision	70.04%	10085	69.97%	10078	70.53%	11186	68.84%	9209
		Recall	90.16%		89.04%		Recall		90.09%	
		F1-score	78.84%		78.71%		F1-score		78.75%	

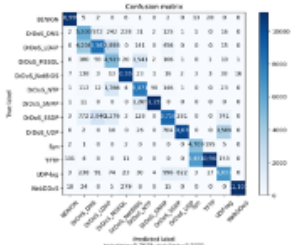


Рис.22 - Матриця помилок для моделі OVO в DB-DDOS

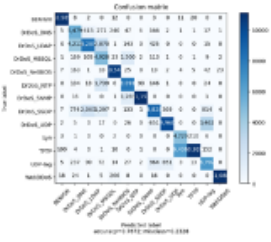


Рисунок 23- Матриця помилок для моделі дерева рішень у DB-DDOS

Апаратне моделювання в реальному часі

Слайд 13

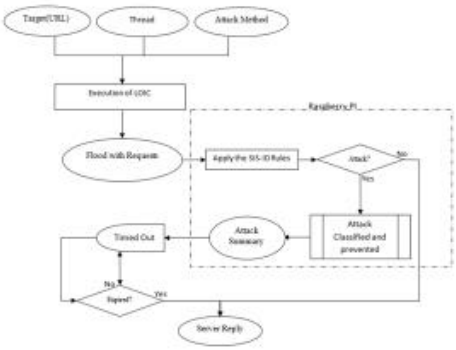


Рисунок 24 - Загальна архітектура апаратного моделювання в реальному часі SIS-ID

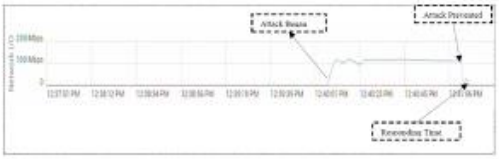


Рисунок 25 - Ефективність виявлення наступної атаки на етапі реального часу

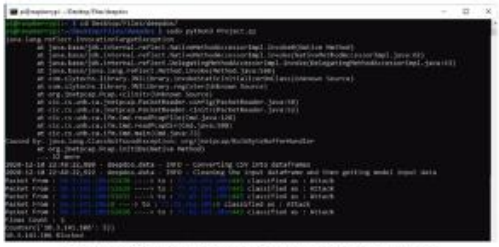


Рисунок 26 - Результат роботи нашого апаратного забезпечення для уникнення майбутньої атаки DOS

Висновки

Слайд 14

Таким чином, виконуючи поставлені завдання у роботі проаналізовано основні поняття кібербезпеки. Виконано огляд термінів та наведено їх класифікацію. Досліджено основні технології кібербезпеки. Для розуміння небезпеки яку несуть кібернетичні атаки і їх результат у першому розділі приведено приклад маштабних кібератак. Визначено та представлено найпоширеніші кібератаки.

Досліджено функціонування системи виявлення вторгнень на основі хосту (HIDS), використовуючи техніку аналізу тексту. Виконуючи аналіз було використано чотири різні методи машинного навчання (KNN, SVM, Decision Tree і MLP), щоб обрати найкращу ефективну модель класифікації. Отриманий результат показав, що метод MLP зафіксував найкращу точність у 90,67 %.

Виконуючи останнє завдання, досліджено виявлення вторгнень на основі інтелектуальної системи безпеки за допомогою методів машинного навчання, які призначена для виявлення зловмисних URL-адрес і розширену для атак розподіленої відмови в обслуговуванні (DDoS). Досліджена інтелектуальна система безпеки показує високі показники виявлення при прогнозуванні як шкідливих URL-адрес, так і атак DDOS на основі декількох моделей машинного навчання.

Досліджувана система SIS-ID бла перевірений як апаратне забезпечення для запобігання вторгненню з ефективністю запобігання атакам відмови в обслуговуванні (DOS) на етапі реального часу. В результаті експерименту атаку було ідентифіковано та заблоковано.

ДЯКУ ЗА УВАГУ, ДОПОВІДЬ ЗАВЕРШЕНО