Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп’ютерної інженерії та управління_____
(повна назва)

Кафедра _____Безпеки інформаційних технологій _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Метод виявлення вторгнень в комп'ютерну мережу на основі технологій машинного навчання
(тема)

Виконала:
студентка __2_ курсу, групи _БІКСм-20-1_
_____Кононова Г.О._____
(прізвище, ініціали)

Спеціальність _____125 Кібербезпека ____
(код і повна назва спеціальності)

Освітня програма __«Безпека інформаційних і комунікаційних систем»
(повна назва освітньої програми)

Керівник __доц. Мартовицький В.О._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ _____Халімов Г.З.__
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерної інженерії та управління _____

Кафедра_____Безпеки інформаційних технологій_____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність_____125 Кібербезпека _____
(код і повна назва)

Тип програми _____освітньо-професійна_____
(освітньо-професійні, або освітньо-наукова)

Освітня програма_____«Безпека інформаційних і комунікативних систем»_____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Кононової Ганни Олександрівни_____
(прізвище, ім'я, по батькові)

1. Тема роботи ___ Метод виявлення вторгнень в комп'ютерну мережу на основі технологій машинного навчання_____
_____

затверджена наказом по університету від _25_ ____10___2021 р. № __166Стз__ ___

2. Термін подання студентом роботи до екзаменаційної комісії _10 грудня_____2021 р.

3. Вихідні дані до роботи _аналіз методів виявлення вторгнень, аналіз наборів багатовимірних даних _____

4. Перелік питань, що потрібно опрацювати в роботі

Аналіз предметної області: безпечка мережі, види мережевих загроз та атак, системи виявлення вторгнень та системи запобігання вторгненням _____

Набір інструментів та набори даних _____

Виявлення аномалій у мережах з використанням машинного навчання _____

Реалізація машиного навчання _____

Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _презентаційний матеріал у вигляді слайдів _____

_____

_____

_____

_

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	25.10.2021	Виконано
2	Затвердження плану і завдання кваліфікаційної роботи	25.10.2021	Виконано
3	Робота з джерелами за тематикою роботи	25.10.2021-10.11.2021	Виконано
4	Аналіз завдання, пошук та аналіз літературних джерел за темою роботи	25.10.2021-10.11.2021	Виконано
5	Виконання кваліфікаційної роботи	25.10.2021-10.12.2021	Виконано
6	Оформлення пояснювальної записки	01.12.2021-10.12.2021	Виконано
7	Здача на перевірку та підпис кваліфікаційної роботи керівнику	10.12.2021	Виконано
8	Проходження перевірки на плагіат та нормоконтроль кваліфікаційної роботи	13.12.2021	Виконано
9	Допуск завідувачем кафедри до захисту кваліфікаційної роботи	15.12.2021	Виконано
10	Захист кваліфікаційної роботи	16.12.2021	Виконано

Дата видачі завдання _1_____вересня_____2021 р.

Студент _____
(підпис)

Керівник роботи _____ ____доц. Мартовицький В.О._
(підпис) (посада, прізвище, ініціали

РЕФЕРАТ

Звіт про кваліфікаційну роботу містить : 85 с., 34 рис., 5 табл., 34 джерел.

БЕЗПЕКА МЕРЕЖІ, ЗАГРОЗИ, ВТОРГНЕННЯ, БАГАТОВИМІРНІ ДАНІ, МАШИННЕ НАВЧАННЯ

Об’єкт дослідження – заналіз та абезпечення безпеки даних за допомогою методів виявлення вторгнень та методів запобігання вторгненням.

Предмет дослідження – вторгнення в комп’ютерну мережу.

Мета роботи – аналіз та дослідження методів виявлення вторгнень в комп’ютерну мережу на основі технологій машинного навчання.

Методи дослідження – аналіз методів виявлення аномалій у мережах з використанням машинного навчання.

У ході виконання роботи представлено види мережевих загроз та атак, проаналізовано системи виявлення вторгнень, а також види систем запобігання вторгненням. Таким чином було описано та проаналізовано набори даних, а саме: багатовимірні дані, виявлення викидів часових рядів, виявлення аномалій а режимі реального часу, та контрольні показники мета-аналізу виявлення аномалій. Запропоновано та реалізовано виявлення вторгнень в комп’ютерну мережу на основі технологій машиного навчання. Аналіз аномалій засобами машинного навчання має великий простір подальших досліджень, особливо в області виявлення аномалій та вторгнень

ABSTRACT

The report on the qualification work contains: 85 pages, 34 figures, 5 tables, 34 sources.

NETWORK SECURITY, THREATS, INVASIONS, MACHINE LEARNING, MULTIDIMENSIONAL DATA

The object of research is the analysis and security of data using intrusion detection and intrusion prevention methods.

The subject of the research is intrusion into a computer network.

The purpose of the work is to analyze and study methods of detecting intrusions into a computer network based on machine learning technologies.

Research methods – analysis of methods for detecting anomalies in netwc using machine learning.

In the course of the work the types of network threats and attacks are presented, the systems of intrusion detection are analyzed, as well as the types of intrusion prevention systems. Thus, data sets were described and analyzed, namely: multidimensional data, time series emission detection, real-time anomaly detection, and anomaly detection meta-analysis benchmarks. Detection of intrusions into a computer network based on machine learning technologies has been proposed and implemented. Analysis of anomalies by machine learning has a lot of space for further research, especially in the field of detection of anomalies and intrusions

ЗМІСТ

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ADS (Anomaly Detection System) – Системи виявлення аномалій;

DoS (Denial of Service) – Відмова в обслуговування;

DDoS (Distributed Denial of Service) – Розподілений DoS;

HTTP (Hypertext Transfer Protocol) – Протокол передачі гіпертексту;

IDS (Intrusion Detection System) – Системи виявлення вторгнень;

IPS (Intrusion Prevention System) – Система попередження вторгнень;

LOF (Local Outlier Factor) – Локальний коефіцієнт викидів;

NAB (Numenta Anomaly Benchmark) – Показник аномалій;

NAC (Network access control) – Контроль доступу до мережі;

NBA (Network Behavior Analysis) – Аналіз поведінки мережі;

NIPS (Network Intrusion Prevention System) – мережева система запобігання вторгненню;

ODDS (Outlier Detection DataSets) – Набори даних виявлення викидів;

PyOD (Python Outlier Detection) – Виявлення віддалених об'єктів у багатовимірних даних;

PySAD (Python Streaming Anomaly Detection) – Виявлення аномалій потокової передачі;

SUOD (Scalable Unsupervised Outlier Detection) – Масштабоване неконтрольоване виявлення викидів;

TCP (Transmission Control Protocol) – Протокол управління передачею даних;

TODS (Automated Time-series Outlier Detection System) – Автоматизована система машинного навчання;

VPN (Virtual Private Network) – Віртуальні приватні мережі;

UDP (User Datagram Protocol) – Протокол дейтаграм користувача;

СВВ – Системи виявлення вторгнень;

Експло́йт (від англ. exploit – експлуатувати) – це комп'ютерна програма, фрагмент програмного коду або послідовність команд, що використовують вразливості в програмному забезпеченні та призначені для проведення атаки на обчислювальну систему. Метою атаки може бути як захоплення контролю над системою (підвищення привілеїв), так і порушення її функціонування (DoS-атака).

ІТ – Інформаційні технології;

ПЗ – Програмне забезпечення;

ПК – Персональний комп’ютер.

ВСТУП

Сьогодні вже широко використовуються системи виявлення вторгнень (СВВ). Вони широко поширені в корпоративних інформаційних системах і комп'ютерних мережах. Існуючі СВВ несуть в собі багато хибної інформації і не виявляють всі відомі атаки на інформаційні ресурси. У цьому плані розвиток СВВ схожий на нещодавні тенденції антивірусного програмного забезпечення. Ранні версії антивірусних програм також невиправдано заважала щоразу, коли користувач створював або завантажував нові файли.

Але останнім часом антивірусне програмне забезпечення значно модернізовано. Зараз користувачам важко звернути увагу на дії антивірусних програм на власних ПК. Однак більшість із користувачів впевнені, що антивірусне ПЗ виявить всі відомі віруси.

Концепція створення системи виявлення вторгнень була оригінальною та запропонованою у 1980 році Джеймсом Андерсоном. Але це науковий напрямок залишався невивченим до 1987 року, доки Дороті Деннінг не опублікувала модель виявлення перешкод. А в 1988 році існувало щонайменше три прототипи СВВ. В наступних роках кількість прототипів неухильно зростала.

Тому що виявлення перешкод стало зрілою технологією і сферою промислового інтересу, майже всі прості проблеми було успішно вирішено. Останнім часом жодних досліджень у цій області не проводилося. Натомість покращуються та модернізуються існуючі методи виявлення вторгнень. Тому очікується, що дослідження будуть зосереджені на виявленні перешкод у відносно невивчених областях, наприклад:

- механізми реагування на атаку;
- архітектура високорозповсюджених систем виявлення вторгнень;
- стандарти взаємодії компонентів системи при виявленні вторгнень;

- нові парадигми виявлення вторгнень.

Оцінка неконтрольованих алгоритмів виявлення вторгнень є постійною проблемою в дослідженні аналізу даних. Мало відомо про сильні та слабкі сторони різних стандартних моделей виявлення вторгнень, а також про вплив вибору параметрів для цих алгоритмів. Дефіцит відповідних контрольних наборів даних є значною перешкодою для оцінки методів, що виділяються. Навіть якщо доступні марковані набори даних, їх придатність для завдання виявлення вторгнень зазвичай невідома. Крім того, упередження загальновживаних заходів оцінки не повністю зрозумілі. Таким чином, важко визначити, наскільки нові методи виявлення вторгнень покращуються порівняно з усталеними. У цій роботі розглядаються найбільш часто використовувані показники для порівняння ефективності різних методів і пропонуються ті, які більше підходять для оцінки результатів виявлення вторгнень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Безпека мережі

Безпека мережі – це широкий термін, який охоплює безліч технологій, пристроїв і процесів. Простіше кажучи, це набір правил і конфігурацій, призначених для захисту цілісності, конфіденційності та доступності комп’ютерних мереж і даних за допомогою програмних і апаратних технологій.

Сучасна мережева архітектура є складною і стикається з середовищем загроз, яке постійно змінюється, і зловмисниками, які завжди намагаються знайти та використати вразливі місця. Ці вразливості можуть існувати в багатьох областях, включаючи пристрої, дані, програми, користувачів і місця розташування. З цієї причини сьогодні використовується багато інструментів і програм керування безпекою мережі, які вирішують окремі загрози та експлойти, а також невідповідність нормативним вимогам [1].

Як працює безпека мережі?

Вирішуючи питання безпеки мережі в організації, слід враховувати багато рівнів. Атаки можуть відбуватися на будь-якому рівні в моделі рівнів безпеки мережі, тому обладнання, програмне забезпечення та політика безпеки мережі мають бути розроблені для кожної області.

Безпека мережі зазвичай складається з трьох різних елементів контролю: фізичного, технічного та адміністративного, що показано на рисунку 1.1. Нижче наведено короткий опис різних типів мережевої безпеки та принципів роботи кожного елемента керування.

Фізична безпека мережі

Засоби контролю фізичної безпеки призначені для запобігання неавторизованому персоналу отримати фізичний доступ до компонентів

мережі, таких як маршрутизатори, кабельні шафи тощо. Контрольований доступ, такий як замки, біометрична аутентифікація та інші пристрої, є важливим у будь-якій організації.

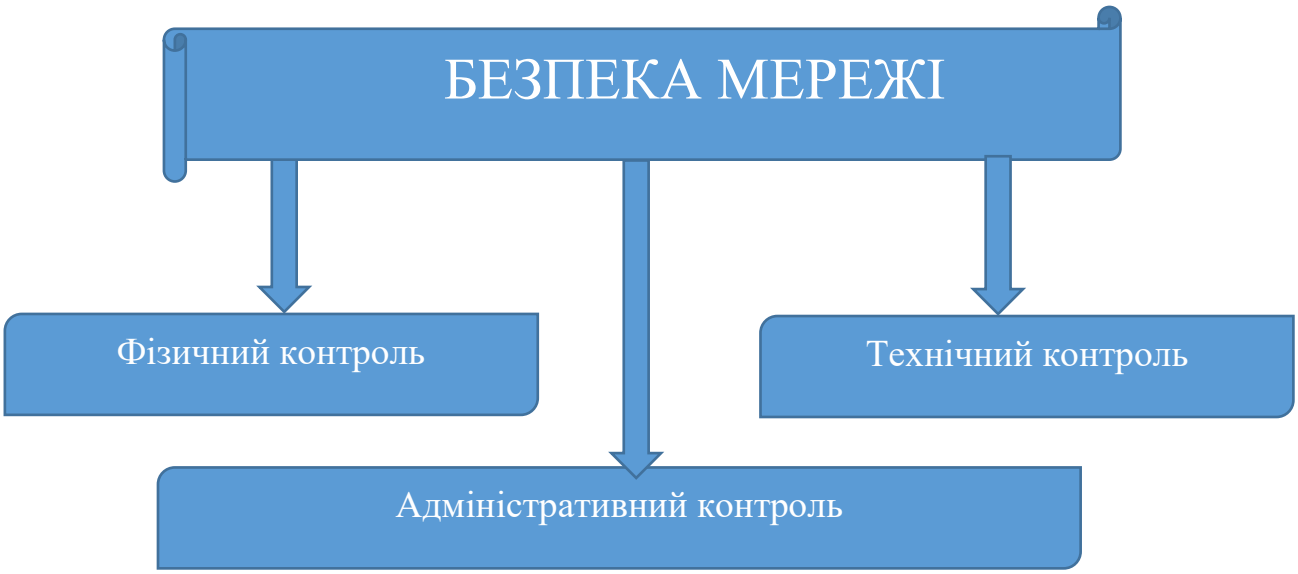


Рисунок 1.1 – Елементи контролю безпеки мережі

Технічна мережева безпека

Засоби технічного контролю безпеки захищають дані, які зберігаються в мережі або передаються через мережу, в мережу або з неї. Захист подвійний; йому потрібно захищати дані та системи від неавторизованого персоналу, а також захищати від зловмисних дій співробітників.

Адміністративна мережева безпека

Адміністративні засоби контролю безпеки складаються з політик безпеки та процесів, які контролюють поведінку користувачів, зокрема спосіб автентифікації користувачів, їхній рівень доступу, а також те, як співробітники IT вносять зміни в інфраструктуру.

Види мережевої безпеки:

1 Контроль доступу до мережі

Щоб гарантувати, що потенційні зловмисники не можуть проникнути у вашу мережу, для користувачів і пристроїв мають бути встановлені комплексні політики контролю доступу. Контроль доступу до мережі (NAC) можна налаштувати на найбільш детальному рівні. Наприклад, ви можете надати адміністраторам повний доступ до мережі, але заборонити доступ до певних конфіденційних папок або заборонити їхнім особистим пристроям приєднатися до мережі.

2 Антивірусне програмне забезпечення

Антивірусне програмне забезпечення захищає організацію від ряду шкідливих програм, включаючи віруси, програми-вимагачі, хробаки та трояни. Найкраще програмне забезпечення не тільки сканує файли при вході в мережу, але й безперервно сканує та відстежує файли.

3 Захист брандмауера

Брандмауери, як випливає з їх назви, діють як бар’єр між ненадійними зовнішніми мережами та вашою довіреною внутрішньою мережею. Адміністратори зазвичай налаштовують набір визначених правил, які блокують або дозволяють трафік до мережі. Наприклад, брандмауер наступного покоління Forcepoint (NGFW) пропонує плавний і централізовано керований контроль мережевого трафіку, будь то фізичний, віртуальний або хмарний.

4 Віртуальні приватні мережі

Віртуальні приватні мережі (VPN) створюють підключення до мережі з іншої кінцевої точки або сайту. Наприклад, користувачі, які працюють вдома, зазвичай підключаються до мережі організації через VPN. Дані між двома точками зашифровані, і користувачеві потрібно пройти аутентифікацію, щоб дозволити зв’язок між пристроєм і мережею. Захищений корпоративний SD-WAN Forcepoint дозволяє організаціям швидко створювати VPN за допомогою перетягування та захищати всі місця за допомогою нашого рішення брандмауера наступного покоління.

1.2 Види мережевих загроз та атак

Основні мережеві атаки в комп’ютерній мережі

Багато людей покладаються на Інтернет для багатьох своїх професійних, соціальних та особистих видів діяльності. Але є також люди, які намагаються пошкодити наші комп’ютери, підключені до Інтернету, порушують нашу конфіденційність і виводять з ладу Інтернет-послуги.

Враховуючи частоту та різноманітність існуючих атак, а також загрозу нових і більш руйнівних атак у майбутньому, безпека мережі стала центральною темою в області комп’ютерних мереж [2].

Наскільки вразливі комп’ютерні мережі? Які типи атак є найбільш поширеними сьогодні?

Шкідливе програмне забезпечення – скорочення від шкідливого програмного забезпечення, спеціально розробленого для порушення, пошкодження або отримання авторизованого доступу до комп’ютерної системи. Значна частина зловмисного програмного забезпечення, яке існує сьогодні, саморозмножується: як тільки воно заражає один хост, з цього хоста воно шукає доступу до інших хостів через Інтернет, а від нещодавно інфікованих хостів воно шукає доступу до ще інших хостів. Таким чином, зловмисне програмне забезпечення, що самовідтворюється, може поширюватися експоненціально швидко.

Вірус – зловмисне програмне забезпечення, яке вимагає певної взаємодії користувача для зараження пристрою користувача. Класичним прикладом є вкладення електронного листа, що містить шкідливий виконуваний код. Якщо користувач отримує та відкриває такий вкладений файл, він випадково запускає шкідливе програмне забезпечення на пристрої.

Черв’як – зловмисне програмне забезпечення, яке може проникнути на пристрій без будь-якої явної взаємодії з користувачем. Наприклад, користувач може використовувати вразливу мережеву програму, до якої зловмисник може

надсилати шкідливе програмне забезпечення. У деяких випадках без будь-якого втручання користувача програма може прийняти зловмисне програмне забезпечення з Інтернету та запустити його, створивши хробака.

Ботнет – мережа приватних комп’ютерів, заражених шкідливим програмним забезпеченням, які контролюються як група без відома власників, наприклад. розсилати спам.

DoS (відмова в обслуговуванні) – атака DoS робить мережу, хост або інші частини інфраструктури непридатними для використання законними користувачами. Більшість DoS-атак в Інтернеті поділяються на одну з трьох категорій:

1) Атака через вразливість: це передбачає відправку кількох добре складених повідомлень уразливому додатку або операційній системі, що працює на цільовому хості. Якщо правильна послідовність пакетів надсилається до вразливої програми чи операційної системи, служба може зупинитися або, що ще гірше, хост може вийти з ладу.

2) Переповнення пропускної спроможності: зловмисник посилає потік пакетів цільовому хосту – таку кількість пакетів, що канал доступу цілі забивається, не даючи легітимним пакетам досягати сервера.

3) Переповнення з'єднання: зловмисник встановлює велику кількість напіввідкритих або повністю відкритих TCP-з'єднань на цільовому хості. Хост може настільки захопитися цими фіктивними з’єднаннями, що перестане приймати законні з’єднання.

DDoS – це такий тип розподіленої атаки DOS, коли кілька зламаних систем використовуються для націлювання на одну систему, що спричиняє атаку «Відмова в обслуговуванні» (DoS). DDoS-атаки з використанням бот-мереж з тисячами хостів є поширеним явищем сьогодні. Атаки DDoS набагато важче виявити та захистити від DoS-атаки з одного хоста.

Пасивний приймач, який записує копію кожного пакета, який пролітає, називається сніфером пакетів. Розмістивши пасивний приймач поблизу

бездротового передавача, цей приймач може отримати копію кожного пакету, який передається! Ці пакети можуть містити всі види конфіденційної інформації, включаючи паролі, номери соціального страхування, комерційні таємниці та особисті повідомлення. Деякі з найкращих засобів захисту від перехоплення пакетів включають криптографію.

IP Spoofing – можливість вводити в Інтернет пакети з помилковою адресою джерела відома як IP spoofing, і це лише один із багатьох способів, за допомогою яких один користувач може маскуватися під іншого користувача. Щоб вирішити цю проблему, нам знадобиться аутентифікація кінцевої точки, тобто механізм, який дозволить нам з упевненістю визначити, чи походить повідомлення з того місця, де воно походить.

Атака «Людина посередині» – як випливає з назви, атака «Людина посередині» відбувається, коли хтось між вами та особою, з якою ви спілкуєтеся, активно відстежує, фіксує та прозоро контролює ваше спілкування. Наприклад, зловмисник може перенаправити обмін даними. Коли комп’ютери спілкуються на низьких рівнях мережевого рівня, комп’ютери можуть бути не в змозі визначити, з ким вони обмінюються даними.

Атака скомпрометованого ключа – ключ – це секретний код або число, необхідне для інтерпретації захищеної інформації. Хоча отримання ключа є складним і ресурсомістким процесом для зловмисника, це можливо. Після того, як зловмисник отримує ключ, цей ключ називають скомпрометованим. Зловмисник використовує скомпрометований ключ, щоб отримати доступ до захищеного зв’язку, при цьому відправник чи одержувач не знають про атаку.

Фішинг – шахрайська практика надсилання електронних листів від авторитетних компаній з метою спонукати людей розкрити особисту інформацію, таку як паролі та номери кредитних карток.

Спуфінг DNS – також відомий як отруєння кешу DNS, є формою злому комп’ютерної безпеки, при якому пошкоджені дані системи доменних імен вводяться в кеш розпізнавача DNS, через що сервер імен повертає неправильну IP-адресу.

Руткіти – це приховані пакети, призначені для отримання прав адміністратора та отримання права доступу до інструменту спільноти. Після встановлення хакери отримують повне та необмежене право доступу до інструменту, а отже, можуть безперешкодно виконувати будь-які дії, включаючи шпигування за клієнтами або крадіжку ексклюзивних даних.

1.3 Системи виявлення вторгнень

Виявлення вторгнень вивчається протягом останніх 20 років. Вторгнення – це діяльність, яка порушує політику безпеки інформаційної системи [3]. Виявлення вторгнень засноване на припущенні, що поведінка порушника істотно відрізнятиметься від нормальної поведінки, що забезпечить виявлення великої кількості несанкціонованих дій.

Системи виявлення вторгнень зазвичай використовуються спільно з іншими системами захисту, такими як контроль доступу та аутентифікації як додатковий захист інформаційних систем [4]. Є багато причин, які роблять виявлення вторгнень важливою частиною у всій системі захисту. По-перше, багато з існуючих систем та додатків, були розроблені та побудовані без урахування вимог безпеки. По-друге, комп'ютерні системи та програми можуть мати недоліки або помилки в їх конфігурації, які можуть бути використані зловмисники для атаки систем або програм. Таким чином, профілактичний метод не може бути таким же ефективним, як і очікувалося.

Системи виявлення вторгнень можна розділити на два класи: системи виявлення сигнатур та системи виявлення аномалій. Система виявлення сигнатур ідентифікує шаблони трафіку даних або додатків, які вважаються

шкідливими, у той час як системи виявлення аномалій і порівнюють діяльність із нормальною поведінкою.

Відповідно до [5], [6] всі методи виявлення аномалій складаються з наступних основних модулів чи етапів (рисунок 1.2). Ці етапи параметризація, навчання та виявлення. Параметризація включає збір вихідних даних з контрольованого середовища. Вихідні дані повинні бути типовими для системи, яка має бути змодельована. Етап навчання моделює систему за допомогою ручних чи автоматичних методів.

Для архітектури клієнт-сервер сервер є хост, який очікує вхідне з'єднання. Коли з'єднання встановлюється між клієнтом і сервером, сервер підтверджує сокет, який буде використовуватися для створення екземпляра. Об'єкт обробника, який працює окремому потоці. Ці обробники зберігатимуться в об'єкті колекції.

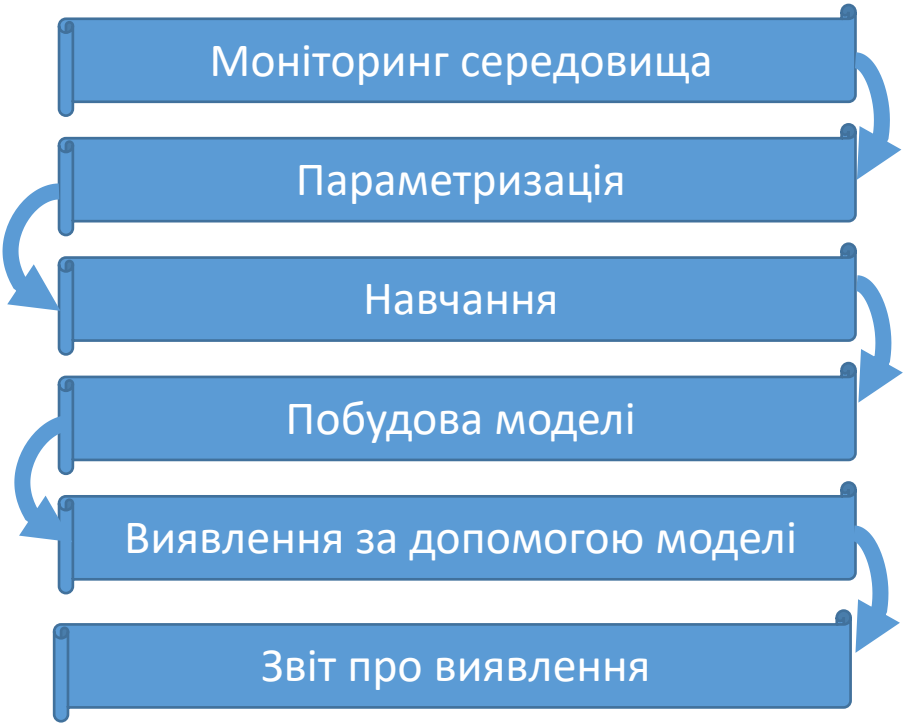


Рисунок 1.2 – Загальна схема виявлення аномалій

Етапи, представлені в моделі, відрізнятимуться залежно від методу, що використовується. При виявленні порівнюється система, створена на етапі моделювання, з обраним параметризованим блоком даних. Порогові критерії будуть обрані визначення аномальної поведінки [6].

Машинне навчання може побудувати необхідну модель автоматично на основі деяких навчальних даних. Застосування такого підходу потребує наявності необхідної підготовки даних, але це завдання є менш складним порівняно з обчисленням аномальної моделі [7]. Зі збільшенням складності та кількості різних атак, методи машинного навчання, які дозволяють створювати та підтримувати системи виявлення аномалій (ADS) з меншим втручанням людини, є єдиним практичним підходом для створення наступного покоління систем виявлення вторгнень.

Застосування методів машинного навчання для виявлення вторгнень дозволить автоматично побудувати модель, засновану на наборі навчальних даних, що містить екземпляри даних, описаних за допомогою набору атрибутів (ознак). Атрибути можуть бути різних типів, наприклад, якісними або кількісними. Були розглянуті різні алгоритми виявлення аномалій, у таблиці 1.1 представлені переваги та недоліки кожного з них [8], [9], [10].

Виявлення аномалій включає контрольовані і неконтрольовані методи. Порівняльний аналіз показав, що контрольовані методи навчання значно перевершують неконтрольовані, якщо тестові дані не містять невідомих атак. Серед контрольованих методів найкраща продуктивність досягається за рахунок нелінійних методів, таких як SVM, багатошаровий персептрон та методів, що базуються на правилах. Неконтрольовані методи, такі як K-середніх, SOM, і один клас SVM показують більш високу продуктивність порівняно з іншими методами, хоча вони відрізняються ефективності виявлення всіх класів атак [11], [12].

Таблиця 1.1 – Переваги та недоліки алгоритмів виявлення аномалій

Методи	Переваги	Недоліки
К-найближчих сусідів	Легко реалізуємо, коли є кілька предикторів. Застосовується для побудови моделей, що обробляють нестандартні типи даних, такі як текст.	Великі вимоги щодо обсягу пам'яті. Залежить від вибору функції подібності, яка вико-ристовується для порівняння екземплярів. Відсутність прин-ципового способу вибору, крім через перехресну перевірку або аналогічний спосіб. Дорога обчислювальна техніка.
Нейронна мережа	Нейронна мережа може виконувати завдання, які не виконає лінійна програма. Коли один елемент не справ-ляється із завданням, метод може продовжити роботу завдяки паралельній обробці даних. Нейронну мережу не потрібно перепрограмувати. Може бути реалізована у будь-якому додатку.	Нейронна мережа потребує навчання. Високий час обробки великих нейронних мереж.
Машина опорних векторів	Знаходження оптимального поділу гіперплощини. Обробляє більшу розмірність даних. Зазвичай працює дуже добре.	Потребує як позитивних, так і негативних прикладах. Потрібно вибрати хорошу функцію ядра. Вимагає багато пам'яті та процесор-ного часу. Є деякі чисельні проблеми стійкості під час вирішення обмеження QP

Продовження таблиці 1.1

Дерево рішень	Простий у реалізації. Потребує невеликої підготовки даних. Можливість обробляти як числові та інші типи даних. Використовує модель білої скриньки. Можливість перевірки моделі за допомогою статистичних тестів. Працює з великими даними за короткий проміжок часу.	Проблема навчання оптимального дерева рішень, як відомо, є NP-повним за декількома аспектами оптимальності і навіть для простих завдань. При створенні дерева рішень можуть вийти неоптимальні та дуже складні дерева, які погано обробляють дані. Існують завдання, які неможливо відобразити деревом рішень, тому що воно не описує її повністю.
Самоорганізовані карти	Простий у реалізації. Працює з нелінійним набором даних. Візуалізація багатовимірних даних на 1 або 2-мірному просторі робить його унікальним, особливо зменшення розмірності.	Потрібно багато часу для обчислень.
К-середніх	Низька складність	Необхідність вказівки К. Чутливі до перешкод та сторонніх точок даних. Кластери чутливі до первісного значення.

Завершення таблиці 1.1

Алгоритм нечіткої кластеризації Fuzzy C-means	Дозволяє точці даних бути у кількох кластерах.	Необхідно визначити кількість кластерів C. Необхідно визначити граничне значення учасників. Кластери чутливі до початкового завдання центроїдів.
Апроксимація	Можна легко змінити модель, щоб адаптувати до різних розподілів наборів даних. Кількість параметрів не збільшується зі збільшенням навчальних даних.	У деяких випадках спостерігається повільна збіжність.

Аналіз показав, що контрольовані методи навчання значно перевершують неконтрольовані, якщо дані не містить невідомих атак. Серед контрольованих методів, найкраща продуктивність досягається за рахунок нелінійних методів, таких як SVM, багатошаровий персептрон та методи, що базуються на правилах. Неконтрольовані методи, такі як K-Середня, SOM, і один клас SVM показують більш високу продуктивність в порівнянні з іншими методами, хоча вони показують різну ефективність виявлення всіх класів атак.

1.4 Система запобігання вторгненням

Система запобігання вторгненням – це технологія захисту мережі/загроз, яка вивчає потоки мережевого трафіку для виявлення та запобігання зловживанням уразливості. Використання вразливості зазвичай

відбувається у формі шкідливого входу в цільову програму або службу, які зловмисники використовують для переривання та отримання контролю над програмою або машиною. Після успішного експлойту зловмисник може вимкнути цільову програму (в результаті чого буде відмовлено в обслуговуванні) або може отримати доступ до всіх прав і дозволів, доступних для зламаної програми.

IPS часто знаходиться безпосередньо за брандмауером і забезпечує додатковий рівень аналізу, який негативно відбирає небезпечний вміст. На відміну від свого попередника – системи виявлення вторгнень (IDS), яка є пасивною системою, яка сканує трафік і повідомляє про загрози – IPS розміщується в мережі (на шляху прямого зв'язку між джерелом і призначенням), активно аналізуючи та вживаючи автоматизовані дії для всіх потоки трафіку, що надходять у мережу [13]. Зокрема, ці дії включають:

1 Надсилання тривоги адміністратору (як це буде видно в IDS).
2 Відкидання шкідливих пакетів.
3 Блокування трафіку з вихідної адреси.
4 Скидання підключення.

Як вбудований компонент безпеки, IPS повинен працювати ефективно, щоб уникнути погіршення продуктивності мережі. Він також повинен працювати швидко, оскільки експлойти можуть відбуватися майже в реальному часі. IPS також повинен виявляти та точно реагувати, щоб усунути загрози та помилкові спрацьовування.

Виявлення

IPS має ряд методів виявлення для пошуку експлойтів, але двома домінуючими механізмами є виявлення на основі сигнатур і виявлення аномалій на основі статистичних даних.

Виявлення на основі сигнатур базується на словнику однозначно ідентифікованих шаблонів (або підписів) у коді кожного експлойту. Коли

експлойт виявляється, його підпис записується і зберігається в словнику сигнатур, що постійно зростає.

Виявлення підписів для IPS поділяється на два типи:

1. Сигнатури, що стикаються з експлойтом, ідентифікують окремі експлойти, запускаючи унікальні шаблони конкретної спроби експлойту. IPS може ідентифікувати конкретні експлойти, знайшовши збіг із сигнатурою експлойту в потоці трафіку.

2. Сигнатури, пов'язані з уразливістю, – це ширші сигнатури, спрямовані на основну вразливість у системі, на яку спрямовано. Ці сигнатури дозволяють захищати мережі від варіантів експлойту, які, можливо, не спостерігалися безпосередньо в дикій природі, але також підвищують ризик помилкових результатів.

Виявлення аномалій бере вибірки мережевого трафіку випадковим чином і порівнює їх із попередньо розрахованим базовим рівнем продуктивності. Коли вибірка активності мережевого трафіку виходить за межі параметрів базової продуктивності, IPS вживає заходів для вирішення ситуації.

IPS спочатку був створений і випущений як окремий пристрій у середині 2000-х років. Однак це сталося з появою сучасних реалізацій, які зараз зазвичай інтегруються в рішення Unified Threat Management (UTM) і брандмауери наступного покоління [14].

Як працюють системи запобігання вторгненням?

Система запобігання вторгненню буде працювати шляхом сканування всього мережевого трафіку. Для цього інструмент IPS зазвичай розташовується безпосередньо за брандмауером, виконуючи роль додаткового рівня, який буде спостерігати за подіями на наявність шкідливого вмісту. Таким чином інструменти IPS розміщуються на шляхах прямого зв’язку між системою та мережею, що дозволяє інструменту аналізувати мережевий трафік.

Нижче наведено три поширені підходи до інструменту IPS для захисту мереж [15]:

- виявлення на основі сигнатур, при якому інструмент IPS використовує раніше визначені сигнатури атак відомих мережевих загроз для виявлення загроз і вжиття заходів;

- виявлення на основі аномалій, при якому IPS шукає несподівану поведінку мережі та блокує доступ до хоста, якщо виявлено аномалію;

- виявлення на основі політики, при якому IPS спочатку вимагає від адміністраторів створення політики безпеки – коли відбувається подія, яка порушує визначену політику безпеки, системним адміністраторам надсилається сповіщення.

Якщо виявлено будь-які загрози, інструмент IPS, як правило, здатний надсилати попередження адміністратору, відкидати будь-які шкідливі мережеві пакети та скинути з’єднання шляхом переналаштування брандмауерів, переупакування корисних даних та видалення заражених вкладень із серверів.

Інструменти IPS можуть допомогти відбити атаки відмови в обслуговуванні (DoS), розподілені атаки відмови в обслуговуванні (DDoS), хробаки, віруси або експлойти, такі як експлойт нульового дня. За словами Майкла Ріда, колишнього співробітника Top Layer Networks (придбаного Corero), ефективна система запобігання вторгненню повинна виконувати більш складний моніторинг та аналіз, наприклад, спостерігати і реагувати на шаблони трафіку, а також окремі пакети. "Механізми виявлення можуть включати відповідність адрес, відповідність рядків і підрядків HTTP, відповідність загальним шаблонам, аналіз з'єднання TCP, виявлення аномалії пакетів, виявлення аномалії трафіку та відповідність порту TCP/UDP [16].

1.5 Види систем запобігання вторгненням

Зазвичай зустрічаються три типи систем запобігання вторгненню. Це такі типи [17]:

1) аналіз поведінки мережі (NBA), який аналізує поведінку мережі на предмет ненормального потоку трафіку – зазвичай використовується для виявлення атак DDoS;

2) мережева система запобігання вторгненню (NIPS), яка аналізує мережу для пошуку підозрілого трафіку - зазвичай оточуючих протоколів;

Системи запобігання вторгненням на основі хоста (HIPS), які встановлюються на одному хості та використовуються для аналізу підозрілої активності на одному конкретному хості.

Крім того, існують інші типи інструментів IPS, у тому числі ті, які аналізують бездротові мережі. Проте в загальних рисах можна сказати, що система запобігання вторгненню включає будь-який продукт або практику, що використовується для запобігання зловживанням доступу до вашої мережі, наприклад брандмауери та антивірусне програмне забезпечення.

Переваги систем запобігання проникненню включають наступне:

- зниження ймовірності інцидентів безпеки;
- забезпечення динамічного захисту від загроз;
- автоматичне сповіщення адміністраторів при виявленні підозрілої активності;
- пом'якшення атак, таких як загрози нульового дня, DoS-атаки, DDoS-атаки та спроби атак грубої сили;
- скорочення обслуговування мереж для IT-персоналу;
- дозволяти або забороняти певний вхідний трафік до мережі.

До недоліків систем запобігання вторгненню можна віднести наступне:

Коли система блокує ненормальну активність у мережі, припускаючи, що вона є зловмисною, це може бути помилково позитивним і призвести до DoS для законного користувача.

Якщо організація не має достатньої пропускної здатності та потужності мережі, інструмент IPS може уповільнити роботу системи.

Якщо в мережі є кілька IPS, дані повинні пройти через кожен, щоб досягти кінцевого користувача, що спричинить втрату продуктивності мережі.

IPS також може бути дорогим.

2 НАБІР ІНСТРУМЕНТІВ ТА НАБОРИ ДАНИХ

2.1 Багатовимірні дані

Python Outlier Detection (PyOD): PyOD – це всеосяжний та масштабований набір інструментів Python для виявлення віддалених об'єктів у багатовимірних даних. Він містить понад 20 алгоритмів виявлення, включаючи нові моделі глибокого навчання та ансамблі викидів [18].

Python Streaming Anomaly Detection (PySAD): PySAD – це середовище виявлення аномалій потокової передачі на Python, що надає повний набір інструментів для експериментів з виявлення аномалій. В даний час він містить понад 15 онлайн-алгоритмів виявлення аномалій і 2 різних методи для інтеграції детекторів PyOD в налаштування потокової передачі [19].

Scikit-learn. Виявлення новинок та викидів. Він підтримує деякі популярні алгоритми, такі як LOF, Isolation Forest та One-class SVM.

Багато додатків вимагають можливості вирішити, чи належить нове спостереження до того самого розподілу, що й існуючі спостереження (це внутрішнє), чи його слід розглядати як різне (це виброс). Часто ця здатність використовується для очищення реальних наборів даних.

Виявлення викидів і виявлення новизни використовуються для виявлення аномалій, коли людина зацікавлена у виявленні аномальних або незвичайних спостережень. Виявлення викидів також відоме як неконтрольоване виявлення аномалій, а виявлення новизни – як напівконтрольоване виявлення аномалії. У контексті виявлення викидів, викиди/аномалії не можуть утворювати щільний кластер, оскільки наявні оцінювачі припускають, що викиди/аномалії розташовані в областях з низькою щільністю. Навпаки, в контексті виявлення новизни новинки/аномалії можуть утворювати щільний кластер, якщо вони

знаходяться в області низької щільності навчальних даних, що вважається нормальним у цьому контексті.

Порівняння алгоритмів виявлення викидів у scikit-learn. Локальний коефіцієнт викидів (LOF) не показує межу рішення чорним кольором, оскільки не має методу прогнозування, який слід застосувати до нових даних, коли він використовується для виявлення викидів.

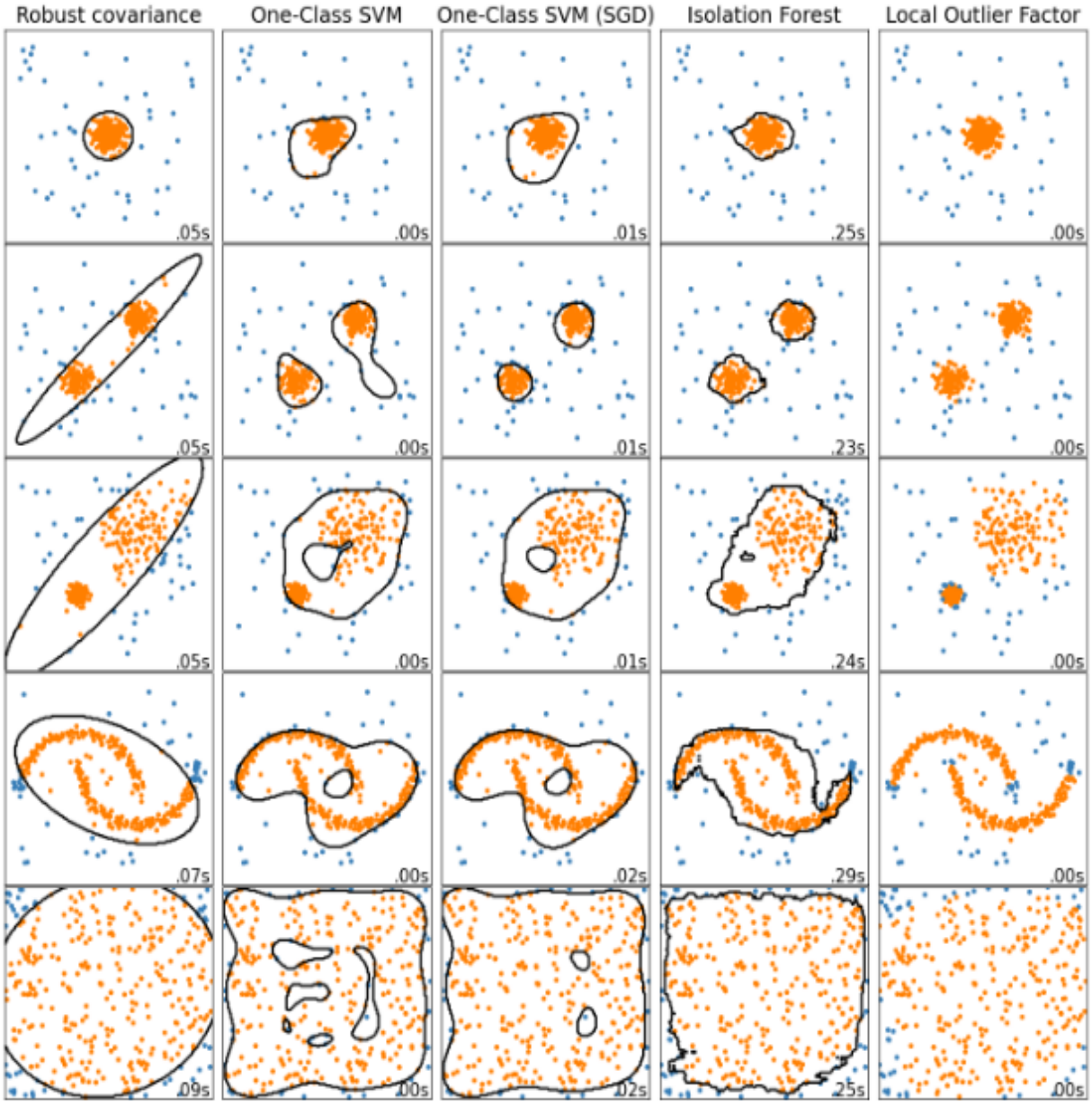


Рисунок 2.1 – Порівняння алгоритмів виявлення викидів у scikit-learn

ensemble.IsolationForest і susjeds.LocalOutlierFactor досить добре працюють із розглянутими тут наборами даних. Відомо, що svm.OneClassSVM чутливий до викидів і, отже, не дуже добре працює для виявлення викидів. При цьому виявлення викидів у великому вимірі або без будь-яких припущень щодо розподілу внутрішніх даних є дуже складним. svm.OneClassSVM все ще може використовуватися з виявленням викидів, але вимагає точного налаштування свого гіперпараметра nu для обробки викидів і запобігання переобладнанню. linear_model.SGDOneClassSVM забезпечує реалізацію лінійного однокласового SVM з лінійною складністю в кількості вибірок. Ця реалізація тут використовується з технікою апроксимації ядра для отримання результатів, подібних до svm.OneClassSVM, який за замовчуванням використовує ядро Гаусса. Нарешті, covariance.EllipticEnvelope припускає, що дані є гауссовими, і вивчає еліпс. Щоб отримати докладнішу інформацію про різні оцінювачі, зверніться до прикладу Порівняння алгоритмів виявлення аномалій для виявлення викидів у наборах даних іграшок та розділах нижче.

Масштабоване неконтрольоване виявлення викидів (SUOD) [20].

SUOD (Scalable Unsupervised Outlier Detection) – це прискорена платформа для навчання та прогнозування великомасштабних неконтрольованих викидів, заснована на PyOD. SUOD також є системою прискорення для навчання та прогнозування широкомасштабного неконтрольованого гетерогенного детектора викидів. Він зосереджується на трьох додаткових аспектах для прискорення (зменшення розмірності для високорозмірних даних, апроксимація моделі для складних моделей і підвищення ефективності виконання для дисбалансу навантаження в розподілених системах), одночасно контролюючи зниження продуктивності виявлення.

З моменту свого заснування у вересні 2019 року SUOD успішно використовується в різних академічних дослідженнях.

На рисунку 2.2 представлено загальну модель масштабованого неконтрольованого виявлення викидів.

Рисунок 2.2 –Загальна модель SUOD

SUOD призначений для:

- Уніфіковані API, детальна документація та приклади для легкого використання.

- Оптимізована продуктивність за допомогою JIT і розпаралелювання, якщо це можливо, за допомогою numba і joblib.

- Повністю сумісний з моделями в PyOD.

- Настроювані модулі та гнучкий дизайн: кожен модуль можна вмикати/вимкнути або повністю замінити користувацькими функціями.

ELKI: Середовище для розробки KDD-програм, яке підтримується індексними структурами: ELKI – це програмне забезпечення для аналізу даних з відкритим вихідним кодом (AGPLv3), написане на Java. У центрі уваги ELKI є дослідження в області алгоритмів, з акцентом на неконтрольовані методи кластерного аналізу та виявлення викидів [15].

Для досягнення високої продуктивності та масштабованості ELKI пропонує структури індексів даних, такі як R*-дерево, які можуть забезпечити значне підвищення продуктивності. ELKI розроблено так, щоб його було легко розширити для дослідників і студентів у цій області, і вітає внесок додаткових методів. ELKI має на меті надати велику колекцію алгоритмів, які дуже параметризуються, щоб забезпечити легку та справедливу оцінку та порівняльний аналіз алгоритмів.

Логотип подано на рисунку 2.3

Рисунок 2.3 – Логотип ELKI

Дослідження інтелекту даних веде до багатьох алгоритмів для подібних завдань.

Справедливе та корисне порівняння цих алгоритмів утруднене з кількох причин:

Реалізації порівняння партнерів під рукою немає.

Якщо надаються реалізації різних авторів, оцінка з точки зору ефективності є упередженою, щоб оцінити зусилля різних авторів у ефективному програмуванні замість оцінки алгоритмічних переваг.

З іншого боку, ефективні інструменти управління даними, такі як індексні структури, можуть виявляти значний вплив на завдання аналізу даних і тому корисні для широкого спектру алгоритмів.

У ELKI алгоритми інтелекту даних і завдання управління даними розділені і дозволяють проводити незалежну оцінку. Це поділ робить ELKI унікальним серед фреймворків інтелекту даних, таких як Weka або Rapidminer, і фреймворків для структур індексів, таких як GiST. У той же час ELKI відкритий для довільних типів даних, вимірювань відстані чи подібності або форматів файлів. Основним підходом є незалежність синтаксичних аналізаторів файлів або з'єднань з базою даних, типів даних, відстаней, функцій відстані та алгоритмів інтелекту. Допоміжні класи, напр. для алгебраїчних або аналітичних обчислень доступні для всіх алгоритмів на рівних умовах. Фреймворк є безкоштовним для наукового використання.

RapidMiner Anomaly Detection Extension.

Розширення Anomaly Detection для RapidMiner містить найбільш відомі неконтрольовані алгоритми виявлення аномалій, призначаючи окремі оцінки аномалій рядкам даних із наборами прикладів. Це дозволяє знаходити дані, які суттєво відрізняються від звичайних, без необхідності маркування даних [20].

Основними алгоритмами є:

- Коефіцієнт локального викиду (LOF)
- Коефіцієнт викиду на основі зв'язку (COF)
- Локальний кореляційний інтеграл (LOCI)
- Імовірність локального викиду (LoOP)
- Коефіцієнт локального викиду на основі кластерів (CBLOF).

CRAN Task View – Перегляд завдань CRAN.

Це подання завдань CRAN містить список пакетів, які можна використовувати для виявлення аномалій. Проблеми виявлення аномалій мають багато різних аспектів, і на методи виявлення може сильно вплинути те, як ми визначаємо аномалії, тип вхідних даних в алгоритм, очікуваний вихід

тощо. Це призводить до широких варіацій у формулюваннях проблеми, які необхідно розглядаються за допомогою різних аналітичних підходів.

Аномалії часто згадуються під кількома альтернативними назвами, такими як викиди, новизна, непарні значення, екстремальні значення, помилки, аберації в різних областях застосування. Ці варіанти також розглядаються для цього виду завдання.

Розвиток цього погляду на завдання є досить новим і все ще знаходиться на ранніх стадіях і тому може змінюватися.

Пакет викидів [R]: набір деяких тестів, які зазвичай використовуються для ідентифікації викидів у R [20].

Anomaly Detection Toolbox – Beta. Колекція популярних алгоритмів виявлення викидів у Matlab. Щоб забезпечити надійну та стійку систему застосування M2M/IoT, дуже важливо виявити будь-які порушення чи аномалії. Такі аномалії можуть бути викликані несправністю датчиків, зловмисними атаками, вторгненнями в мережу або незвичайними/цікавими подіями в системах віддаленого моніторингу. Різні датчики будуть генерувати аномалії з різними характеристиками. Аномалії в даних IoT різних характеристик вимагають різних типів алгоритмів виявлення аномалій.

2.2 Виявлення викидів часових рядів

TODS (Automated Time-series Outlier Detection System) – це повноцінна автоматизована система машинного навчання для виявлення викидів на багатовимірних даних часових рядів [21].

TODS надає вичерпні модулі для побудови систем виявлення викидів на основі машинного навчання, включаючи: обробку даних, обробку часових рядів, аналіз ознак (вилучення), алгоритми виявлення та модуль підкріплення. Функціональні можливості, що надаються за допомогою цих модулів, включають попередню обробку даних для загальних цілей,

згладжування/перетворення даних часових рядів, вилучення функцій із часових/частотних областей, різні алгоритми виявлення та залучення людського досвіду для калібрування системи. Можна виконати три поширені сценарії виявлення викидів на даних часових рядів: поточкове (часові моменти як викиди), виявлення за зразком (підпослідовності як викиди) і системне виявлення (набори часових рядів як викиди) і широкий спектр відповідних алгоритмів надається в TODS.

TODS призначений для:

- Систем машинного навчання повного стеке, що підтримує вичерпні компоненти попередньої обробки, вилучення функцій, алгоритми виявлення, а також інтерфейс «людина в циклі».

- Широкого діапазону алгоритмів, включаючи всі алгоритми точкового виявлення, які підтримуються PyOD, найсучасніші алгоритми виявлення шаблонів (колективних), такі як DeepLog, Telemanon, а також різноманітні алгоритми ансамблю для виконання системно виявлення.

Автоматизоване машинне навчання має на меті забезпечити процес без знань, який побудує оптимальний конвеєр на основі наданих даних шляхом автоматичного пошуку найкращої комбінації з усіх існуючих модулів.

Skyline – це система виявлення аномалій у реальному часі, аналізу часових рядів та моніторингу продуктивності, створена для забезпечення пасивного моніторингу показників без необхідності налаштовувати модель/порогові значення для кожного з них [22]. Він розроблений для використання скрізь, де існує велика кількість часових рядів з високою роздільною здатністю, які потребують постійного моніторингу. Після налаштування потоку метрик додаткові показники автоматично додаються до Skyline для аналізу. Алгоритми Skyline намагаються автоматично визначити, що означає аномальний показник кожного показника. Після налаштування та запуску Skyline дозволяє користувачеві навчати його тому, що не є аномальним за метрикою.

Banpei – це пакет Python для виявлення аномалій [23]. Виявлення аномалій – це техніка, яка використовується для виявлення незвичайних моделей, які не відповідають очікуваній поведінці.

Telemanom: структура для використання LSTM для виявлення аномалій у багатовимірних даних часових рядів [24].

Telemanom використовує звичайні LSTM за допомогою Keras/Tensorflow для виявлення аномалій у багатоваріантних даних датчиків. LSTM навчаються вивчати нормальну поведінку системи з використанням закодованої командної інформації та попередніх значень телеметрії. Прогнози генеруються на кожному кроці часу, а помилки в прогнозах представляють відхилення від очікуваної поведінки. Потім Telemanom використовує новий непараметричний підхід без нагляду для визначення порогу цих помилок та визначення аномальних послідовностей помилок.

DeepADoTS: конвеєр порівняльного аналізу для виявлення аномалій у даних часових рядів для кількох найсучасніших методів глибокого навчання [25].

Прийнято дотримуватись API scikit-learn, пропонуючи методи інтерфейсу fit(X) і predict(X). Перший оцінює розподіл даних без нагляду, а другий повертає оцінку аномалії для кожного екземпляра – чим вище, тим більш впевнена модель, що екземпляр є аномалією. Для порівняння ефективності методів використовуються значення ROC AUC. Також використовуємо MNIST для демонстрації використання моделі, оскільки вона вже доступна в TensorFlow і не вимагає завантаження зовнішніх даних, навіть якщо дані не мають тимчасового аспекту. Отримуємо графік, використовуючи методи інтерфейсу fit(X) і predict(X), що представлено на рисунку 2.4.

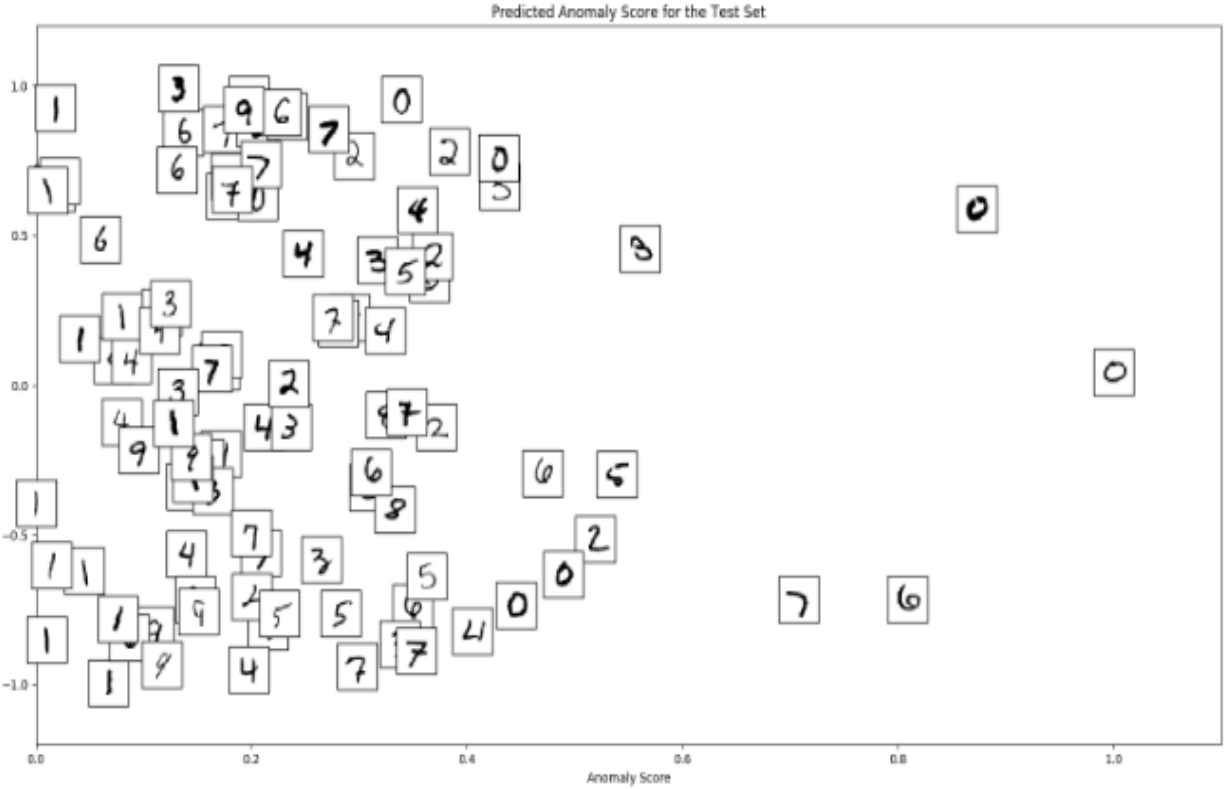


Рисунок 2.4 – Графік оцінок аномалій

Можна побачити, що глобальні викиди (нулі) і локальні викиди (дивно написані цифри) отримують високі оцінки аномалій.

NAB: Numenta Anomaly Benchmark – це новий тест для оцінки алгоритмів виявлення аномалій у потокових програмах у реальному часі [26].

NAB – це новий еталон для оцінки алгоритмів виявлення аномалій у потокових програмах реального часу. Він складається з понад 50 позначених файлів даних із реальних і штучних часових рядів, а також з нового механізму оцінки, розробленого для додатків у реальному часі.

Включені інструменти, які дозволяють запускати NAB на власних алгоритмах виявлення аномалій. Неконтрольоване виявлення аномалій у режимі реального часу для потокових даних – основний документ, що охоплює алгоритм виявлення аномалій на основі NAB та Numenta на основі HTM.

AnomalyDetection – це пакет R з відкритим кодом для виявлення аномалій, який є надійним зі статистичної точки зору за наявності сезонності та основної тенденції [27]. Пакет AnomalyDetection можна використовувати в найрізноманітніших контекстах. Наприклад, виявлення аномалій у системних показниках після випуску нового програмного забезпечення, залучення користувачів після A/B-тесту або для проблем в економетриці, фінансовій інженерії, політичних та соціальних науках.

Основний алгоритм, який називається сезонним гібридним електростатичним розрядом (S-H-ESD), заснований на узагальненому тесті ESD для виявлення аномалій. Зверніть увагу, що S-H-ESD можна використовувати для виявлення як глобальних, так і локальних аномалій. Це досягається за допомогою декомпозиції часових рядів і використання надійних статистичних показників, а саме медіани разом з ESD. Крім того, для довгих часових рядів (скажімо, 6 місяців щохвилинних даних) алгоритм використовує кускове наближення – це пов'язано з тим, що виділення тенденцій за наявності аномалій у нетривіальних – для виявлення аномалій.

Крім часових рядів, пакет також можна використовувати для виявлення аномалій у векторі числових значень. Ми вважали це дуже корисним, оскільки багато разів відповідні позначки часу недоступні. Пакет забезпечує багату підтримку візуалізації. Користувач може вказати напрямок аномалій, вікно, яке цікавить (наприклад, останній день, остання година), увімкнути/вимкнути покускове наближення; крім того, вісь x і y анотовані таким чином, щоб полегшити візуальний аналіз даних.

Пакет «anomalize» забезпечує «охайний» робочий процес для виявлення аномалій у даних [28]. Основними функціями є time_decompose(), anomalize() і time_recompose(). У комбінації досить просто розкласти часові ряди, виявити аномалії та створити діапазони, що відокремлюють «нормальні» дані від аномальних даних у масштабі (тобто для кількох часових рядів). Розкладання

часових рядів використовується для видалення тенденційних і сезонних компонентів за допомогою функції time_decompose(), а методи включають сезонне розкладання часових рядів за Лессом ("stl") і сезонне розкладання за кусковими медіанами ("twitter"). Функція anomalize() реалізує два методи для виявлення аномалій залишків, включаючи використання внутрішнього квартильного діапазону ("iqr") і узагальненого екстремального студентизованого відхилення ("gesd"). Ці методи засновані на тих, що використовуються в пакеті «прогноз» і пакеті «AnomalyDetection» Twitter. Зверніться до пов’язаних функцій, щоб отримати конкретні посилання на ці методи.

2.3 Виявлення аномалій в режимі реального часу

Виявлення аномалій у реальному часі у Open Distro для Elasticsearch від Amazon: плагіни для виявлення аномалій на основі машинного навчання для Open Distro для Elasticsearch. [29]. Функція включає в себе гарне поєднання алгоритмів машинного навчання, методів статистики, роботи систем, візуалізації та інтерфейсу користувача, а також корпоративних примітивів (для роботи з аномаліями).

Дані Analytics продовжують зростати в геометричній прогресії з часом. Експоненційний сплеск даних зменшує використання традиційного аналітичного робочого процесу користувача, який представляє собою набір стандартних запитів і панелей інструментів. Це пов’язано з тим, що для виконання запитів потрібно відстежувати зміни даних і розподіл кожного поля з часом, чого важче досягти, коли обсяги даних значно збільшуються; неоднорідність (наприклад, поведінка атаки в слідах безпеки) ускладнює розуміння самих даних.

Необхідно створити набір аналітичних функцій у реальному часі для Elasticsearch, що полегшить користувачам Open Distro for Elasticsearch

автоматичне визначення шаблонів у реальному часі для потоків даних під час надходження. Бажано надати користувачам інтерактивне та кероване дослідження даних, не змушуючи їх турбуватися про налаштування «чорного ящика» аналітики, який включає моделі, гіперпараметри та мітки. Розпочато створення та випуск функції виявлення аномалій як відкритого дистрибутива для Elasticsearch. Планується розглянути основні аспекти, на яких будується виявлення аномалій: алгоритми машинного навчання Random Cut Forest (RCF), що лежать в основі виявлення, архітектуру системи та робочий процес.

datastream.io Це фреймворк з відкритим кодом для виявлення аномалій у реальному часі за допомогою Python, Elasticsearch і Kibana [30]. Його можна використовувати через командний рядок або імпортувати його у свій код Python. Ви можете візуалізувати свої потоки даних за допомогою вбудованого сервера Bokeh або ви можете перепонаправити їх у Elasticsearch та візуалізувати за допомогою Kibana. У будь-якому випадку dsio створить відповідну інформаційну панель для вашого потоку. Крім того, якщо запускається dsio через блокнот Jupyter, він вбудовує панель керування потоковим ефектом Бокех в той самий блокнот.

2.4 Набори даних

Outlier Detection DataSets (ODDS) – набори даних виявлення викидів.

У ODDS відкрито надається доступ до великої колекції наборів даних для виявлення викидів з основною правдою (якщо доступно). Зосереджуємось на тому, щоб надати набори даних з різних доменів і представити їх під єдиною парасолькою для дослідницької спільноти. Таким чином, впорядковуємо набори даних на основі їх типів у різні таблиці в певному порядку [31].

Багатовимірні набори даних точок: на кожну точку даних є один запис, і кожен запис містить кілька атрибутів.

Набори даних графіка часових рядів для виявлення подій: дані тимчасового графіка, де графік динамічно змінюється з часом, коли надходять нові вузли та ребра або зникають існуючі вузли та ребра.

Набори точок часового ряду (багатовимірні/одномірні): тимчасові дані точки, де кожна точка має один або кілька атрибутів, і атрибути змінюються з часом.

Сценарій змагання/атаки та набори даних безпеки: дані виявлення шахрайства з думкою з онлайн-системи огляду. Дані кібербезпеки, напр. виявлення вторгнень за допомогою сценаріїв DoS, DDoS тощо.

Відеодані переповненої сцени для виявлення аномалій: відеокліпи, отримані за допомогою камери.

Набори даних графіка часових рядів для виявлення подій подано у таблиці 2.1.

Набори точок часового ряду подано у таблиці 2.2.

Сценарій змагання/атаки та набори даних безпеки представлено у таблиці 2.3.

Відеодані переповненої сцени для виявлення аномалій представлено у таблиці 2.4.

Таблиця 2.1 – Набори даних графіка часових рядів для виявлення подій

Набір даних	Вузли	Тривалість	Опис
ChallengeNetwork	125	9 днів	Дані потоку мережевого трафіку імітації кібер-задачі
EnronInc	80,884	4 роки	Мережа зв'язку електронної пошти з плином часу в Enron Inc.
NYTNews	320K	7,5 років	Граф спільного згадування об’єктів для New York Times News Corpus за 7,5 років
RealityMining	9104	50 тижнів	спілкування та близькості, дані 97 викладачів, студентів і співробітників Массачусетського технологічного інституту
TwitterSecurity2014	130K	4 місяці	Мережа спільного згадування суб'єктів із Twitter, пов'язана з тероризмом та внутрішньою безпекою
TwitterWorldCup2014	54K	1 місяць	Мережа спільного згадування суб'єктів із Twitter, пов'язана з Чемпіонатом світу з футболу 2014 року
VAST2012MC2	5K	2 дні	Операційна експертиза регіонального офісу Bank of Money
VAST2013MC3	1.2K	2 тижні	Дані потоку комп’ютерної мережі Big Marketing
VAST2014	–	3 дні	з мітками часу, дані про мережу та транзакції від GAStech

Таблиця 2.2 – Набори точок часового ряду (багатомірний/ одноваріантний)

Набір даних	Тип набору даних	Розмір	Тривалість	Опис
DataMarket – TSDL	Одновимірні	Кілька наборів даних	–	Бібліотека даних часових рядів (TSDL) була створена Робом Хайндманом, професором статистики Університету Монаша, Австралія
Yahoo – контрольний набір даних для TSAD	багатомірний	від 741 до 1680 спостережень на серію з регулярними інтервалами	367 часових рядів	Цей набір даних випущений Yahoo Labs для виявлення незвичайного трафіку на серверах Yahoo.
Numenta Anomaly Benchmark (NAB)	багатомірний	Кілька наборів даних	–	еталон для виявлення аномалій потокової передачі, де використовуються дані часового ряду, надані датчиком

Таблиця 2.3 – Сценарій змагання/атаки та набори даних безпеки

Набір даних	Розмір	Опис
YelpCHI	67 395 в-в про готелі та ресторани	Огляди від Yelp.com для готелів і ресторанів Чикаго
YelpNYC	359052 відгуки про ресторани	Відгуки про ресторани Нью-Йорка від Yelp.com
YelpZip	608 598 відгуків про ресторани.	Відгуки про ресторани з поштовим індексом для Нью-Йорка, Нью-Джерсі, Коннектикуту та Пенсільванія
YelpAcademic	2,7 млн відгуків yelp	Огляди різних компаній з Yelp.com для академічного завдання.
AmazonReview	34686770 в-в про продукт	Відгуки від Amazon.com
SWMReview	1,132,373 рецензії	Набір даних SWM Review містить огляди в категорії розваги з популярного ПЗ
BeerAdvocate	1586259 від-гуків про пиво	Відгуки про пиво від BeerAdvocate
RateBeer	2924127 від-гуків про пиво	Відгуки про пиво від RateBeer
CellarTracker	2025995 відгуків про вино	Відгуки про вино від CellarTracker
FineFoods	568 454 відгуки про їжу	Огляди їжі від Amazon
Фільми	7911684 огляди фільмів	Огляди фільмів від Amazon
Виявлення вторгнень DARPA	Кілька наборів даних	Група кіберсистем і технологій лабораторії Лінкольна MTI під спонсорством DARPA ITO та AFRL/SNHS зібрала та розповсюдила перші стандартні набори даних для виявлення вторгнень

Таблиця 2.4 – Відеодані переповненої сцени для виявлення аномалій

Набір даних	Розмір	Опис
Набір даних виявлення аномалій UCSD	98 відеокліпів	Анотований набір даних для виявлення аномалій UCSD був отриманий за допомогою стаціонарної камери, встановленої на висоті з видом на пішохідні доріжки.
Набори даних про активність натовпу Університету Міннесоти	Кілька наборів даних	Дані для моніторингу людської діяльності Університетом Міннесоти
Набір даних про аномальну поведінку	Кілька наборів даних	Набори даних для виявлення аномальної поведінки у відео
Набір відеоданих Virat	~8,5 годин відео	Це дані відеоспостереження для виявлення людської активності/подій.
Дані виявлення домінуючих і рідкісних подій Університету Макгілла	3 відеокліпи (43, 96 хвилин)	Це дані відеоспостереження для виявлення домінуючих і рідкісних подій, зняті камерами зі станції метро.

Розмір даних для виявлення аномалій без нагляду

Ці набори даних можна використовувати для порівняльного аналізу неконтрольованих алгоритмів виявлення аномалій (наприклад, «Коефіцієнт локального викиду» LOF). Набори даних були отримані з кількох джерел і в основному засновані на наборах даних, які спочатку використовувалися для контрольованого машинного навчання. Публікуючи ці модифікації, тепер

можливе порівняння різних алгоритмів для неконтрольованого виявлення аномалій [32].

2.5 Контрольні показники мета-аналізу виявлення аномалій

Щоб їх знайти спочатку визначаємо підходи до порівняльного аналізу алгоритмів виявлення аномалій у літературі та створимо великий корпус контрольних показників виявлення аномалій, які відрізняються за своєю конструкцією в кількох аспектах, які вважаються важливими для реальних додатків:

a) точкова складність;

b) відносна частота аномалій;

c) групованість аномалій;

d) релевантність ознак.

Застосуємо репрезентативний набір алгоритмів виявлення аномалій до цього корпусу, що дає дуже велику колекцію експериментальних результатів [33]. Проаналізуємо ці результати, щоб зрозуміти багато явищ, які спостерігалися в попередній роботі. Спочатку спостерігається вплив планування експерименту на результати експерименту. Далі результати оцінюються за допомогою двох показників: ROC Area Under the Curve і Average Precision. Використовуємо перевірку статистичних гіпотез, щоб продемонструвати цінність (або її відсутність) наших контрольних показників. Потім пропонуємо кілька підходів до узагальнення наших експериментальних результатів, роблячи кілька висновків про вплив нашої методології, а також про сильні та слабкі сторони деяких алгоритмів. Нарешті, порівнюємо результати з тривіальним рішенням як альтернативним засобом нормалізації звітної продуктивності алгоритмів. Передбачуваних внесків цієї статті багато; на додаток до надання великого загальнодоступного корпусу контрольних показників виявлення аномалій, надаємо онтологію для опису контекстів

виявлення аномалій, методологію для контролю за різними аспектами створення контрольних показників, рекомендації щодо майбутнього експериментального дизайну та обговорення багатьох потенційних підводних каменів спроб вимірювати успіх у цій сфері.

Тест аномалії Сколтеху (SKAB)

Пропонуємо Skoltech Anomaly Benchmark (SKAB), призначений для оцінки алгоритмів виявлення аномалій. SKAB дозволяє працювати з двома основними проблемами (є дві розмітки для аномалій) [34]:

Виявлення викидів – аномалії розглядаються і позначаються як одноточкові аномалії;

Виявлення точки зміни – аномалії розглядаються і позначаються як колективні аномалії.

SKAB складається з наступних артефактів:

- набори даних;
- таблиці лідерів для виявлення проблем та виявлення точок змін;
- модулі Python для оцінки алгоритмів;
- блокноти Python з алгоритмами виявлення аномалій.

Випробувальна система IIot знаходиться в Сколковському науково-технічному інституті (Сколтех). Усі подробиці про тестовий стенд і процес експерименту представлені в наступних артефактах:

- Позиційний документ (на даний момент подано до публікації).

- Слайди про проект.

- Набори даних.

Усі результати (крім розривів і CPDE) розраховуються для готових алгоритмів без будь-якої настройки гіперпараметрів.

3 ВИЯВЛЕННЯ АНОМАЛІЙ У МЕРЕЖАХ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

3.1 Попередня обробка

Для здійснення обробки даних необхідно запустити файл preprocessing.ipynb за допомогою jupyter notebook (Див. додаток А preprocessing.ipynb). Щоб ця програма працювала, файли набору даних CIC-IDS2017 мають бути в папці «CSV» у тому самому місці, що й програма. В результаті виконання цього файлу створюється файл з назвою «all_data.csv». Цей файл є необхідною умовою для роботи подальшої реалізації програми та відтворення кроків.

Наступним кроком відтвореня та результатів проєкту є запуск файлу statistics.ipynb (Див. додаток А statistics.ipynb). Ця програма перевіряє файл "all_data.csv" і друкує на цьому екрані статистику атак і доброякісного реєстру. Це не є обов’язковою умовою для будь-якого файлу. Це лише дає інформацію. Після компіляції коду отримаємо наступні результати, що представлені на рисунках 3.1 – 3.5.

```
BENIGN                        2359289
DoS Hulk                       231073
PortScan                       158930
DDoS                            41835
DoS GoldenEye                   10293
FTP-Patator                      7938
SSH-Patator                      5897
DoS slowloris                    5796
DoS Slowhttptest                 5499
Bot                              1966
Web Attack - Brute Force         1507
Web Attack - XSS                  652
Infiltration                       36
Web Attack - Sql Injection         21
Heartbleed                         11
Name: Label, dtype: int64
```

Рисунок 3.1 – Кількість виявлених атак

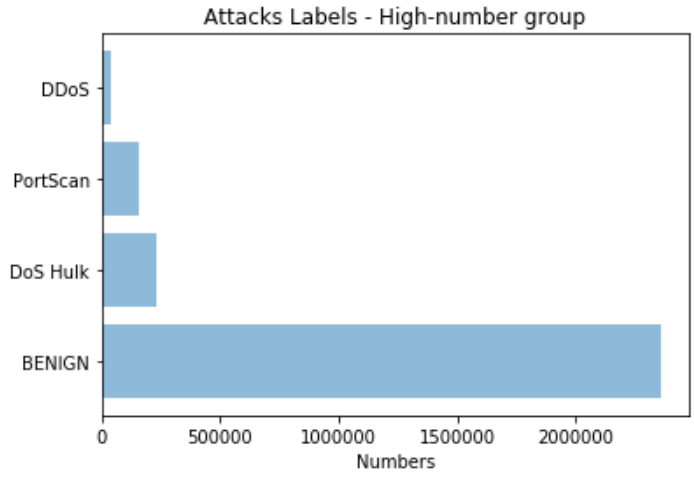


Рисунок 3.2 – Діаграма кількості атак

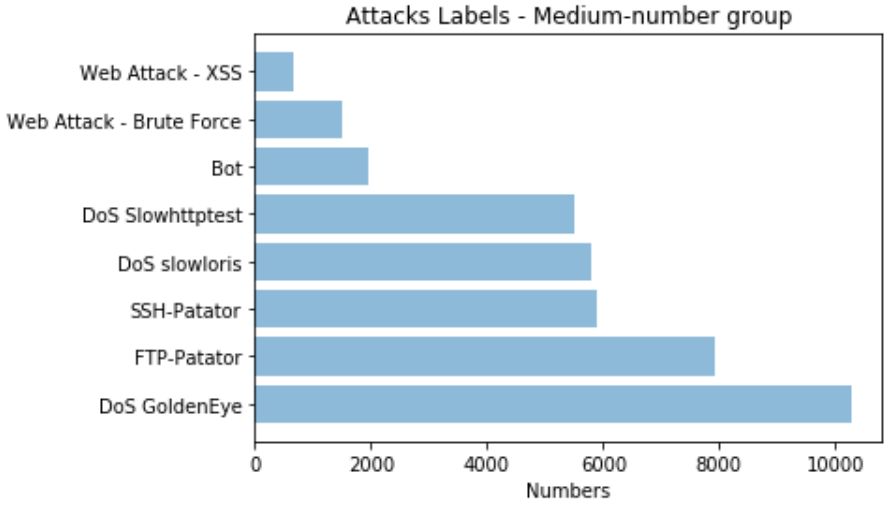


Рисунок 3.3 – Діаграма кількості атак

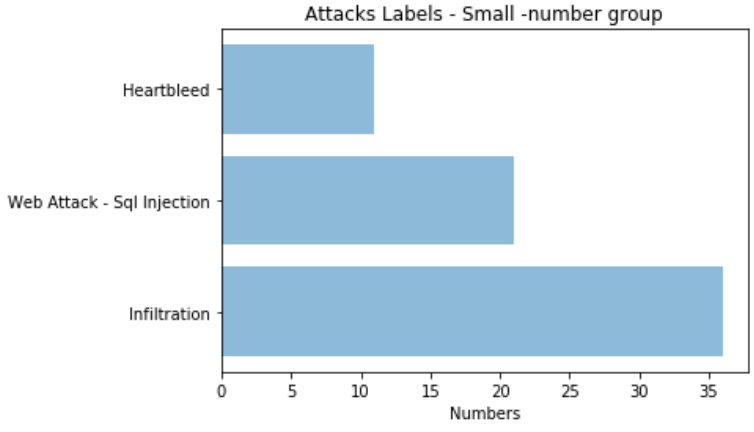
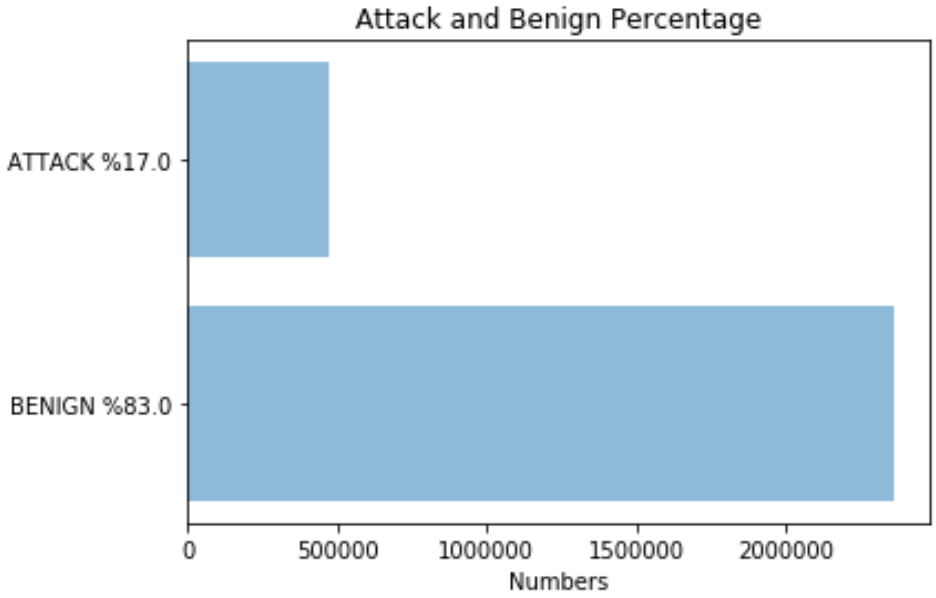


Рисунок 3.4 – Діаграма кількості атак




```
mission accomplished!
Total operation time: =  13.035051822662354 seconds
```

Рисунок 3.5 – Демонстрація найменшої та найбільшої кількості атак

Також, як результат можна побачити час виконання файлу, що складає майже 13 секунд.

3.2 Фільтрація атак

Для здійснення фільтрації атак необхідно запустити файл attack_filter.ipynb за допомогою jupyter notebook (Див. додаток А attack_filter.ipynb). Ця програма використовує файл «all_data.csv» для створення файлів атак, а потім зберігає їх у розташуванні «./attacks/». Всього набір даних містить 12 типів атак. Тому для цих атак створюється 12 файлів CSV. У кожному файлі міститься 30% атак і 70% небезпечних реєстрів. Цей крок є необхідною умовою для виконання наступних етапів. Час останнього запуску цього файлу був записаний як 304 секунди.

3.3 Вибір функції

Цей крок складається з двох файлів. Пеший відповідає за вибір функції для атаки (Див. додаток А feature_selection_for_attack_files.ipynb), другий за вибір фунцій на основі всіх даних. Ця програма використовує файли атаки, розташовані в папці «атаки». Мета цієї програми – визначити, які функції є важливими для кожної атаки. Для цього використовується алгоритм Random Forest Regressor для обчислення ваги важливості об’єктів у наборі даних. Ці набуті функції використовуються в розділі машинного навчання. В якості виведення даних на екран, він сортує його функції та ваги від великого до малого та показує їх на гістограмі (у середньому 20 атрибутів на тип атаки). На рисунках 3.6 – 3.16 показано функції для певних атак.

```
Bot
                             importance
Features
Bwd Packet Length Mean         0.338129
Flow IAT Max                   0.024407
Flow Duration                  0.008495
Flow IAT Min                   0.007067
Flow IAT Mean                  0.005609
Flow Bytes/s                   0.003518
Flow IAT Std                   0.002213
Flow Packets/s                 0.000667
Fwd Packet Length Mean         0.000418
Total Length of Bwd Packets    0.000333
Total Backward Packets         0.000223
Bwd Packet Length Max          0.000212
Fwd IAT Total                  0.000141
Bwd Packet Length Std          0.000083
Total Fwd Packets              0.000077
Fwd Packet Length Std          0.000075
Fwd Packet Length Min          0.000069
Total Length of Fwd Packets    0.000048
Fwd Packet Length Max          0.000036
Bwd Packet Length Min          0.000005
```

```
Bot=["Bwd Packet Length Mean","Flow IAT Max","Flow Duration","Flow IAT Min","Flow IAT Mean"]
```

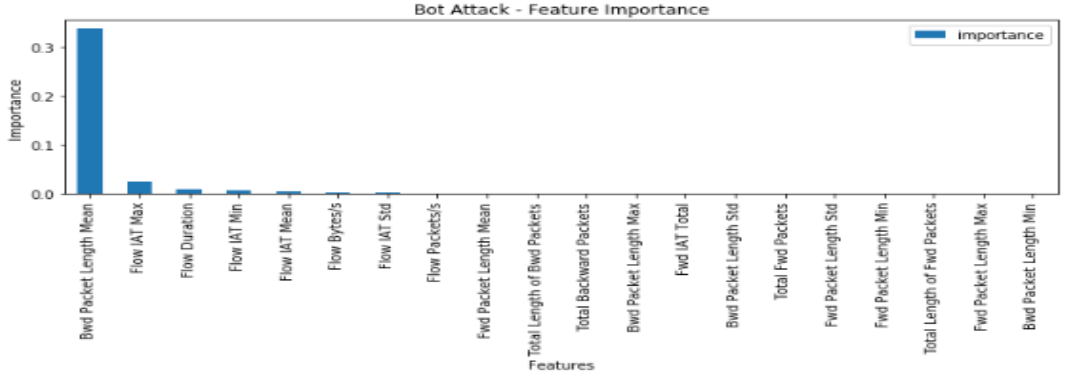


Рисунок 3.6 – Bot Attack

```
DDoS importance list:
DDoS
                             importance
Features
Bwd Packet Length Std          0.471368
Total Backward Packets         0.093576
Fwd IAT Total                  0.010827
Flow Duration                  0.006320
Flow IAT Min                   0.006110
Total Length of Fwd Packets    0.006037
Flow IAT Std                   0.005439
Flow IAT Mean                  0.005238
Flow Bytes/s                   0.004741
Flow IAT Max                   0.004703
Fwd Packet Length Max          0.001647
Fwd Packet Length Std          0.001406
Flow Packets/s                 0.001019
Fwd Packet Length Mean         0.000619
Bwd Packet Length Max          0.000577
Bwd Packet Length Min          0.000405
Bwd Packet Length Mean         0.000197
Total Length of Bwd Packets    0.000085
Total Fwd Packets              0.000042
Fwd Packet Length Min          0.000017
```

```
DDoS=["Bwd Packet Length Std","Total Backward Packets","Fwd IAT Total","Flow Duration","Flow IAT Min"]
```

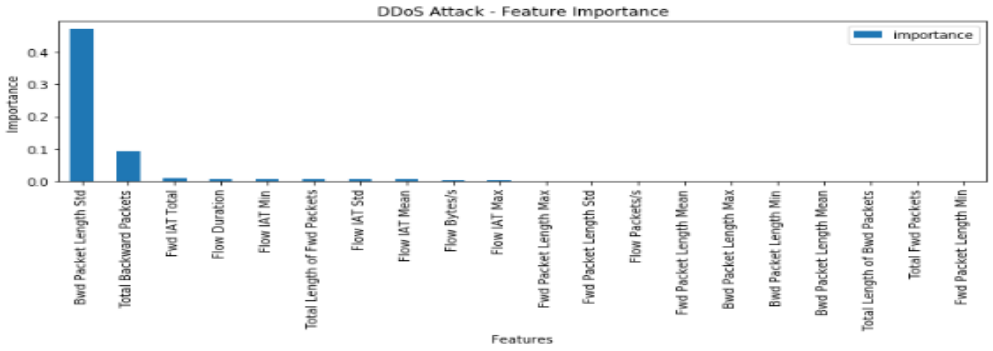


Рисунок 3.7 – DDos Attack

```
DoS GoldenEye importance list:
DoS GoldenEye
                             importance
Features
Flow IAT Max                   0.413073
Bwd Packet Length Std          0.111039
Flow IAT Min                   0.054021
Total Backward Packets         0.044895
Fwd Packet Length Min          0.009531
Flow IAT Mean                  0.004040
Flow IAT Std                   0.003863
Fwd Packet Length Max          0.002899
Bwd Packet Length Mean         0.001228
Flow Bytes/s                   0.000928
Flow Duration                  0.000927
Flow Packets/s                 0.000892
Fwd IAT Total                  0.000719
Bwd Packet Length Max          0.000445
Total Length of Fwd Packets    0.000406
Fwd Packet Length Mean         0.000393
Total Length of Bwd Packets    0.000217
Total Fwd Packets              0.000065
Bwd Packet Length Min          0.000033
Fwd Packet Length Std          0.000027
```

```
DoS GoldenEye=["Flow IAT Max","Bwd Packet Length Std","Flow IAT Min","Total Backward Packets","Fwd Packet Length Min"]
```

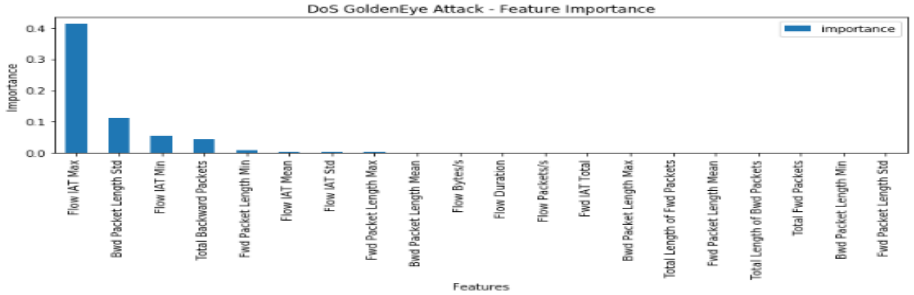


Рисунок 3.8 – Dos GoldenEye Attack

```
DoS Hulk importance list:
DoS Hulk
                              importance
Features
Bwd Packet Length Std         5.148222e-01
Fwd Packet Length Std         7.321079e-02
Fwd Packet Length Max         4.515548e-03
Flow IAT Min                  1.675778e-03
Flow Duration                 1.218072e-03
Total Backward Packets        3.813481e-04
Flow IAT Std                  2.572354e-04
Flow IAT Max                  2.517998e-04
Total Length of Bwd Packets   1.778769e-04
Fwd IAT Total                 1.739909e-04
Flow IAT Mean                 9.875828e-05
Flow Packets/s                8.114421e-05
Bwd Packet Length Mean        5.449508e-05
Flow Bytes/s                  2.752602e-05
Total Fwd Packets             1.227050e-05
Bwd Packet Length Max         1.004453e-05
Bwd Packet Length Min         9.303096e-06
Fwd Packet Length Mean        8.013636e-06
Total Length of Fwd Packets   4.604820e-06
Fwd Packet Length Min         1.810830e-08
```

```
DoS Hulk=["Bwd Packet Length Std","Fwd Packet Length Std","Fwd Packet Length Max","Flow IAT Min","Flow Duration"]
```

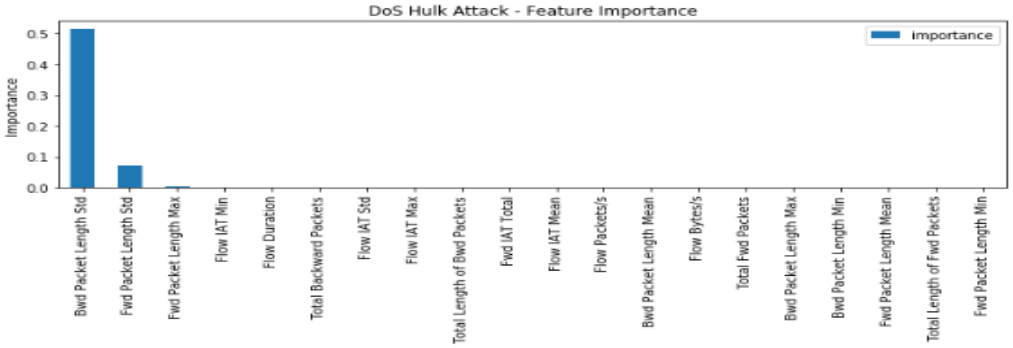


Рисунок 3.9 – Dos Hulk Attack

```
DoS Slowhttptest importance list:
DoS Slowhttptest
                              importance
Features
Flow IAT Mean                  0.653023
Fwd Packet Length Min          0.101739
Bwd Packet Length Mean         0.029976
Total Length of Bwd Packets    0.007153
Fwd Packet Length Std          0.007107
Fwd Packet Length Mean         0.006074
Bwd Packet Length Max          0.003857
Bwd Packet Length Std          0.002934
Flow IAT Min                   0.002184
Fwd Packet Length Max          0.001245
Total Fwd Packets              0.000808
Flow Duration                  0.000802
Total Length of Fwd Packets    0.000604
Bwd Packet Length Min          0.000545
Flow Bytes/s                   0.000421
Flow IAT Max                   0.000327
Fwd IAT Total                  0.000284
Flow IAT Std                   0.000252
Flow Packets/s                 0.000159
Total Backward Packets         0.000130
```

```
DoS Slowhttptest=["Flow IAT Mean","Fwd Packet Length Min","Bwd Packet Length Mean","Total Length of Bwd Packets","Fwd Packet Length Std"]
```

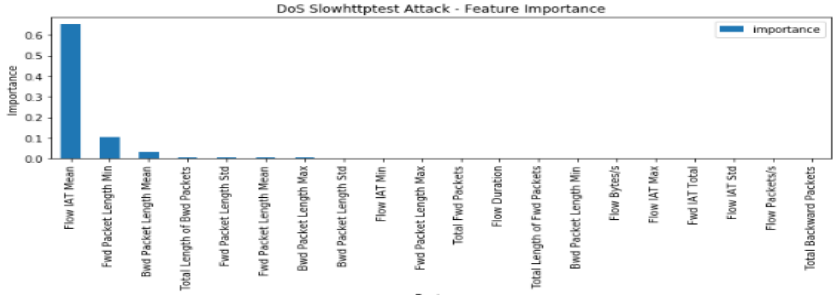


Рисунок 3.10 – Dos Slowhttptest Attack

```
FTP-Patator importance list:
FTP-Patator
                              importance
Features
Fwd Packet Length Max         1.098307e-01
Fwd Packet Length Std         2.437956e-02
Fwd Packet Length Mean        2.200624e-03
Bwd Packet Length Std         8.997715e-04
Bwd Packet Length Mean        7.081365e-04
Bwd Packet Length Max         6.054917e-04
Total Length of Bwd Packets   4.015506e-04
Flow IAT Min                  2.611238e-04
Total Length of Fwd Packets   1.766284e-04
Total Fwd Packets             1.634693e-04
Flow Duration                 1.330083e-04
Fwd IAT Total                 9.598814e-05
Flow IAT Mean                 9.187611e-05
Total Backward Packets        8.884482e-05
Flow IAT Max                  8.359459e-05
Flow Packets/s                7.030109e-05
Fwd Packet Length Min         4.619749e-05
Flow IAT Std                  2.987659e-05
Flow Bytes/s                  2.244798e-05
Bwd Packet Length Min         4.764849e-07
```

```
FTP-Patator=["Fwd Packet Length Max","Fwd Packet Length Std","Fwd Packet Length Mean","Bwd Packet Length Std","Bwd Packet Length Mean"]
```

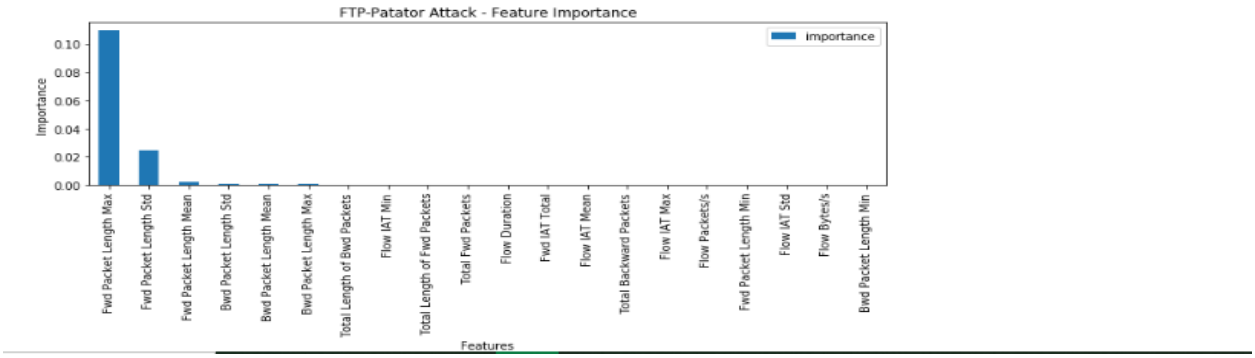


Рисунок 3.11 – FTP-Patator Attack

```
Heartbleed importance list:
Heartbleed
                              importance
Features
Total Backward Packets            0.052
Fwd Packet Length Max             0.048
Flow IAT Min                      0.048
Bwd Packet Length Max             0.048
Total Length of Fwd Packets       0.044
Total Length of Bwd Packets       0.044
Bwd Packet Length Mean            0.044
Bwd Packet Length Std             0.036
Total Fwd Packets                 0.028
Flow Duration                     0.016
Fwd IAT Total                     0.004
Fwd Packet Length Std             0.004
Fwd Packet Length Mean            0.000
Flow Bytes/s                      0.000
Flow Packets/s                    0.000
Flow IAT Mean                     0.000
Flow IAT Std                      0.000
Flow IAT Max                      0.000
Fwd Packet Length Min             0.000
Bwd Packet Length Min             0.000
```

```
Heartbleed=["Total Backward Packets","Fwd Packet Length Max","Flow IAT Min","Bwd Packet Length Max","Total Length of Fwd Packets"]
```

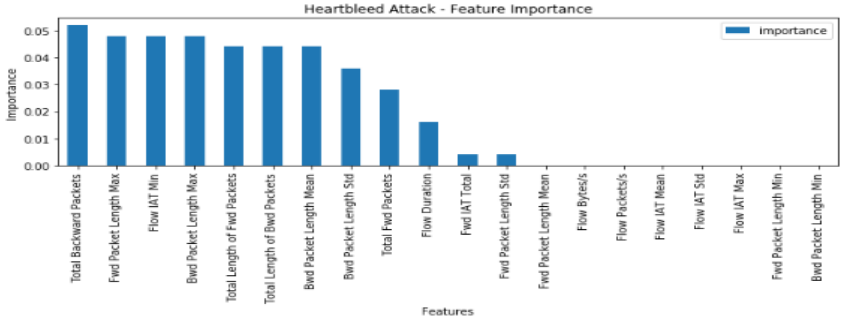


Рисунок 3.12 – Heatbleed Attack

```
Infiltration importance list:
Infiltration
                            importance
Features
Fwd Packet Length Max         0.204751
Fwd Packet Length Mean        0.163696
Flow Duration                 0.052250
Total Length of Fwd Packets   0.026823
Bwd Packet Length Mean        0.018539
Fwd Packet Length Std         0.017846
Flow IAT Max                  0.017182
Flow Bytes/s                  0.010812
Flow IAT Mean                 0.008816
Flow IAT Min                  0.008310
Fwd IAT Total                 0.007726
Flow IAT Std                  0.002862
Bwd Packet Length Max         0.002737
Bwd Packet Length Std         0.002030
Fwd Packet Length Min         0.001937
Total Fwd Packets             0.001862
Total Backward Packets        0.001680
Bwd Packet Length Min         0.001610
Flow Packets/s                0.001406
Total Length of Bwd Packets   0.000000
```

```
Infiltration=["Fwd Packet Length Max","Fwd Packet Length Mean","Flow Duration","Total Length of Fwd Packets","Bwd Packet Length Mean"]
```

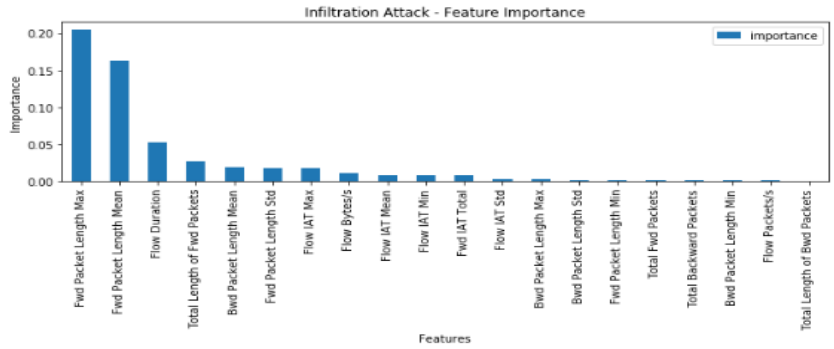


Рисунок 3.13 – Infiltration Attack

```
PortScan importance list:
PortScan
                            importance
Features
Flow Bytes/s                  0.313933
Total Length of Fwd Packets   0.304613
Fwd IAT Total                 0.000427
Flow Duration                 0.000398
Fwd Packet Length Max         0.000150
Flow IAT Max                  0.000059
Flow IAT Mean                 0.000054
Flow Packets/s                0.000031
Flow IAT Min                  0.000031
Total Length of Bwd Packets   0.000022
Total Fwd Packets             0.000021
Fwd Packet Length Mean        0.000019
Bwd Packet Length Std         0.000018
Flow IAT Std                  0.000017
Total Backward Packets        0.000014
Fwd Packet Length Std         0.000009
Bwd Packet Length Max         0.000004
Bwd Packet Length Mean        0.000002
Bwd Packet Length Min         0.000002
Fwd Packet Length Min         0.000001
```

```
PortScan=["Flow Bytes/s","Total Length of Fwd Packets","Fwd IAT Total","Flow Duration","Fwd Packet Length Max"]
```

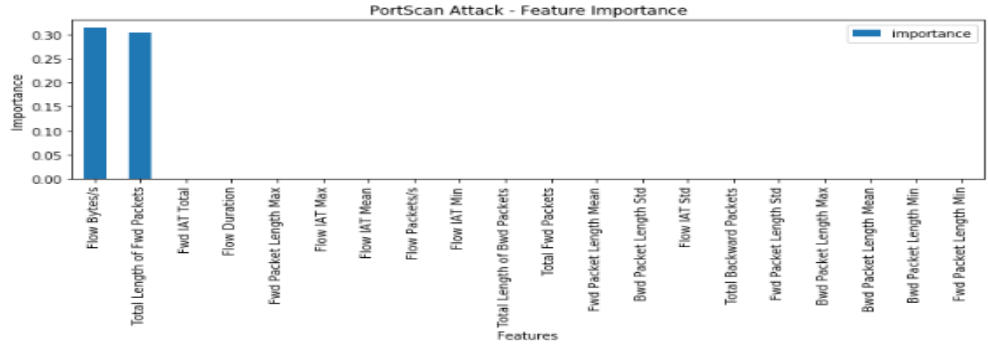


Рисунок 3.14 – PortScan Attack

```
SSH-Patator importance list:
SSH-Patator
                             importance
Features
Fwd Packet Length Max          0.000881
Flow Duration                  0.000748
Flow IAT Max                   0.000497
Total Length of Fwd Packets    0.000448
Flow IAT Mean                  0.000425
Flow Packets/s                 0.000423
Flow Bytes/s                   0.000375
Fwd IAT Total                  0.000329
Flow IAT Std                   0.000177
Fwd Packet Length Mean         0.000158
Flow IAT Min                   0.000111
Total Backward Packets         0.000100
Bwd Packet Length Min          0.000099
Fwd Packet Length Std          0.000070
Total Fwd Packets              0.000070
Fwd Packet Length Min          0.000040
Bwd Packet Length Max          0.000032
Total Length of Bwd Packets    0.000027
Bwd Packet Length Mean         0.000014
Bwd Packet Length Std          0.000008
```

```
SSH-Patator=["Fwd Packet Length Max","Flow Duration","Flow IAT Max","Total Length of Fwd Packets","Flow IAT Mean"]
```

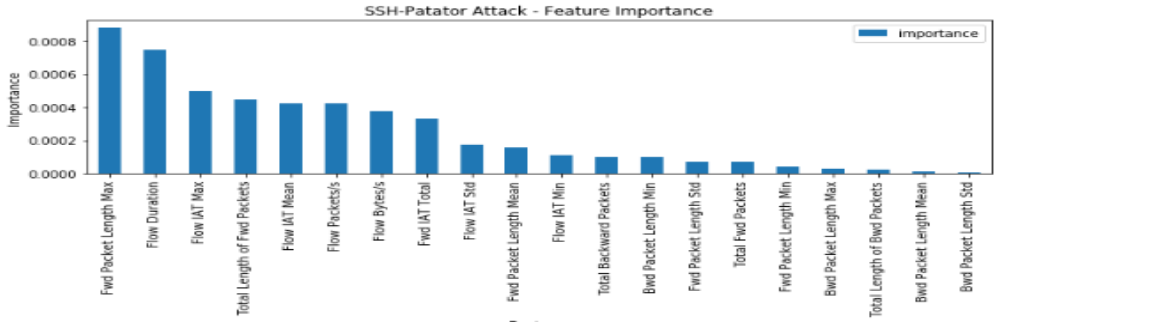
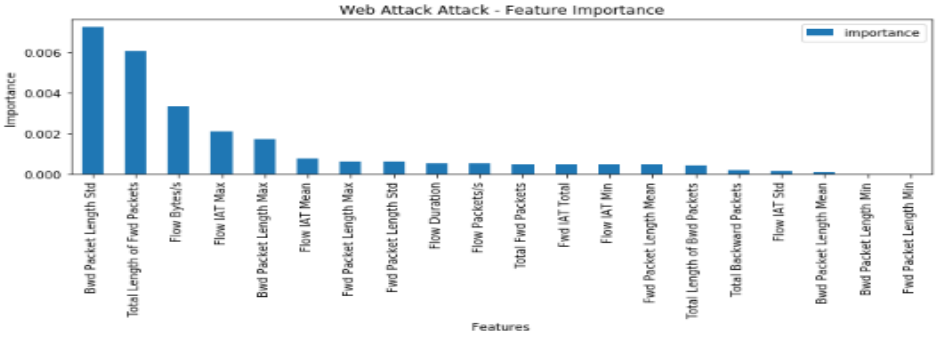


Рисунок 3.15 – SSH-Patator Attack

```
Total Length of Fwd Packets    0.006046
Flow Bytes/s                   0.003366
Flow IAT Max                   0.002102
Bwd Packet Length Max          0.001728
Flow IAT Mean                  0.000760
Fwd Packet Length Max          0.000638
Fwd Packet Length Std          0.000616
Flow Duration                  0.000541
Flow Packets/s                 0.000526
Total Fwd Packets              0.000506
Fwd IAT Total                  0.000505
Flow IAT Min                   0.000499
Fwd Packet Length Mean         0.000490
Total Length of Bwd Packets    0.000454
Total Backward Packets         0.000177
Flow IAT Std                   0.000140
Bwd Packet Length Mean         0.000102
Bwd Packet Length Min          0.000016
Fwd Packet Length Min          0.000008
```

```
Web Attack=["Bwd Packet Length Std","Total Length of Fwd Packets","Flow Bytes/s","Flow IAT Max","Bwd Packet Length Max"]
```




```
mission accomplished!
Total operation time: =  4817.430465221405 secomds
```

Рисунок 3.16 – Web Attack

Останній запуск цього файлу був записаний як 4817 секунд.

Далі, на рисунку 3.17, представлено вибір функції для всіх даних.

Ця програма застосовує дії з попереднього файлу до всього набору даних. Таким чином, він створює вагові коефіцієнти важливості функції, які дійсні для всього набору даних. Він використовує файл "all_data.csv" і алгоритм регресора випадкового лісу. Як результат виведення на екран, він сортує його функції та ваги від великого до малого та показує їх на гістограмі (всього 20 атрибутів для всіх атак).

```
all_data importance list:
all_data
                             importance
Features
Bwd Packet Length Std          0.246627
Flow Bytes/s                   0.178777
Total Length of Fwd Packets    0.102417
Fwd Packet Length Std          0.063889
Flow IAT Std                   0.009898
Flow IAT Min                   0.006946
Fwd IAT Total                  0.005121
Flow Duration                  0.004150
Bwd Packet Length Max          0.004007
Flow IAT Max                   0.003579
Flow IAT Mean                  0.003266
Total Length of Bwd Packets    0.001305
Fwd Packet Length Min          0.000670
Bwd Packet Length Mean         0.000582
Flow Packets/s                 0.000541
Fwd Packet Length Mean         0.000526
Total Backward Packets         0.000169
Total Fwd Packets              0.000138
Fwd Packet Length Max          0.000125
Bwd Packet Length Min          0.000084

all_data=["Bwd Packet Length Std","Flow Bytes/s","Total Length of Fwd Packets","Fwd Packet Length Std","Flow IAT Std"]
----------------------------------------------------------------------------------------------------

mission accomplished!
Total operation time: =  25929.417772769928 seconds
```

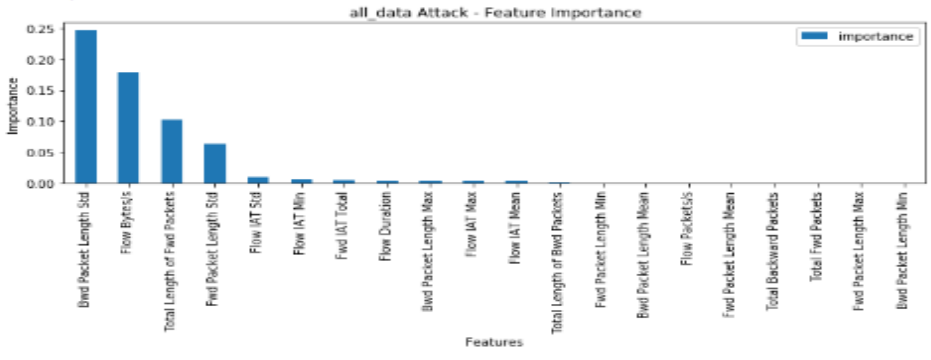


Рисунок 3.17 – Гістограма вагових коефіцієнтів важливості функції для всього набору даних

Час останнього запуску цього файлу був записаний як 25929 секунд.

4 РЕАЛІЗАЦІЯ МАШШИННОГО НАВЧАННЯ

4.1 Машинне навчання. Реалізація для файлів атаки

Ця програма використовує файли атаки з папки "./attacks/" як набір даних. Використовуються 4 функції з найбільшою вагою для кожного файлу, створені файлом feature_selection_for_attack_files. Цей файл застосовує 7 алгоритмів машинного навчання до кожного файлу 10 разів і друкує результати цих операцій на екрані та у файлі "./attacks/results_1.csv". Він також створює графічні зображення з результатами та друкує їх як на екрані, так і в папці "./attacks/result_graph_1/", також вони подані на рисунках 4.1-4.8.

File	ML algorithm	accuracy	Precision	Recall	F1-score	Time
Bot	Naive Bayes	0.55	0.82	0.55	0.54	0.0031
Bot	QDA	0.67	0.85	0.67	0.68	0.0225
Bot	Random Forest	0.96	0.96	0.96	0.96	0.0281
Bot	ID3	0.96	0.96	0.96	0.96	0.0078
Bot	AdaBoost	0.97	0.97	0.97	0.97	0.1813
Bot	MLP	0.68	0.65	0.68	0.64	0.1453
Bot	Nearest Neighbors	0.95	0.96	0.95	0.95	0.0176

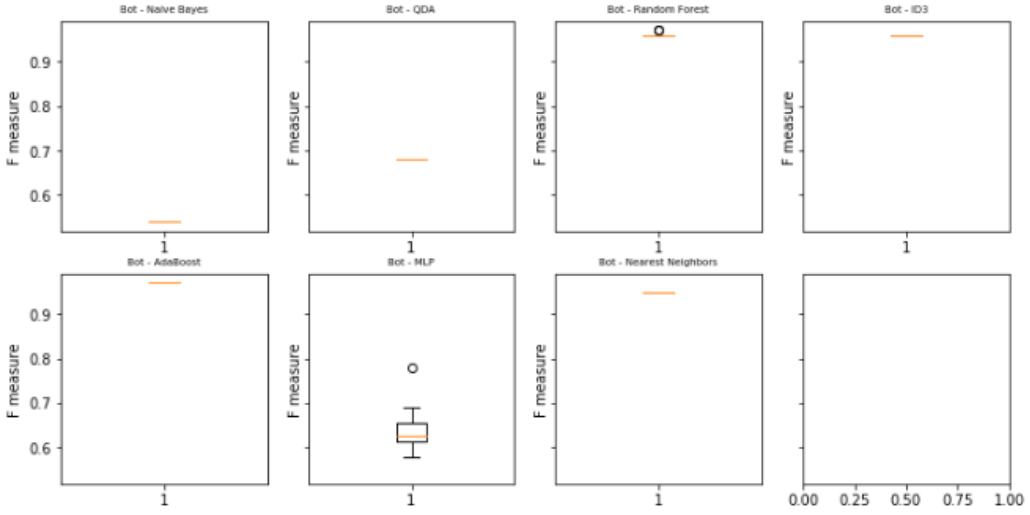


Рисунок 4.1 – Оцінка роботи алгоритмів машинного навчання для Bot

File	ML algorithm	accuracy	Precision	Recall	F1-score	Time
DDoS	Naive Bayes	0.78	0.77	0.78	0.77	0.0491
DDoS	QDA	0.41	0.8	0.41	0.34	0.0541
DDoS	Random Forest	0.96	0.96	0.96	0.96	0.4187
DDoS	ID3	0.96	0.97	0.96	0.96	0.1891
DDoS	AdaBoost	0.96	0.96	0.96	0.96	2.6724
DDoS	MLP	0.78	0.8	0.78	0.76	3.4024
DDoS	Nearest Neighbors	0.92	0.93	0.92	0.92	1.3234

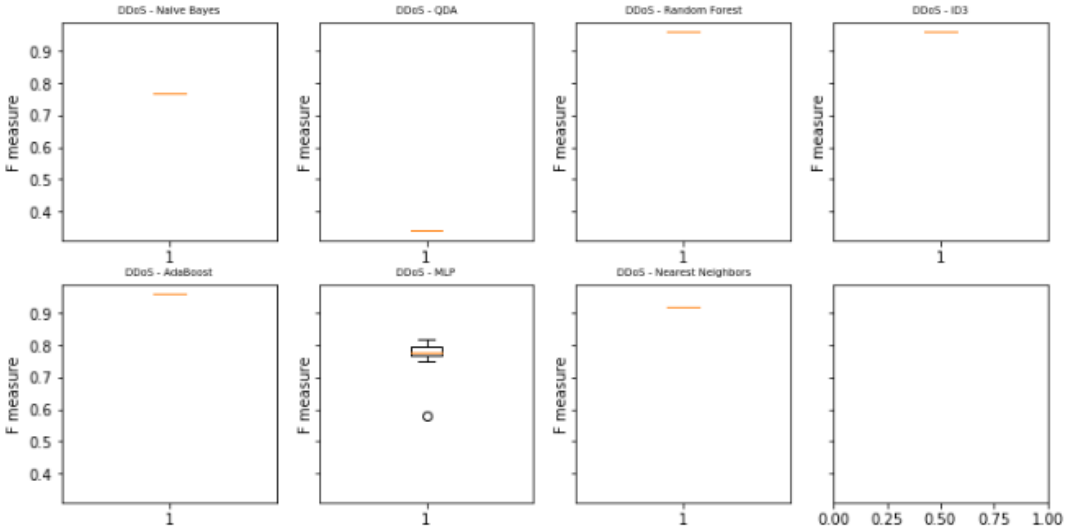


Рисунок 4.2 – Оцінка роботи алгоритмів машинного навчання для DDoS

File	ML algorithm	accuracy	Precision	Recall	F1-score	Time
DoS GoldenEye	Naive Bayes	0.83	0.82	0.83	0.81	0.0094
DoS GoldenEye	QDA	0.7	0.84	0.7	0.71	0.0141
DoS GoldenEye	Random Forest	0.99	0.99	0.99	0.99	0.0884
DoS GoldenEye	ID3	0.99	0.99	0.99	0.99	0.0345
DoS GoldenEye	AdaBoost	0.99	0.99	0.99	0.99	0.5689
DoS GoldenEye	MLP	0.65	0.75	0.65	0.64	0.6825
DoS GoldenEye	Nearest Neighbors	0.98	0.98	0.98	0.98	0.1172

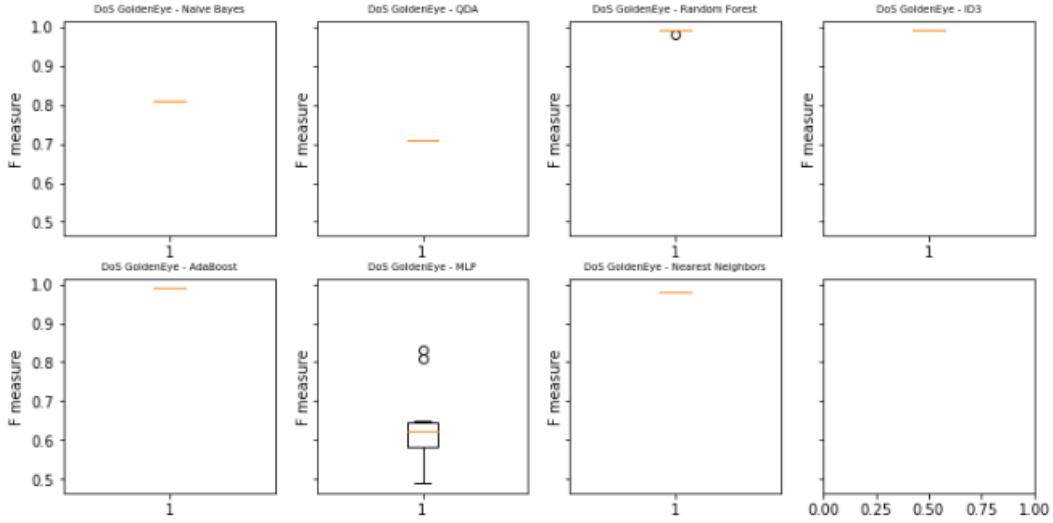


Рисунок 4.3 – Оцінка роботи алгоритмів машинного навчання для DoS

File	ML algorithm	accuracy	Precision	Recall	F1-score	Time
DoS Hulk	Naive Bayes	0.34	0.8	0.34	0.23	0.3127
DoS Hulk	QDA	0.41	0.81	0.41	0.36	0.353
DoS Hulk	Random Forest	0.93	0.94	0.93	0.93	3.4215
DoS Hulk	ID3	0.96	0.96	0.96	0.96	0.865
DoS Hulk	AdaBoost	0.96	0.96	0.96	0.96	19.8474
DoS Hulk	MLP	0.95	0.95	0.95	0.95	14.8005
DoS Hulk	Nearest Neighbors	0.96	0.96	0.96	0.96	219.6141

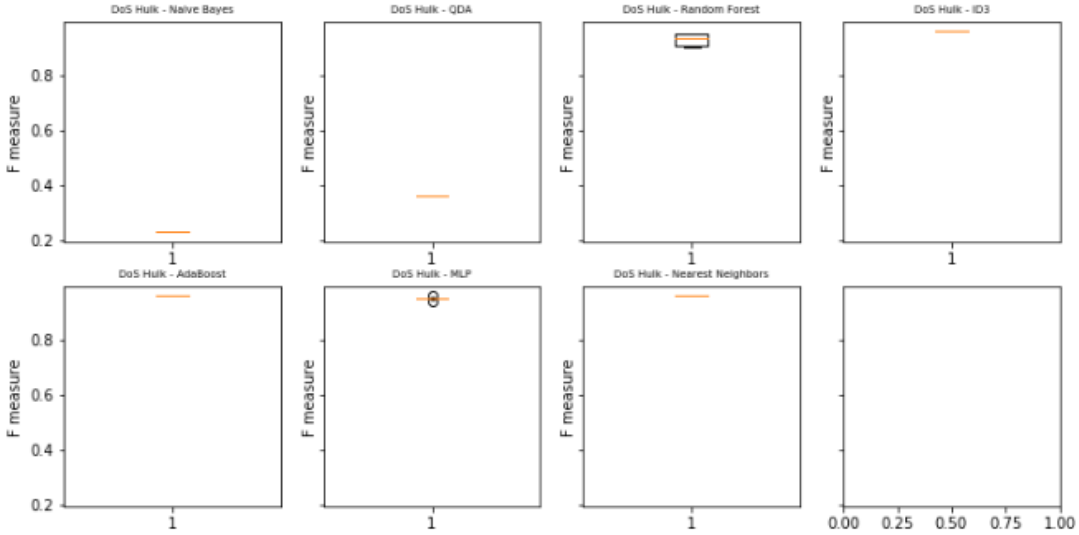


Рисунок 4.4 – Оцінка роботи алгоритмів машинного навчання для DoS Hulk

File	ML algorithm	accuracy	Precision	Recall	F1-score	Time
DoS Slowhttptest	Naive Bayes	0.41	0.7	0.41	0.35	0.0073
DoS Slowhttptest	QDA	0.43	0.72	0.43	0.38	0.0062
DoS Slowhttptest	Random Forest	0.98	0.98	0.98	0.98	0.0516
DoS Slowhttptest	ID3	0.98	0.98	0.98	0.98	0.0187
DoS Slowhttptest	AdaBoost	0.99	0.99	0.99	0.99	0.297
DoS Slowhttptest	MLP	0.78	0.84	0.78	0.78	0.3972
DoS Slowhttptest	Nearest Neighbors	0.99	0.99	0.99	0.99	0.0845

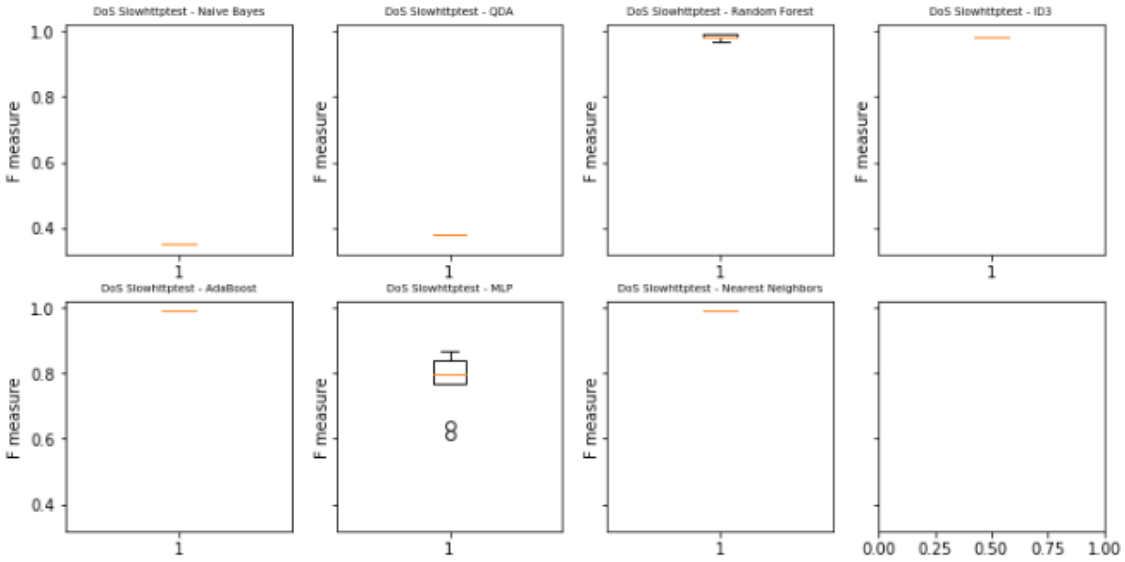


Рисунок 4.5 – Оцінка роботи алгоритмів машинного навчання для DoS Slowthhptest

File	ML algorithm	accuracy	Precision	Recall	F1-score	Time
DoS slowloris	Naive Bayes	0.42	0.8	0.42	0.37	0.0059
DoS slowloris	QDA	0.49	0.78	0.49	0.46	0.0078
DoS slowloris	Random Forest	0.95	0.95	0.95	0.95	0.0513
DoS slowloris	ID3	0.96	0.96	0.96	0.96	0.0173
DoS slowloris	AdaBoost	0.95	0.95	0.95	0.95	0.3207
DoS slowloris	MLP	0.75	0.79	0.75	0.74	0.3717
DoS slowloris	Nearest Neighbors	0.95	0.95	0.95	0.95	0.0419

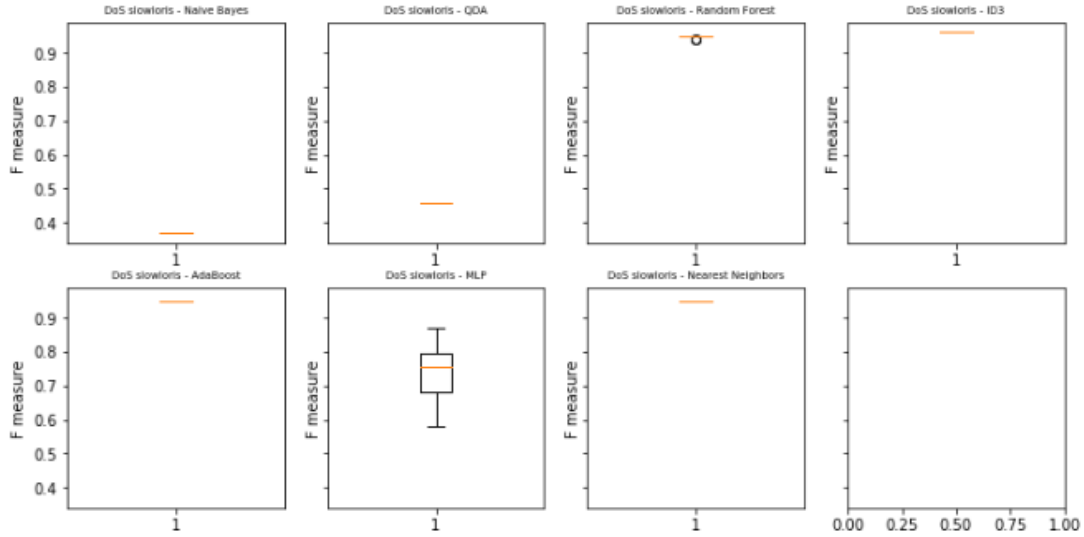


Рисунок 4.6 – Оцінка роботи алгоритмів машинного навчання для DoS Slowloris

File	ML algorithm	accuracy	Precision	Recall	F1-score	Time
FTP-Patator	Naive Bayes	1.0	1.0	1.0	1.0	0.0078
FTP-Patator	QDA	1.0	1.0	1.0	1.0	0.0083
FTP-Patator	Random Forest	1.0	1.0	1.0	1.0	0.0544
FTP-Patator	ID3	1.0	1.0	1.0	1.0	0.0156
FTP-Patator	AdaBoost	1.0	1.0	1.0	1.0	0.3859
FTP-Patator	MLP	1.0	1.0	1.0	1.0	2.2922
FTP-Patator	Nearest Neighbors	1.0	1.0	1.0	1.0	0.2547

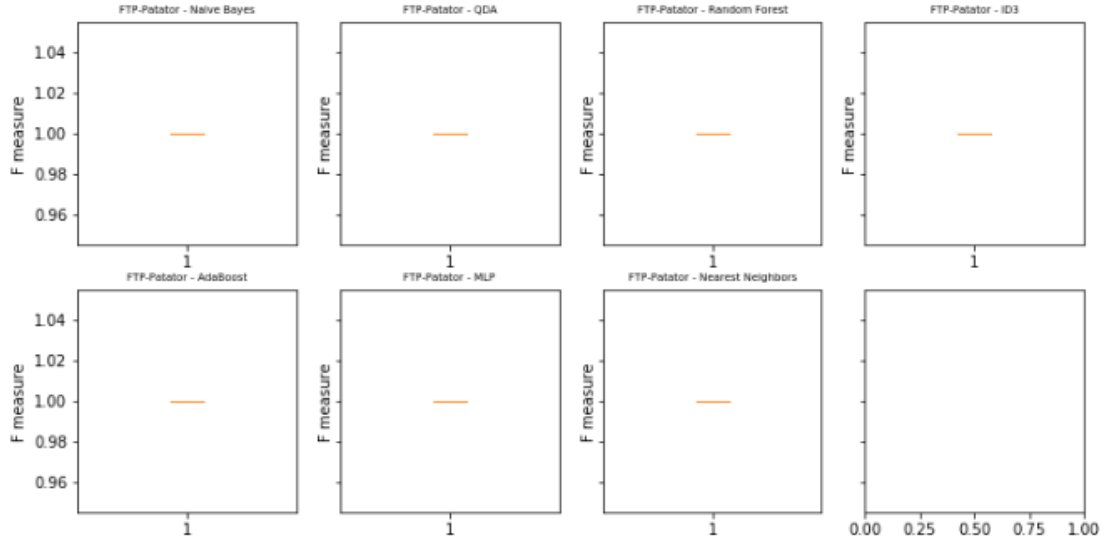
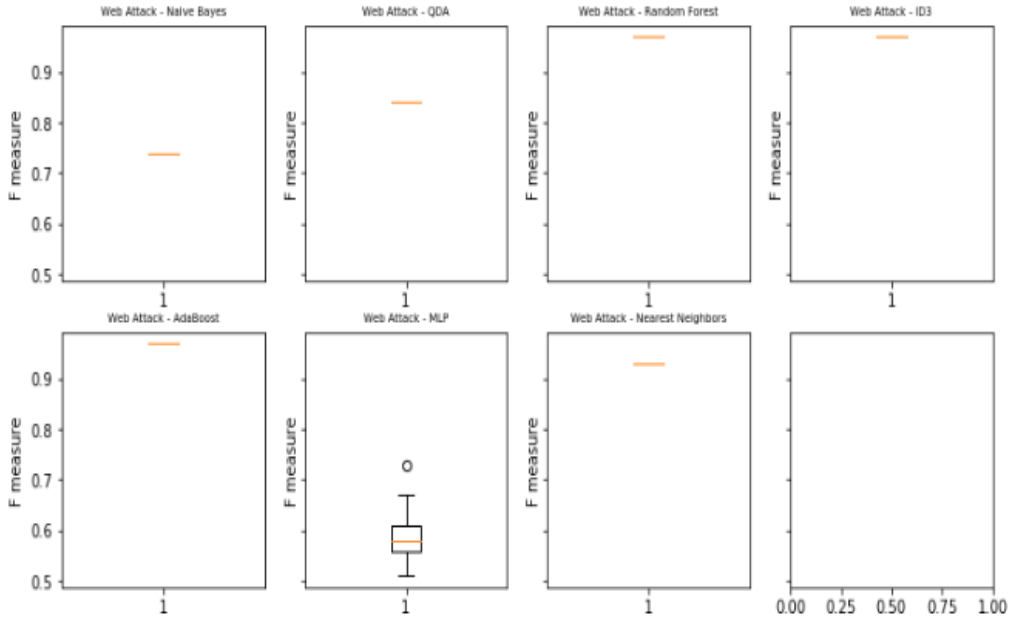


Рисунок 4.7 – Оцінка роботи алгоритмів машинного навчання для FTP-Patator

File	ML algorithm	accuracy	Precision	Recall	F1-score	Time
Web Attack	Naive Bayes	0.73	0.86	0.73	0.74	0.0044
Web Attack	QDA	0.84	0.89	0.84	0.84	0.0042
Web Attack	Random Forest	0.97	0.97	0.97	0.97	0.0317
Web Attack	ID3	0.97	0.97	0.97	0.97	0.0097
Web Attack	AdaBoost	0.97	0.97	0.97	0.97	0.1716
Web Attack	MLP	0.64	0.63	0.64	0.6	0.1182
Web Attack	Nearest Neighbors	0.93	0.94	0.93	0.93	0.0174




```
mission accomplished!
Total operation time: =  3601.525671482086 seconds
```

Рисунок 4.8 – Оцінка роботи алгоритмів машинного навчання для Web Attack

Час останнього запуску цього файлу був записаний як 3601 секунда.

4.2 Машинне навчання. Реалізація з 18 функціями

Ця програма реалізує методи машинного навчання у файлі "all_data.csv". Використовує функції, що були використані на попередньому етапі. Набір функцій, які будуть використовуватися, складається з поєднання 4 функцій з найвищою вагою, досягнутою для кожної атаки на кроці

«machine_learning_implementation_for_attack_files». Таким чином, з кожного з 12 типів атаки отримують 4 функції, в результаті чого утворюється пул функцій, що складається з 48 атрибутів. Після видалення повторів кількість ознак становить 18.

Цей файл застосовує 7 алгоритмів машинного навчання до файлу "all_data.csv" 10 разів і друкує результати цих операцій на екрані та у файлі "./attacks/results_2.csv". Він також створює графічні зображення в квадраті з результатами та друкує їх як на екрані, так і в папці "./attacks/result_graph_2/", що показано на рисунках 4.9-4.12.

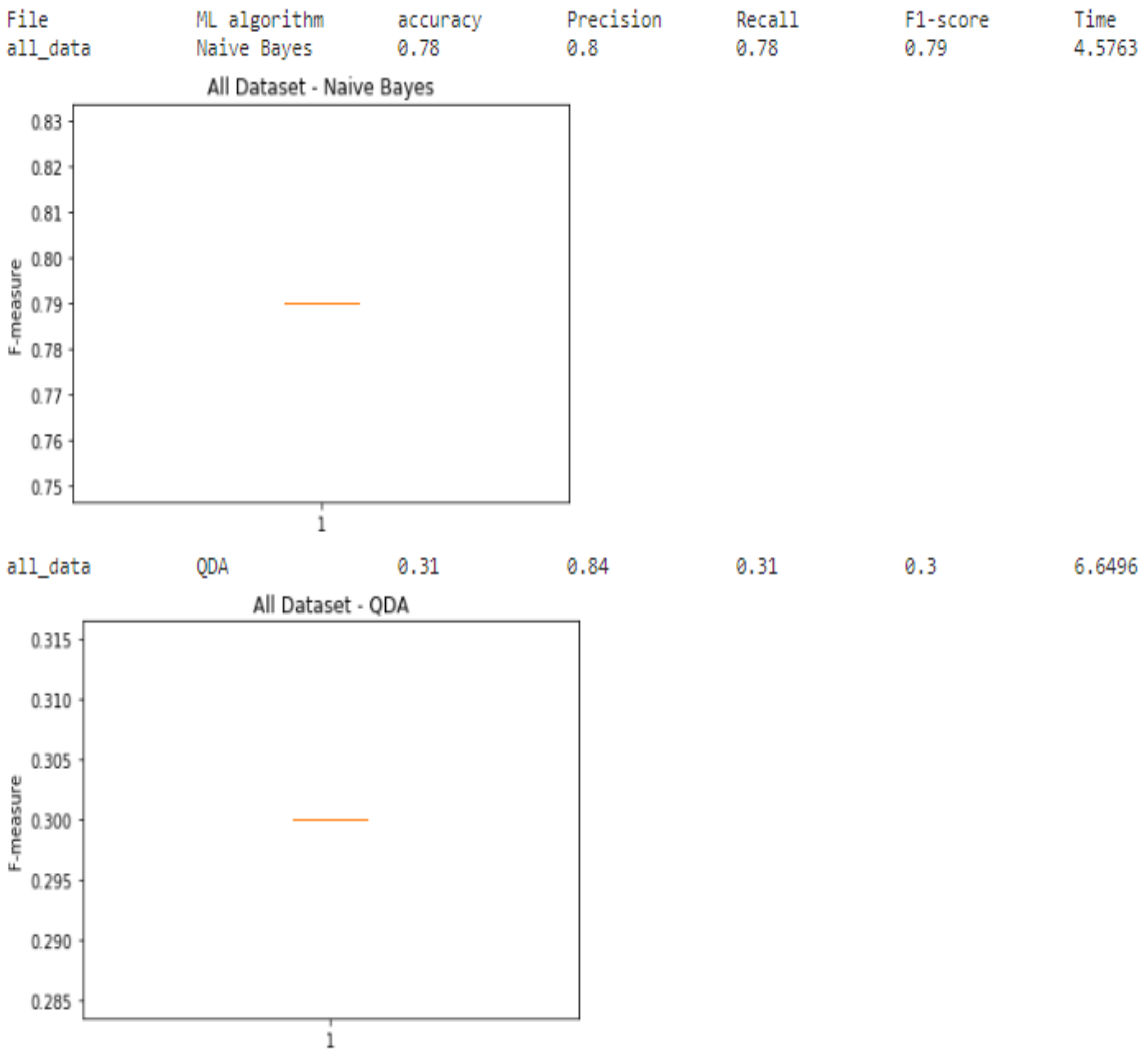


Рисунок 4.9 – Оцінка роботи алгоритмів машинного навчання

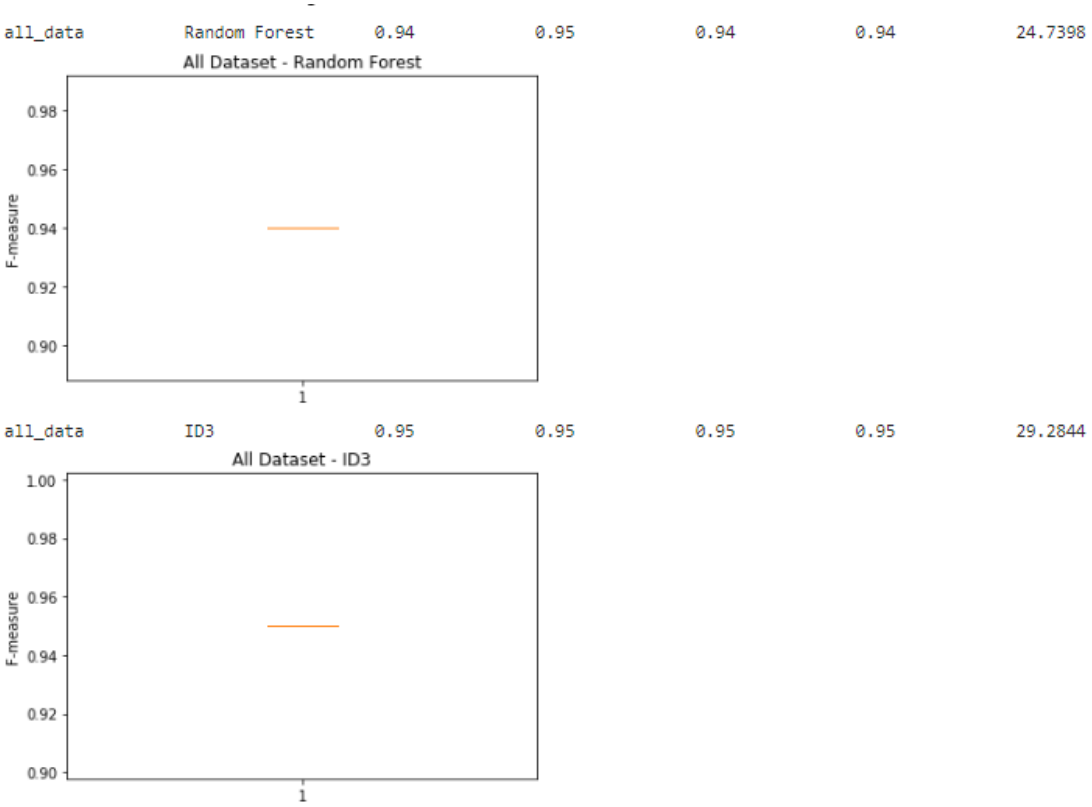


Рисунок 4.10 – Оцінка роботи алгоритмів машинного навчання

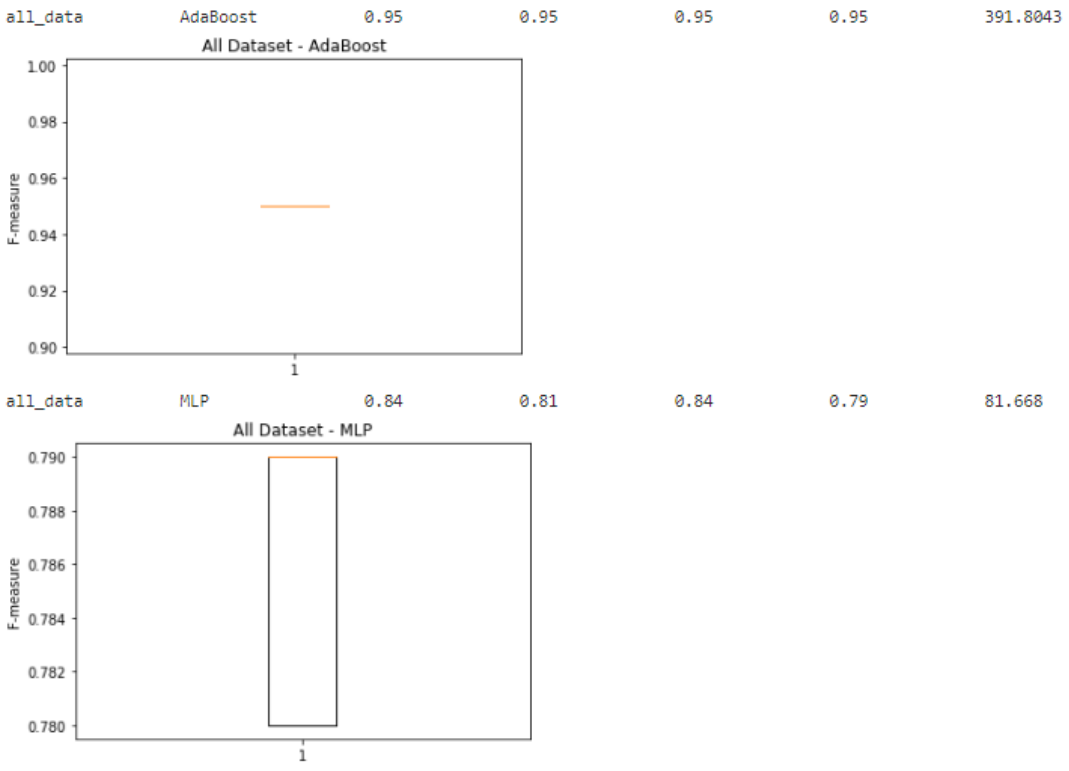


Рисунок 4.11 – Оцінка роботи алгоритмів машинного навчання

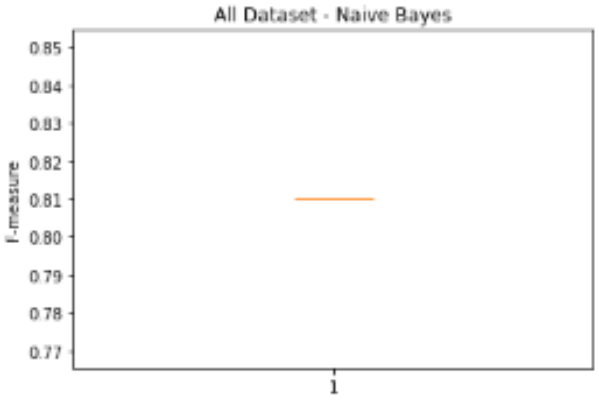
Рисунок 4.12 – Оцінка роботи алгоритмів машинного навчання

Час останнього запуску цього файлу був записаний як 25082 секунди.

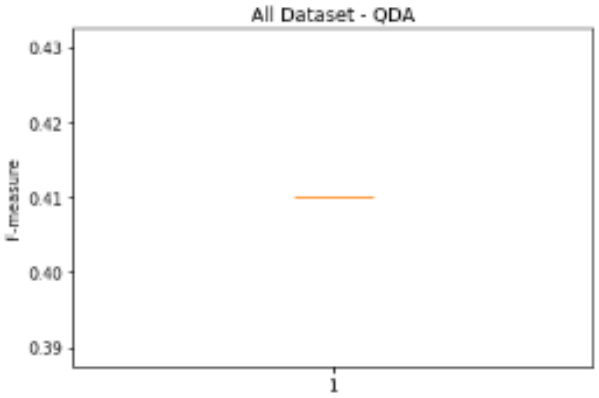
4.3 Машинне навчання. Реалізація з 7 функціями

Ця програма реалізує методи машинного навчання у файлі "all_data.csv". Використовуються 7 функцій з найбільшою вагою, створених файлом feature_selection_for_all_data. Цей файл застосовує 7 алгоритмів машинного навчання до файлу "all_data.csv" 10 разів і друкує результати цих операцій на екрані та у файлі "./attacks/results_3.csv". Він також створює графічні зображення з результатами та друкує їх як на екрані, так і в папці "./attacks/result_graph_3/". Це подано на рисунках 4.13 та 4.14.

File	ML algorithm	accuracy	Precision	Recall	F1-score	Time
all_data	Naive Bayes	0.82	0.8	0.82	0.81	1.6258



all_data	QDA	0.38	0.83	0.38	0.41	1.9025



all_data	Random Forest	0.94	0.95	0.94	0.94	20.512

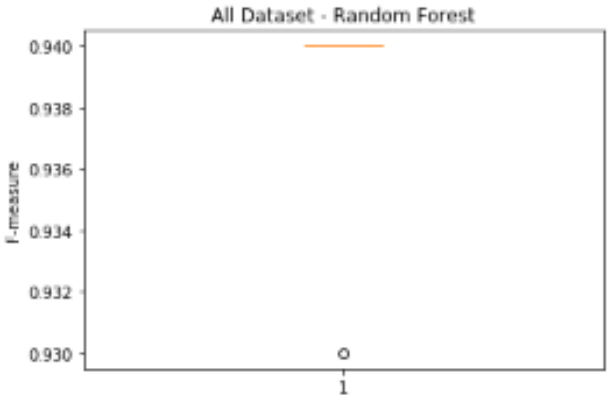


Рисунок 4.13 – Оцінка роботи алгоритмів машинного навчання

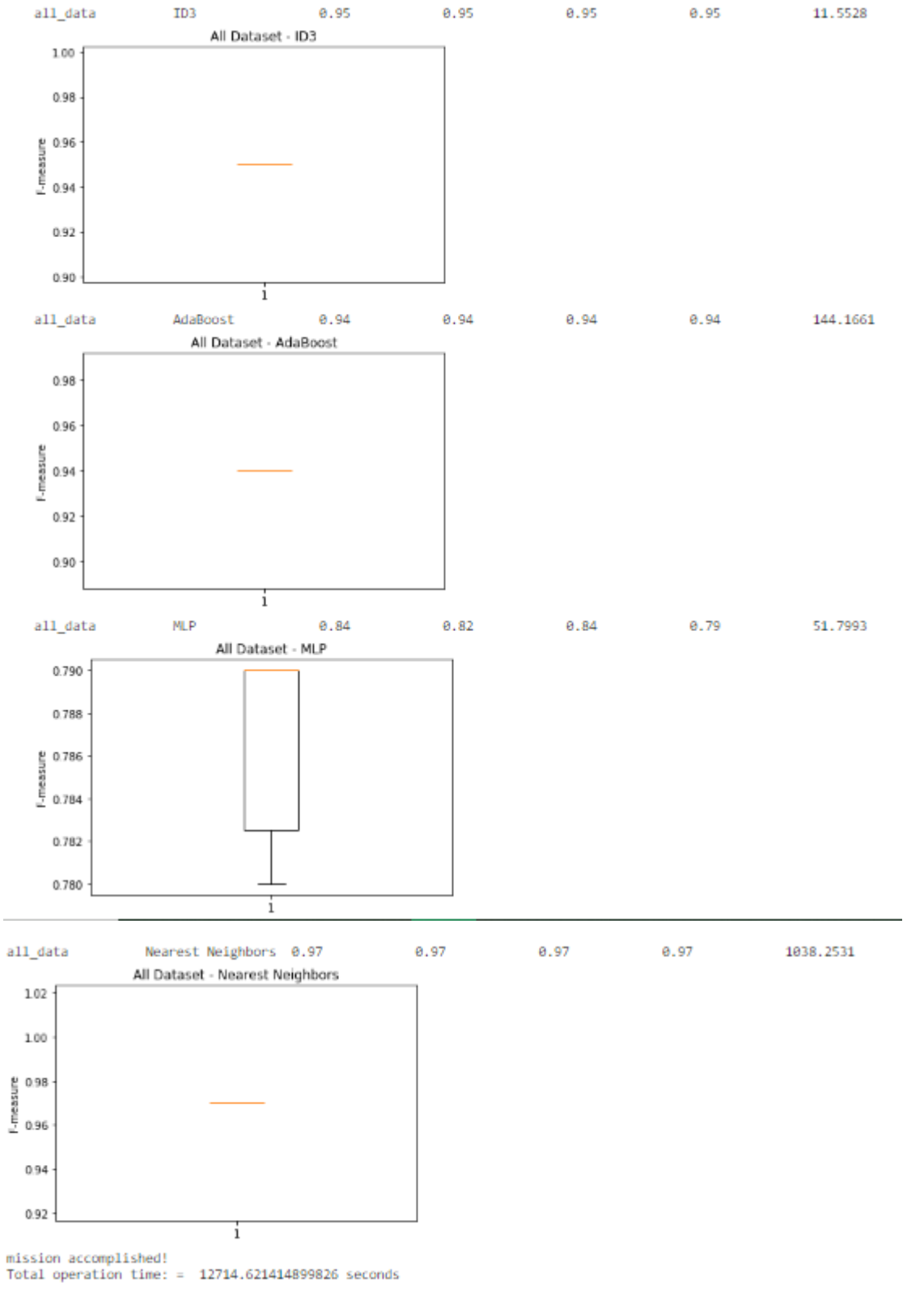


Рисунок 4.14 – Оцінка роботи алгоритмів машинного навчання

Час останнього виконання цього файлу був записаний як 12714 секунд.

4.4 Порівняння мір

Для реалізації частини необхідно працювати з файлом "all_data.csv". Він знаходить функцію, що забезпечує найвищу f-міру для алгоритмів Naive Bayes, QDA і MLP, і друкує їх на екрані, що показано на рисунку 4.15.

```
ML algorithm      Feature Name                  F1-score    Accuracy   Feature List
Naive Bayes       Bwd Packet Length Std         0.86        0.88       [1,            ------> New feature found!!!
Naive Bayes       Flow Bytes/s                  0.84        0.86       [1, 2,
Naive Bayes       Total Length of Fwd Packets   0.86        0.88       [1, 2,         ------> New feature found!!!
Naive Bayes       Fwd Packet Length Std         0.85        0.86       [1, 2, 3,
Naive Bayes       Flow IAT Std                  0.85        0.86       [1, 2, 3,
Naive Bayes       Flow IAT Min                  0.86        0.88       [1, 2, 3,      ------> New feature found!!!
Naive Bayes       Fwd IAT Total                 0.82        0.83       [1, 2, 3, 4,
Naive Bayes       Flow Duration                 0.82        0.83       [1, 2, 3, 4,
Naive Bayes       Bwd Packet Length Max         0.85        0.87       [1, 2, 3, 4,
Naive Bayes       Flow IAT Max                  0.83        0.85       [1, 2, 3, 4,
Naive Bayes       Flow IAT Mean                 0.85        0.87       [1, 2, 3, 4,
Naive Bayes       Total Length of Bwd Packets   0.07        0.18       [1, 2, 3, 4,
Naive Bayes       Fwd Packet Length Min         0.86        0.88       [1, 2, 3, 4,   ------> New feature found!!!
Naive Bayes       Bwd Packet Length Mean        0.85        0.86       [1, 2, 3, 4, 5,
Naive Bayes       Flow Packets/s                0.86        0.87       [1, 2, 3, 4, 5,  ------> New feature found!!!
Naive Bayes       Fwd Packet Length Mean        0.86        0.87       [1, 2, 3, 4, 5, 6,  ------> New feature found!!!
Naive Bayes       Total Backward Packets        0.13        0.21       [1, 2, 3, 4, 5, 6, 7,
Naive Bayes       Total Fwd Packets             0.52        0.48       [1, 2, 3, 4, 5, 6, 7,
Naive Bayes       Fwd Packet Length Max         0.84        0.84       [1, 2, 3, 4, 5, 6, 7,
Naive Bayes       Bwd Packet Length Min         0.85        0.85       [1, 2, 3, 4, 5, 6, 7,
F1= 0.86 Naive Bayes  The most efficient feature list = ['Bwd Packet Length Std', 'Total Length of Fwd Packets', 'Flow IAT Min', 'Fwd Packet Length Mi
n', 'Flow Packets/s', 'Fwd Packet Length Mean']

QDA               Bwd Packet Length Std         0.86        0.88       [1,            ------> New feature found!!!
QDA               Flow Bytes/s                  0.86        0.88       [1, 2,         ------> New feature found!!!
QDA               Total Length of Fwd Packets   0.86        0.87       [1, 2, 3,      ------> New feature found!!!
QDA               Fwd Packet Length Std         0.3         0.31       [1, 2, 3, 4,
QDA               Flow IAT Std                  0.84        0.85       [1, 2, 3, 4,
QDA               Flow IAT Min                  0.86        0.88       [1, 2, 3, 4,   ------> New feature found!!!
QDA               Fwd IAT Total                 0.84        0.85       [1, 2, 3, 4, 5,
QDA               Flow Duration                 0.84        0.86       [1, 2, 3, 4, 5,
QDA               Bwd Packet Length Max         0.84        0.85       [1, 2, 3, 4, 5,
QDA               Flow IAT Max                  0.84        0.85       [1, 2, 3, 4, 5,
QDA               Flow IAT Mean                 0.85        0.87       [1, 2, 3, 4, 5,
QDA               Total Length of Bwd Packets   0.08        0.18       [1, 2, 3, 4, 5,
QDA               Fwd Packet Length Min         0.85        0.87       [1, 2, 3, 4, 5,
QDA               Bwd Packet Length Mean        0.84        0.86       [1, 2, 3, 4, 5,
QDA               Flow Packets/s                0.85        0.86       [1, 2, 3, 4, 5,
QDA               Fwd Packet Length Mean        0.84        0.85       [1, 2, 3, 4, 5,
QDA               Total Backward Packets        0.11        0.2        [1, 2, 3, 4, 5,
QDA               Total Fwd Packets             0.1         0.19       [1, 2, 3, 4, 5,
QDA               Fwd Packet Length Max         0.85        0.86       [1, 2, 3, 4, 5,
QDA               Bwd Packet Length Min         0.61        0.56       [1, 2, 3, 4, 5,
F1= 0.86 QDA  The most efficient feature list = ['Bwd Packet Length Std', 'Flow Bytes/s', 'Total Length of Fwd Packets', 'Flow IAT Min']


MLP               Bwd Packet Length Std         0.86        0.88       [1,            ------> New feature found!!!
MLP               Flow Bytes/s                  0.86        0.89       [1, 2,         ------> New feature found!!!
MLP               Total Length of Fwd Packets   0.76        0.83       [1, 2, 3,
MLP               Fwd Packet Length Std         0.87        0.89       [1, 2, 3,      ------> New feature found!!!
MLP               Flow IAT Std                  0.76        0.83       [1, 2, 3, 4,
MLP               Flow IAT Min                  0.88        0.89       [1, 2, 3, 4,   ------> New feature found!!!
MLP               Fwd IAT Total                 0.77        0.84       [1, 2, 3, 4, 5,
MLP               Flow Duration                 0.78        0.84       [1, 2, 3, 4, 5,
MLP               Bwd Packet Length Max         0.87        0.89       [1, 2, 3, 4, 5,
MLP               Flow IAT Max                  0.79        0.84       [1, 2, 3, 4, 5,
MLP               Flow IAT Mean                 0.79        0.84       [1, 2, 3, 4, 5,
MLP               Total Length of Bwd Packets   0.78        0.83       [1, 2, 3, 4, 5,
MLP               Fwd Packet Length Min         0.88        0.89       [1, 2, 3, 4, 5,  ------> New feature found!!!
MLP               Bwd Packet Length Mean        0.87        0.89       [1, 2, 3, 4, 5, 6,
MLP               Flow Packets/s                0.87        0.89       [1, 2, 3, 4, 5, 6,
MLP               Fwd Packet Length Mean        0.87        0.89       [1, 2, 3, 4, 5, 6,
MLP               Total Backward Packets        0.88        0.89       [1, 2, 3, 4, 5, 6,  ------> New feature found!!!
MLP               Total Fwd Packets             0.88        0.89       [1, 2, 3, 4, 5, 6, 7,  ------> New feature found!!!
MLP               Fwd Packet Length Max         0.88        0.9        [1, 2, 3, 4, 5, 6, 7, 8,  ------> New feature found!!!
MLP               Bwd Packet Length Min         0.88        0.9        [1, 2, 3, 4, 5, 6, 7, 8, 9,  ------> New feature found!!!
F1= 0.88 MLP  The most efficient feature list = ['Bwd Packet Length Std', 'Flow Bytes/s', 'Fwd Packet Length Std', 'Flow IAT Min', 'Fwd Packet Length M
in', 'Total Backward Packets', 'Total Fwd Packets', 'Fwd Packet Length Max', 'Bwd Packet Length Min']


mission accomplished!
operation time: =  2092.819859981537 secomds
```

Рисунок 4.15 – Оцінка роботи алгоритмів машинного навчання для алгоритмів Naive Bayes, QDA і MLP

Час останнього запуску цього файлу був записаний як 2092 секунди.

4.5 Остаточна реалізація машинного навчання

Ця реалізації цієї частини програми необхідно використати файл "all_data.csv" як набір даних. Щоб підвищити продуктивність алгоритмів Naive Bayes, QDA і MLP, він використовує функції, створені файлом ml_F-criterion_comparison. В інших чотирьох алгоритмах він використовує 7 особливостей з найвищою значимістю, створених файлом feature_selection_for_all_data.

Цей файл застосовує 7 алгоритмів машинного навчання до файлу "all_data.csv" 10 разів і друкує результати цих операцій на екрані та у файлі "./attacks/results_final.csv". Він також створює графічні зображення з результатами та друкує їх як на екрані, так і в папці "./attacks/result_graph_final/"., що представлені на рисунках 4.16-4.17.

Feature Number	Feature
1	Bwd Packet Length Std
2	Flow Bytes/s
3	Total Length of Fwd Packets
4	Fwd Packet Length Std
5	Flow IAT Std
6	Flow IAT Min
7	Fwd IAT Total
8	Flow Duration
9	Bwd Packet Length Max
10	Flow IAT Max
11	Flow IAT Mean
12	Total Length of Bwd Packets
13	Fwd Packet Length Min
14	Bwd Packet Length Mean
15	Flow Packets/s
16	Fwd Packet Length Mean
17	Total Backward Packets
18	Total Fwd Packets
19	Fwd Packet Length Max
20	Bwd Packet Length Min

ML algorithm	Feature Name	F1-score	Accuracy	Feature List
Naive Bayes	Bwd Packet Length Std	0.86	0.88	[1, ------> New feature found!!!
Naive Bayes	Flow Bytes/s	0.84	0.86	[1, 2,
Naive Bayes	Total Length of Fwd Packets	0.86	0.88	[1, 2, ------> New feature found!!!
Naive Bayes	Fwd Packet Length Std	0.85	0.86	[1, 2, 3,
Naive Bayes	Flow IAT Std	0.85	0.86	[1, 2, 3,
Naive Bayes	Flow IAT Min	0.86	0.88	[1, 2, 3, ------> New feature found!!!
Naive Bayes	Fwd IAT Total	0.82	0.83	[1, 2, 3, 4,
Naive Bayes	Flow Duration	0.82	0.83	[1, 2, 3, 4,
Naive Bayes	Bwd Packet Length Max	0.85	0.87	[1, 2, 3, 4,
Naive Bayes	Flow IAT Max	0.83	0.85	[1, 2, 3, 4,
Naive Bayes	Flow IAT Mean	0.85	0.87	[1, 2, 3, 4,
Naive Bayes	Total Length of Bwd Packets	0.07	0.18	[1, 2, 3, 4,
Naive Bayes	Fwd Packet Length Min	0.86	0.88	[1, 2, 3, 4, ------> New feature found!!!
Naive Bayes	Bwd Packet Length Mean	0.85	0.86	[1, 2, 3, 4, 5,
Naive Bayes	Flow Packets/s	0.86	0.87	[1, 2, 3, 4, 5, ------> New feature found!!!
Naive Bayes	Fwd Packet Length Mean	0.86	0.87	[1, 2, 3, 4, 5, 6, ------> New feature found!!!
Naive Bayes	Total Backward Packets	0.13	0.21	[1, 2, 3, 4, 5, 6, 7,
Naive Bayes	Total Fwd Packets	0.52	0.48	[1, 2, 3, 4, 5, 6, 7,
Naive Bayes	Fwd Packet Length Max	0.84	0.84	[1, 2, 3, 4, 5, 6, 7,
Naive Bayes	Bwd Packet Length Min	0.85	0.85	[1, 2, 3, 4, 5, 6, 7,

F1= 0.86 Naive Bayes The most efficient feature list = ['Bwd Packet Length Std', 'Total Length of Fwd Packets', 'Flow IAT Min', 'Fwd Packet Length Min', 'Flow Packets/s', 'Fwd Packet Length Mean']

Рисунок 4.16 – Оцінка роботи та підвищеня продуктивності алгоритмів машинного навчання для алгоритмів Naive Bayes

```
QDA             Bwd Packet Length Std         0.86       0.88       [1,            ------> New feature found!!!
QDA             Flow Bytes/s                  0.86       0.88       [1, 2,         ------> New feature found!!!
QDA             Total Length of Fwd Packets   0.86       0.87       [1, 2, 3,      ------> New feature found!!!
QDA             Fwd Packet Length Std         0.3        0.31       [1, 2, 3, 4,
QDA             Flow IAT Std                  0.84       0.85       [1, 2, 3, 4,
QDA             Flow IAT Min                  0.86       0.88       [1, 2, 3, 4,   ------> New feature found!!!
QDA             Fwd IAT Total                 0.84       0.85       [1, 2, 3, 4, 5,
QDA             Flow Duration                 0.84       0.86       [1, 2, 3, 4, 5,
QDA             Bwd Packet Length Max         0.84       0.85       [1, 2, 3, 4, 5,
QDA             Flow IAT Max                  0.84       0.85       [1, 2, 3, 4, 5,
QDA             Flow IAT Mean                 0.85       0.87       [1, 2, 3, 4, 5,
QDA             Total Length of Bwd Packets   0.08       0.18       [1, 2, 3, 4, 5,
QDA             Fwd Packet Length Min         0.85       0.87       [1, 2, 3, 4, 5,
QDA             Bwd Packet Length Mean        0.84       0.86       [1, 2, 3, 4, 5,
QDA             Flow Packets/s                0.85       0.86       [1, 2, 3, 4, 5,
QDA             Fwd Packet Length Mean        0.84       0.85       [1, 2, 3, 4, 5,
QDA             Total Backward Packets        0.11       0.2        [1, 2, 3, 4, 5,
QDA             Total Fwd Packets             0.1        0.19       [1, 2, 3, 4, 5,
QDA             Fwd Packet Length Max         0.85       0.86       [1, 2, 3, 4, 5,
QDA             Bwd Packet Length Min         0.61       0.56       [1, 2, 3, 4, 5,
F1= 0.86 QDA  The most efficient feature list = ['Bwd Packet Length Std', 'Flow Bytes/s', 'Total Length of Fwd Packets', 'Flow IAT Min']

MLP             Bwd Packet Length Std         0.86       0.88       [1,            ------> New feature found!!!
MLP             Flow Bytes/s                  0.86       0.89       [1, 2,         ------> New feature found!!!
MLP             Total Length of Fwd Packets   0.76       0.83       [1, 2, 3,
MLP             Fwd Packet Length Std         0.87       0.89       [1, 2, 3,      ------> New feature found!!!
MLP             Flow IAT Std                  0.76       0.83       [1, 2, 3, 4,
MLP             Flow IAT Min                  0.88       0.89       [1, 2, 3, 4,   ------> New feature found!!!
MLP             Fwd IAT Total                 0.77       0.84       [1, 2, 3, 4, 5,
MLP             Flow Duration                 0.78       0.84       [1, 2, 3, 4, 5,
MLP             Bwd Packet Length Max         0.87       0.89       [1, 2, 3, 4, 5,
MLP             Flow IAT Max                  0.79       0.84       [1, 2, 3, 4, 5,
MLP             Flow IAT Mean                 0.79       0.84       [1, 2, 3, 4, 5,
MLP             Total Length of Bwd Packets   0.78       0.83       [1, 2, 3, 4, 5,
MLP             Fwd Packet Length Min         0.88       0.89       [1, 2, 3, 4, 5,  ------> New feature found!!!
MLP             Bwd Packet Length Mean        0.87       0.89       [1, 2, 3, 4, 5, 6,
MLP             Flow Packets/s                0.87       0.89       [1, 2, 3, 4, 5, 6,
MLP             Fwd Packet Length Mean        0.87       0.89       [1, 2, 3, 4, 5, 6,
MLP             Total Backward Packets        0.88       0.89       [1, 2, 3, 4, 5, 6,  ------> New feature found!!!
MLP             Total Fwd Packets             0.88       0.89       [1, 2, 3, 4, 5, 6, 7,  ------> New feature found!!!
MLP             Fwd Packet Length Max         0.88       0.9        [1, 2, 3, 4, 5, 6, 7, 8,  ------> New feature found!!!
MLP             Bwd Packet Length Min         0.88       0.9        [1, 2, 3, 4, 5, 6, 7, 8, 9,  ------> New feature found!!!
F1= 0.88 MLP  The most efficient feature list = ['Bwd Packet Length Std', 'Flow Bytes/s', 'Fwd Packet Length Std', 'Flow IAT Min', 'Fwd Packet Lengt
h Min', 'Total Backward Packets', 'Total Fwd Packets', 'Fwd Packet Length Max', 'Bwd Packet Length Min']


mission accomplished!
operation time: =  2092.819859981537 secomds
```

Рисунок 4.16 – Оцінка роботи та підвищеня продуктивності алгоритмів машинного навчання для алгоритмів Naive Bayes, QDA і MLP

Час останнього виконання цього файлу був записаний як 18561 секунда.

ВИСНОВКИ

Проблема виявлення аномалій в даних знаходить застосування в багатьох областях, в яких необхідно встановити незвичні події в діяльності, що генерує такі дані. Метою всіх методів виявлення аномалій зазвичай є створення алгоритмічної, імовірнісної або статистичної моделі, що характеризує нормальну поведінку системи.

Методи машинного навчання пропонують ряд алгоритмів ефективного навчання параметрів цих моделей. Відхилення від цих моделей використовуються для визначення викидів. Хороша обізнаність з галуззю з якої отримані вихідні дані часто має вирішальне значення для розробки простих та точних моделей, які не перенавчаються на тестових вибірках.

Проблема виявлення аномалій стає особливо складною коли існують значні зв’язки між різними точками даних. В часових рядах та мережах даних шаблони взаємовідносин поміж окремими точками відіграють ключову роль в визначенні викидів.

В роботі представлено види мережевих загроз та атак, проаналізовано системи виявлення вторгнень, а також види систем запобігання вторгненням. Таким чином було описано та проаналізовано набори даних, а саме: багатовимірні дані, виявлення викидів часових рядів, виявлення аномалій а режимі реального часу, та контрольні показники мета-аналізу виявлення аномалій.

Запропоновано та реалізовано виявлення вторгнень в комп’ютерну мережу на основі технологій машиного навчання.

Аналіз аномалій засобами машинного навчання має великий простір подальших досліджень, особливо в області виявлення аномалій та вторгнень.

ПЕРЕЛІК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Експлойт [Електронний ресурс]. – Режим доступу до ресурсу: https://www.eset.com/ua-ru/support/information/entsiklopediya-ugroz/eksploit/

2. Види атак [Електронний ресурс]. – Режим доступу до ресурсу: https://www.geeksforgeeks.org/basic-network-attacks-in-computer-network/

3. CYBER EDU. What is Network Security? [Електронний ресурс]. – Режим доступу до ресурсу: https://www.forcepoint.com/cyber-edu/network-security

4. Network Attacks and Network Security Threats [Електронний ресурс]. – Режим доступу до ресурсу: https://www.cynet.com/network-attacks/network-attacks-and-network-security-threats/

5. A Python Toolbox for Scalable Outlier Detection (Anomaly Detection) [Електроний ресурс]. – Режим доступа до ресурса: https://GitHub-yzhao062/pyod

6. Streaming Anomaly Detection Framework in Python (Outlier Detection for Streaming Data) [Електроний ресурс]. – Режим доступа до ресурса: https://GitHub - selimfirat/pysad:

7. Novelty and Outlier Detection – scikit-learn 1.0.1 documentation [Електроний ресурс]. – Режим доступа до ресурса:https://scikit-learn.org

8. Никишова А. В. Интеллектуальная система обнаружения атак на основе многоагентного подхода // Вестник Волгоградского государственного университета. Серия 10. Инновационная деятельность.. – 2011. – № 5. – С. 35–37.

9. Аткина В. С. Оценка эффективности катастрофоустойчивых решений // Вестник Волгоградского государственного университета. Серия 10. Инновационная деятельность.. – 2012. – № 6. – С. 45–48.

10. Estevez J., Garcya P., Dyaz J. «Anomaly detection methods in wired networks: a survey and taxonomy». Computer Networks, том.27 – №.16. – 2004. – С. 1569–84.

11. Garcıa T., Dıaz V., Macia F., Vazquezb. «Anomaly-based network intrusion detection». Computers and security, том 28. – 2009. – С. 18 –28.

12. Omar S., Ngadi A., Jebur H. «Machine Learning Techniques for Anomaly Detection: An Overview». International Journal of Computer Applications, том 79. – № 2.— 2013 – С. 33–41.

13. An Acceleration System for Large-scare Unsupervised Heterogeneous Outlier Detection (Anomaly Detection) [Електроний ресурс] // Режим доступа до ресурса:https://GitHub - yzhao062/SUOD

14. UTM [Електронний ресурс]. – Режим доступу до ресурсу: https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-prevention-system-ips

15. ELKI Data Mining Framework [Електроний ресурс]. – Режим доступа до ресурса:https://elki-project.github.io

16. RapidMiner Extension for Anomaly Detection [Електроний ресурс]. – Режим доступа до ресурса: https://GitHub-Markus-Go/rapidminer-anomalydetection:

17. IPS [Електронний ресурс]. – Режим доступу до ресурсу: https://www.techtarget.com/searchsecurity/definition/intrusion-prevention

18. cran.r-project [Електроний ресурс]. – Режим доступа до ресурса: https://cran.r-project.org/view=AnomalyDetection

19. r-project [Електроний ресурс]. – Режим доступа до ресурса: http://www.r-project.org/

20. AnomalyDetectionToolbox [Електроний ресурс]. – Режим доступа до ресурса:https://dsmi-lab-ntust.github.io

21. TODS: An Automated Time-series Outlier Detection System [Електроний ресурс]. – Режим доступа до ресурса: https:// GitHub-datamllab/tods:

22. Anomaly detection [Електроний ресурс]. – Режим доступа до ресурса: https://GitHub - earthgecko/skyline:

23. Banpei. Anomaly detection library based on singular spectrum transformation(sst) [Електроний ресурс] – Режим доступа до ресурса: https://GitHub - tsurubee/banpei:

24. Telemanom [Електроний ресурс] – Режим доступа до ресурса: https://arxiv.org/abs/1802.04431

25. Repository of the paper "A Systematic Evaluation of Deep Anomaly Detection Methods for Time Series" [Електроний ресурс] – Режим доступа до ресурса:https://GitHub - KDD-OpenSource/DeepADoTS:.

26. The Numenta Anomaly Benchmark [Електроний ресурс] – Режим доступа до ресурса:https://GitHub - numenta/NAB:

27. Anomaly Detection with R [Електроний ресурс] – Режим доступа до ресурса:https:// GitHub - twitter/AnomalyDetection

28. CRAN - Package anomalize [Електроний ресурс] – Режим доступа до ресурса:https://r-project.org

29. Real Time Anomaly Detection in Open Distro for Elasticsearch [Електроний ресурс] – Режим доступа до ресурса:https://Open Distro.io

30. An open-source framework for real-time anomaly detection using Python, ElasticSearch and Kibana [Електроний ресурс] – Режим доступа до ресурса: https://GitHub - MentatInnovations/datastream.io

31. ODDS – Outlier Detection DataSets [Електроний ресурс] – Режим доступа до ресурса:https://stonybrook.edu

32. Розмір даних для виявлення аномалій без нагляду [Електроний ресурс] – Режим доступа до ресурса:https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/OPQMVF

33. Контрольні показники мета-аналізу виявлення аномалій [Електроний ресурс]. – Режим доступа до ресурса: https://ir.library.oregonstate.edu/concern/datasets/47429f155

34. Skoltech Anomaly Benchmark [Електроний ресурс] – Режим доступа до ресурса: https://github.com/waico/skab