Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістра

студента *Явіра Ярослава Романовича*

академічної групи *125м-22-2*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Дослідження алгоритмів виявлення DDoS атак із використанням машинного навчання*

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	доц. Ковальова Ю.В.			
розділів:				
спеціальний	доц. Ковальова Ю.В..			
економічний	доц. к.е.н. Пілова Д.П.	95	відмінно	

Рецензент				

Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2023

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу ступеня магістра

студенту ***Явіру Я. Р.*** **Академічної групи** **125м-22-2**

(прізвище та ініціали) (шифр)

спеціальності ***125 Кібербезпека***

за освітньо-професійною програмою ***Кібербезпека***

на тему ***Дослідження алгоритмів виявлення DDoS атак із використанням машинного навчання***

Затверджену наказом ректора НТУ «Дніпровська політехніка» від _____№ ___

Розділ	**Зміст**	**Термін виконання**
Розділ № 1	*Огляд літератури за темою, постановка задачі.*	
Розділ № 2	*Розробка системи класифікації трафіку за допомогою машинного навчання*	
Розділ № 3	*Розрахунок витрат пов'язаними з впровадженням методів захисту*	

Завдання видано_____ Ковальова Ю.В
(підпис керівника) (прізвище, ініціали)
Дата видачі завдання: _____

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____ Явір Я.Р
(підпис студента) (прізвище, ініціали

РЕФЕРАТ

Пояснювальна записка: ст. 89, рис.20, табл. 9, додатків 4, джерел 25

Предмет дослідження: комп'ютерні мережі і розподілені атаки типу DDoS, спрямовані на відмову в обслуговуванні в цих мережах.

Методи дослідження: спостереження, обстеження, аналіз, опис.

Мета проекту: створити алгоритм який здатний класифікувати DDoS трафік за допомогою машинного навчання

У першому розділі кваліфікаційної роботи наведено опис, що саме підлягає визначенню DDoS, які є види DDoS атак та як вони реалізовуються, що таке машинне навчання, які саме методи навчання використовуються для виявлення DDoS атак.

У другому розділі представлені різноманітні алгоритми класифікації трафіку, датасети за якими можна визначаються ознаки за якими відбувається класифікація, алгоритм створення моделі за допомогою якої можна визначати чи належить трафік до DDoS атаки.

У третьому розділі розраховано витрати на впровадження данного методу протидії DDos атакам, прорахована економічна доцільність створення та використання алгоритму з машинним навчанням.

ІНФОРМАЦІЙНА БЕЗПЕКА, DDoS-АТАКА, МАШИННЕ НАВЧАННЯ, МЕТОДИ РЕАЛІЗАЦІЇ ДДОС АТАК, КЛАСИФІКАЦІЯ

ABSTRACT

Explonetory note: p. 89, fig. 20, tables 9, applications 4, sources 25.

The subject of research: computer networks and distributed DDoS attacks aimed at denial of service in these networks.

Research methods: observation, survey, analysis, description.

The goal of the project: to create an algorithm capable of classifying DDoS traffic.

In first section of the qualification work describes what is to be defined as DDoS, what are the types of DDoS attacks and how they are implemented, what is machine learning, what learning methods are used to detect DDoS attacks.The second section provides

The second section presents a variety of traffic classification algorithms, datasets that can be used to determine the features used for classification, and an algorithm for creating a model that can be used to determine whether traffic belongs to a DDoS attack.

The third section calculates the costs of implementing this method of countering DDos attacks, calculates the economic feasibility of creating and using an algorithm with machine learning.

INFORMATION SECURITY, DDoS ATTACK, MACHINE LEARNING, DDOS ATTACK IMPLEMENTATION METHODS, CLASSIFICATION

СПИСОК УМОВНИХ СКОРОЧЕНЬ

DoS – атака на відмову в обслуговуванні.

DDoS – розподілена атака на відмову в обслуговуванні.

ПЗ – програмне забезпечення.

IRC – Інтернет-ретрансляційний чат.

IoT – Internet of things (інтернет речей)

HTTP – Hyper Text Transfer Protocol (протокол передачі гіпертекстових документів)

UDP – User Datagram Protocol (протоколу датаграм користувача)

TCP – Transmission Control Protocol (Протокол керування передачею)

ICMP – Internet Control Message Protocol (міжмережевий протокол керуючих повідомлень)

IDS – Intrusion Detection System (Система виявлення вторгнень)

ML – машинне навчання

OSI - Модель взаємозв'язку відкритих систем

ISO – Міжнародна організація зі стандартизації

IP – адреса – Internet protocol Address

ROC – Receiver Operating Characteristic analysis (Аналіз робочих характеристик приймача)

TP – True positive (істинно позитивний)

TN – True negative (істинно негативний)

FN – False negative (помилково негативний)

FP – False positive (помилково позитивний)

ЗМІСТ

С

ВСТУП

У сучасному світі цифрових технологій, в якому інтернет-комунікації стають все більш важливою частиною нашого повсякденного життя, питання безпеки в мережі набуває особливого значення серед різноманітних загроз кібербезпеці, DDoS (розподілені атаки типу відмови в обслуговуванні) атаки стали одними з найбільш розповсюджених та небезпечних. Наприклад DDoS-атаки зросли у кількості на 542% за період з IV кварталу 2019 року по I квартал 2020 року. А в II кварталі їх кількість подвоїлася. При цьому кількість дрібних атак продовжує збільшуватися, а великі атаки стають все більш масштабними. Такі дані наводять Вівек Ганті (Vivek Ganti) і Омер Йоахімік (Omer Yoachimik) у блозі Cloudflare

Зростання обсягу та складності мережевого трафіку, разом із постійним розвитком нових видів атак, робить традиційні методи виявлення неефективними. У цьому контексті, машинне навчання виступає як потужний інструмент, здатний адаптуватися до змінних патернів трафіку та виявляти складні атаки, які можуть бути не помічені звичайними методами.

У даній роботі за мету ставиться є вирішення проблеми підвищення ефективності виявлення DDoS атак за рахунок використання алгоритмів машинного навчання. Для досягнення цієї мети необхідно вирішити наступні завдання:

- проаналізувати існуючі методи виявлення DdoS атак;
- розібрати код програми для виявлення DDoS атак;
- провести тестування алгоримів виявлення DDoS атак на наборі даних представлених в Інтернеті.
- провести оцінку алгоритмів та вибрати найкращий з них

Практичне значення роботи полягає в тому що результати можуть використовуватися для побудови системи виявлення DDOS Атак.

РОЗДІЛ 1. СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ.

1.1 DDoS атаки

Почнемо з того що дамо визначення, шо таке DoS. DoS – Denial of Service розшифровується як “відмова в обслуговуванні”, це один з найпоширеніших і базових видів кібератак, метою яких є виведення з ладу цільового ресурсу (сервера або системи).

Distributed Denial of Service – розподілена атака на відмову в обслуговуванні, це підвид DoS атаки який виконується водночас з багатьох пристроїв. Ці пристрої можуть бути реальними користувачами так і ботами.

Головна відмінність між DoS (Відмова в Обслуговуванні) та DDoS (Розподілена Відмова в Обслуговуванні) атаками полягає в тому, що остання включає використання численних компрометованих хостів. Ці заражені машини продовжують заливати цільові служби трафіком, перешкоджаючи нормальному доступу звичайних користувачів до цих системних сервісів. Процес DDoS атаки зображений на рисунку 1.1:

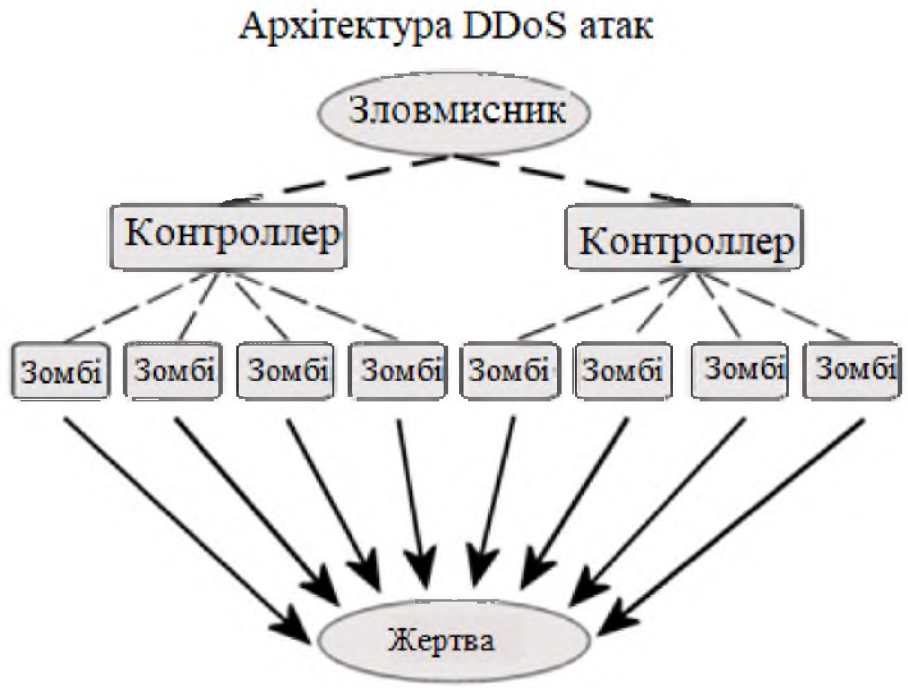

Рисунок 1.1 – як відбувається DDoS атака

В літературі наведено багато видів категоризації DDoS атак, це зроблено для того щоб розуміти тактики і технік для протидії для певних видів атак. Загалом їх категоризують за рівнем моделі OSI, на якому відбувається даний вид атаки, але є й інші типи категорій. Якщо їх усіх об'єднати то вийде класифікація зображена на рисунку 1.2.

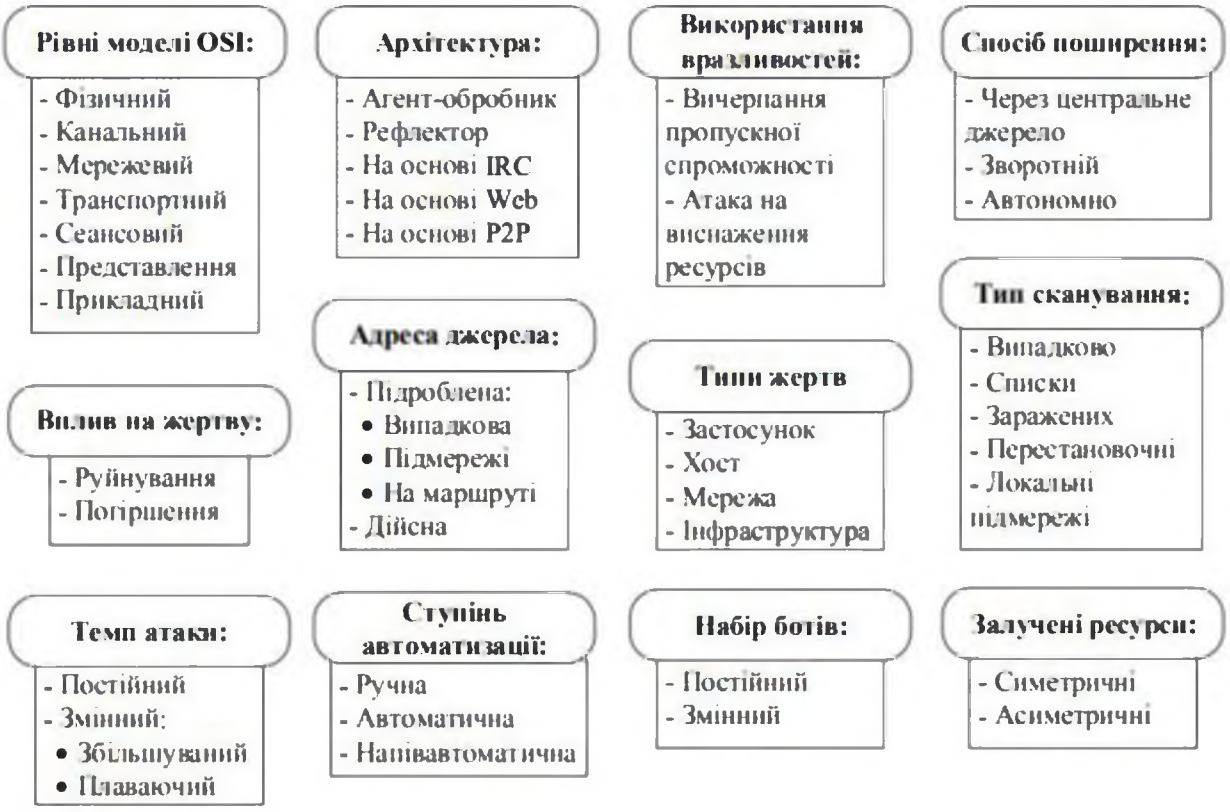

Рисунок 1.2 – Розширена класифікація DdoS атак.

1.2 Модель OSI

Важливо зазначити за модель OSI (Моделі взаємозв'язку відкритих систем), оскільки на протоколах даної моделі будується обмін інформації в усьому інтернеті, а отже усі ddos атаки використовують протоколи даної моделі для своєї реалізації.

У 1977 році ISO (Міжнародна організація зі стандартизації) почала розробку стандартів універсальної архітектури зв'язку, яка отримала назву Еталонної моделі взаємодії відкритих систем (Open System Interconnection, OSI), або скорочено – «модель OSI/ISO». Модель OSI є концепцією

застосування відкритих стандартів, спрямованою на забезпечення сумісності між різними системами, що дозволяє мінімізувати кількість угод, які не мають безпосереднього відношення до організації самого з'єднання між системами.[3]

У результаті розроблено еталонну модель яка зображена на рис. 1.3 , яка містить сім рівнів

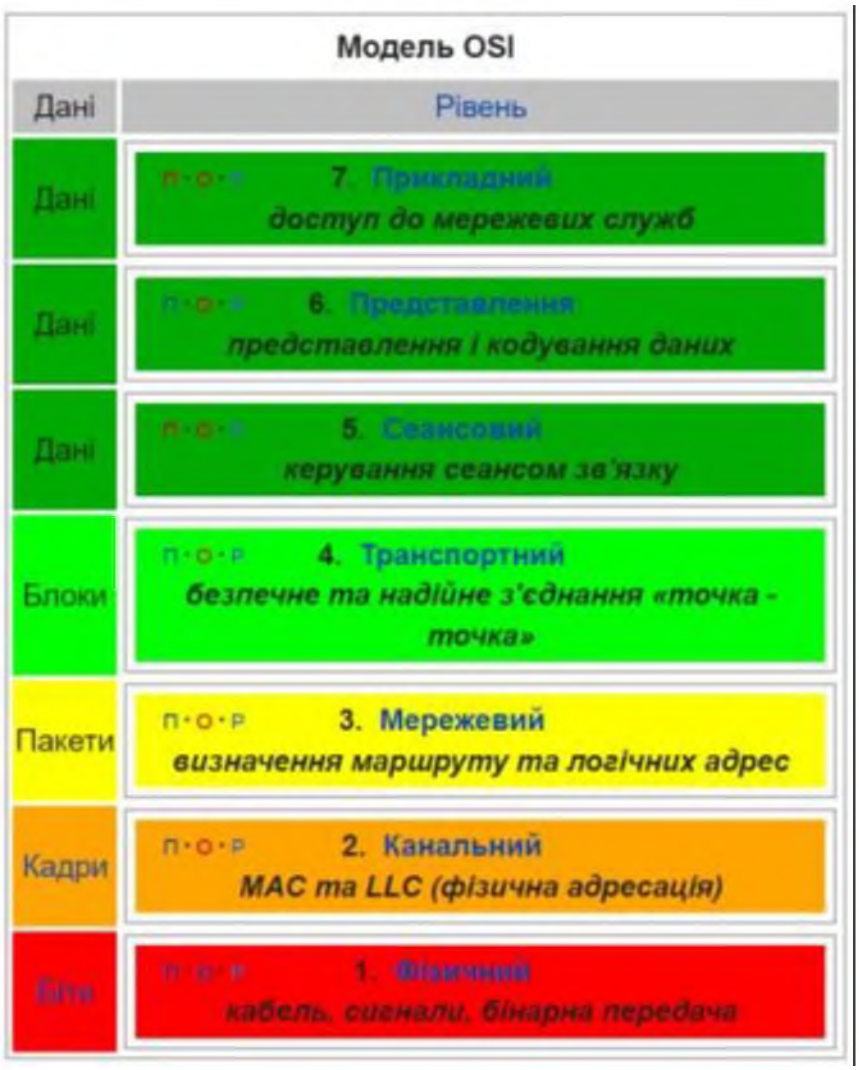

Рисунок 1.3 – Мережева модель OSI

Отже модель OSI – це стандартний набір правил та протоколів, який визначає різні рівні комунікації для комп'ютерних мереж і систем. Ця модель розділяє процес комунікації на сім рівнів, кожен з яких відповідає за конкретну функцію та взаємодіє з рівнями нижче і вище.

Прикладний рівень (Application layer) - це рівень на якому здійснюється керування терміналами й прикладними процесами в кінцевих системах мережі,

які є джерелами та споживачами інформації. Цей рівень надає сервіси безпосередньо для прикладних програмам користувачів. Щоб уникнути несумісності між призначеними для користувача програмами, прикладний рівень визначає стандартні способи надання сервісів цього рівня.

Він надає програмістам набір відкритих стандартних інтерфейсів прикладного програмування (Application Programming Interface, API), які можна використовувати для виконання таких функцій мережевого застосування, як передача файлів, віддалена реєстрація та ін.

Протоколи цього рівня: HTTP, SMTP, FTP та інші.

Представницький рівень (Presentation layer) здійснює інтерпретацію й перетворення передаваних у мережі даних до типу, зрозумілому прикладним процесам; забезпечує подання даних в узгоджених форматах і синтаксисі, трансляцію й інтерпретацію програм з різних мов, шифрування й стиснення даних.

Протоколи цього рівня: XML, TDI, XDR та інші.

Сеансовий рівень (Session layer) забезпечує виконання функцій керування сеансом зв'язку (сесією), орієнтованим на наскрізну передачу повідомлень. Сеансовий рівень після сесії здійснює поступове завершення, виконує процедуру квитування (відправки службового повідомлення про завершення сеансу зв'язку), що дозволяє запобігти втраті даних у разі, коли одна зі сторін має намір перервати діалог, а інша – ні.

Протоколи цього рівня: SSL, TLS, PPTP, DLC та інші.

Транспортний рівень **(**Transport Layer**) -** забезпечує надійну передачу даних між двома пристроями. Він вирішує питання контролю за потоком, розділення даних на пакети та відновлення даних після помилок передачі.

Протоколи цього рівня: TCP, UDP, RTP та інші.

Мережевий рівень (Network layer) виконує головну телекомунікаційну функцію – забезпечення зв'язку між кінцевими системами мережі. Цей зв'язок може бути реалізовано шляхом надання наскрізного каналу, скомутованого з окремих ділянок відповідно до оптимально обраного маршруту, логічного

віртуального каналу або безпосередньої маршрутизації блоку даних у процесі його доставки. При цьому мережевий рівень звільняє вищі рівні від знань про те, через які ділянки мережі або через які мережі проходить маршрут передачі інформації.

Основною функцією мережевого рівня є маршрутизація. Вона полягає в прийнятті рішення, через які конкретно проміжні пункти повинен пройти маршрут передавання даних, які направляються з однієї кінцевої системи в іншу, та як має виконуватися комутація (яка відповідає конкретному маршруту) між входами та виходами мережевих пристроїв, розташованих у проміжних пунктах мережі.

Протоколи цього рівня: ICMP, IGMP, IPX та інші.

Канальний рівень (Data–link layer) відповідає за якісну передачу даних між двома пунктами, пов'язаними фізичним каналом з урахуванням особливостей середовища-передавача. Якщо з'єднання встановлюється між двома кінцевими системами, не пов'язаними безпосередньо, то воно буде включати декілька незалежно функціонуючих фізичних каналів передачі даних. При цьому фізичні середовища передачі можуть відрізнятися (мідь, оптичне волокно, ефір). Несумісними можуть виявитися й вимоги до формату подання даних у кожному каналі, що називається лінійним кодуванням. У цій ситуації канальний рівень бере на себе функції адаптації даних до типу фізичного каналу зв'язку, надаючи вище розташованим рівням «прозоре з'єднання».

Протоколи цього рівня: ARCnet, ATM, SLIP та інші.

Фізичний рівень (Physical layer) відповідає за розміщення біт інформації у фізичному середовищі. На фізичному рівні можуть використовуватися такі типи середовищ: кабель «вита пара», коаксіальний кабель, оптичне волокно, територіальний цифровий канал і ефір. Основними характеристиками фізичних середовищ передачі є такі параметри, як смуга пропускання, перешкодозахищеність, хвильовий опір та інші. Тут реалізуються фізичні інтерфейси пристроїв з передавальним середовищем та пристроями, між якими виконується передавання бітів.

Протоколи цього рівня: RS-485, ITU-T, RJ-11 та інші.

1.3 Загальний метод реалізації DDoS атак

DDoS атаки, незважаючи на свою різноманітність, зазвичай включають в себе декілька основних етапів у своїй реалізації. Ось загальний опис цих етапів:

1. Планування та Підготовка: На цьому етапі зловмисник визначає цільову систему та стратегію атаки. Даний етап може включати в себе вивчення слабких місць цілі та вибір методу атаки (наприклад, об'ємні, протокольні або атаки на рівні додатків).

2. Створення "Botnets": Зловмисник мобілізує мережу інфікованих пристроїв (ботнет), які будуть використані для генерації штучного трафіку. Це може бути зроблено шляхом самостійного інфікування вразливих пристроїв або шляхом найму Botnets.

3. Тестування та Налаштування: Зловмисник може провести тестові атаки, щоб визначити ефективність та непомітність своєї стратегії. На цьому етапі також відбувається налаштування параметрів атаки, щоб максимізувати її вплив.

4. Запуск Атаки: Зловмисник ініціює DDoS атаку, керуючи ботнетом для надсилання великої кількості запитів до цільового сервера або мережі. Ці запити намагаються перевантажити систему та викликати збої в її роботі.

5. Підтримання та Модифікація Атаки: В залежності від реакції цілі, атаку може бути підтримано або модифіковано. Наприклад, якщо ціль ефективно протистоїть атакі, зловмисник може змінити тактику або збільшити інтенсивність атаки.

1.4 Botnet.

1.4.1 Історія Botnet

Головною відмінністю DDoS атак є використання

Історія Botnet розпочинається з 1993 року коли з'являється перший бот під назвою Eggdrop. Його створила щоб допомогти моделювати IRC (технологія багатокористувацьких конференцій в текстовому режимі через мережу Інтернет). Eggdrop був призначений для керування IRC-каналами. Він автоматизував багато завдань, таких як протидія спаму, управління доступом користувачів до каналів, та модерація дискусій. Однією з його ключових особливостей було те що користувачі могли писати свої власні скрипти, щоб розширювати функціонал бота. Крім того у нього була функція за допомогою якої можна було утворити зв'язок з іншими Eggdrop-ботами, це дозволило покращити контроль і безпеку на досить великих майданчиках IRC-мереж. Також був доступний список користувачів, список заборонених, список запрошених/звільнених та список ігнорованих; все це (якщо існує зв'язок з іншими ботами) могло бути загальним для всіх ботів. [5]. Хоча й Eggdrop був першим ботом але він не використовувався для зловмисних цілей, але дав великий крок у формуванні та створенні ботнетів.

У 1998 році після Eggdrop з'являється перший bot направлений на інфікування комп'ютера для подальших зловмисних цілей, він називався GTbot. Хакер передавав GTbot через IMC чат під видом троянської програми використовуючи власні навики соціальної інженерії. Після того як жертва запускала переданий файл розпочиналося інфікування, GTbot інсталювався на комп'ютер жертви і приєднувався до певного IRC-каналу и хакер міг управляти їм. Через цей канал, зловмисник міг надсилати команди всім підключеним ботам, виконуючи координовані дії, такі як атаки на сервери або мережі. Через відносно просту структуру та відсутність складного шифрування, GTbot та подібні до нього ботнети могли бути відносно легко виявлені та видалені з

використанням антивірусного програмного забезпечення. Далі починається фаза великих ботнетів.

На початку 2000х з'являється Gaobot також відомий як Agobot. Даний Botnet став відомим завдяки своїм різноманітним функціям та можливостям, які значно перевищували можливості багатьох інших ботнетів того часу. Він міг виконувати широкий спектр зловмисних дій, включаючи DDoS-атаки, крадіжку даних, завантаження та встановлення додаткового шкідливого ПЗ (Програмного забезпечення), злом паролів і багато іншого. Методом контролю все ще міг виступати IRC чат але в ньому вперше з'являється P2P з'єднання що полегшувало комунікацію з інфікованими машинами , що робило його більш стійким до спроб виявлення та відключення. Його поява підняла планку для антивірусного та іншого захисного програмного забезпечення, спонукаючи до розвитку більш складних методів виявлення та нейтралізації шкідливого ПЗ.

До 2010их років Botnets стрімко розвиваються, хакери відмовляються використовувати IRC мережі та починають використовувати P2P архітектуру що робило їх стійкішими до спроб відключення, оскільки вони не покладалися на єдині точки відмови, як це було у ботнетів з централізованою командно-контрольною структурою. Також вони починають використовувати поліморфічний код та техніки обфускації. Як це працювало:

- перейменовувалися назви змінних, класів, функцій на незрозумілі або випадкові символи;
- додавалася частина коду яка не впливала на функціональність програми, але ускладнювала його аналіз;
- використовувалися складні умовні оператори або цикли для заміни прямого потоку виконання, роблячи логіку програми не очевидною.

Також поширилося автоматичне оновлення Botnets. Найвідоміший ботнети цих часів: Zeus.

Zeus або Zbot був одним із найбільш руйнівних шкідливих програм. Він був розроблений для крадіжки банківської інформації, такої як логіни, паролі. Використовував він кейлогінг інформації, такої як логіни, для вилучення

конфіденційної інформації користувачів. Розповсюджувався він через фішинг та шкідливі посилання. Zeus завдав збитків на мільйони доларів, оскільки через свою простоту він був досить популярним серед кіберзлочинців.

В наші часи Botnets продовжують еволюціонувати, стаючи більш складними і небезпечними. Технологічний розвиток, особливо у сферах штучного інтелекту, машинного навчання та інтернету речей (IoT), надав нові можливості для зловмисників.

IoT став гарним підґрунтям для розвитку Botnets оскільки розробляється дуже велика кількість речей яка має підключення до Інтернету, яка в той же час немає стандартизованого рівня захисту, що робить їх гарною метою для хакерів. Одним із найвідоміших ботнетів який базувався на IoT пристроях став славнозвісній botnet Mirai. Mirai сканував інтернет у пошуках IoT пристроїв, що використовували однаковий незмінний логіни та пароль установлений розробником. Усього він використовував 61 пару логін-пароль для доступу до облікового запису шляхом перебору[8]. Після успішного входу, Mirai інфікував пристрій шкідливим Програмним забезпеченям і залучав його до ботнету, який в свою чергу використовувався хакером.

1.4.2 Загальні відомості про Botnet

Під час огляду історії створення Botnet-у можна надати таке визначення Botnet – це розподілена атакуюча мережа, яка складається з підключених до Інтернету пристроїв, кожен з яких є ботом. Ці пристрої які можуть включати в себе: комп'ютери, телефони та пристрої IoT (Інтернету речей).

Будують Botnets з різних причин: для розсилки спаму, крадіжки даних, крипто-валютного майнінгу та інші. Але в основному їх використовують для DDoS атак оскільки, у цьому виді атак важливо створити велику кількість хостів, яка буде навантажувати цільовий сервіс (комп'ютер, сервер) трафіком.

Щоб побудувати ботнет, зловмисник шукає комп'ютери які погано захищені. Як правило уразливий хост може бути скомпрометований у декілька способів:

- Відправкою шкідливих електронних листів (Phishing attack)

Злодмисники відправляють електронні листи, які виглядають, ніби вони відправлені від імені довірених джерел, наприклад, банків, соціальних мереж або компаній. Ці листи містять вкладення або посилання на шкідливий код. Жертва, відкривши такий лист, інфікує свій пристрій через це код в слід чого він може стати частиною ботнету.

- Використовують соціальну інженерію

Зловмисники можуть використовувати техніки соціальної інженерії, щоб ввести жертву в оману та отримати доступ до її пристрою. Наприклад, вони можуть надавати себе за технічну підтримку і вимагати доступ до системи.

- Використовують віруси типу “хробаки”

Зловмисники можуть створювати віруси та розміщувати їх в Інтернеті які потім можуть бути завантажені на пристрій. Після завантаження, хробак сканує пристрій на вразливості. Далі зловмисник використовує ці вразливості, щоб додати цей пристрій до ботнету.

Звісно зловмисник може використовувати декілька методів одночасно, а або комбінувати їх. Після інфікування та встановлення програмного забезпечення зловмисник може використовувати ці пристрої як забажає. На рис. 1.4 зображений ботнет.

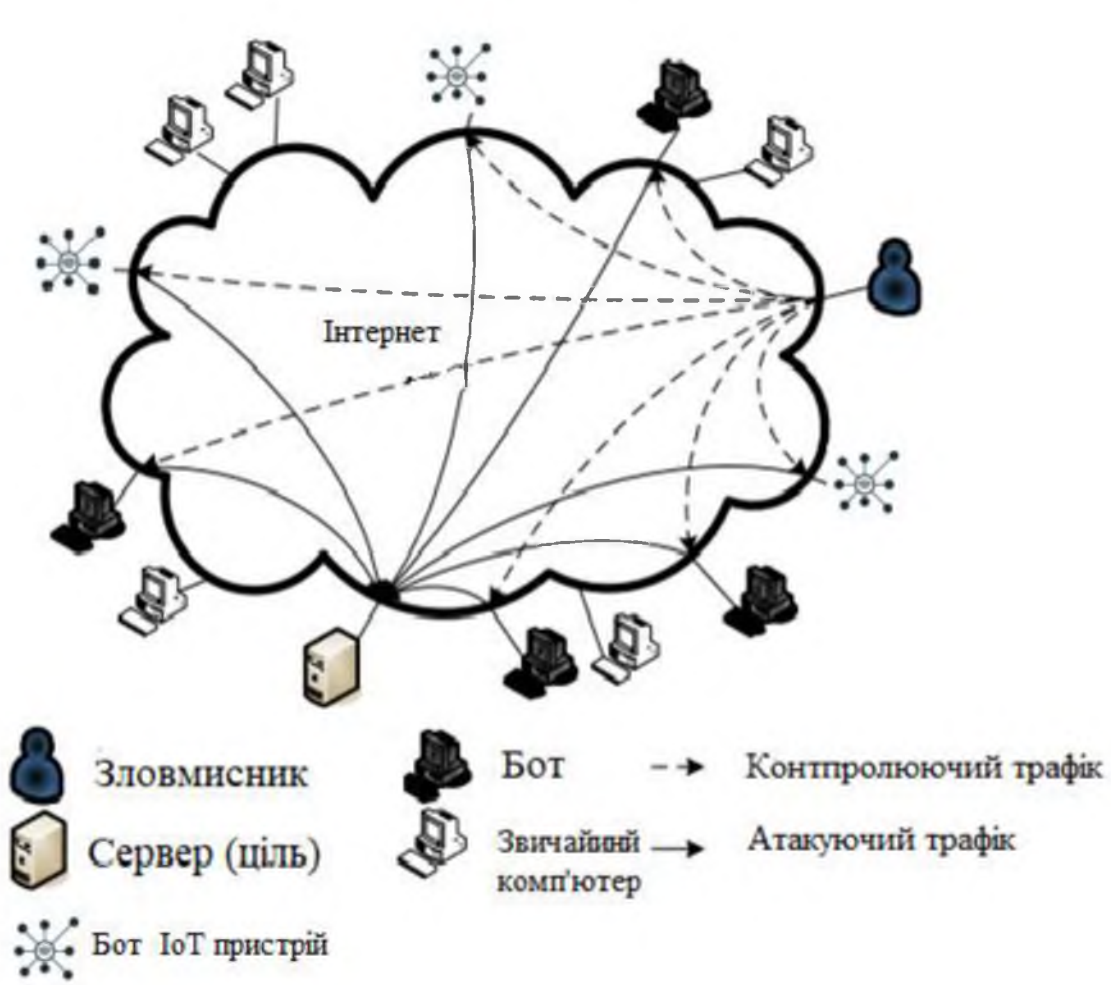

Рисунок 1.4 – Ботнет

1.5 Види DDoS атак

1.5.1 HTTP flood

Найрозповсюдженішим компонентом мережі Інтернет є www-програми, які як правило, використовують HTTP протокол (протокол передачі гіпертекстових документів), який ініціює запит шляхом встановлення TCP (Протоколу керування передачею) з'єднання на певний порт (як правило 80 порт). Через це більшість захисного програмного забезпечення залишає TCP порт 80 відкритим для вільного проходження HTTP трафіку. Широка розповсюдженість WWW систем зробила HTTP чудовою мішенню для нападників. Основна ідея, атаки HTTP flood полягає в тому, щоб перевантажити сервер великою кількістю законних HTTP запитів, таким чином перешкоджаючи його здатності обслуговувати легітимний трафік.

Для того щоб надіслати HTTP запит необхідно спочатку встановити дійсне TCP з'єднання, що вимагає наявності дійсної IP адреси. Нападник організовує ботнет, який буде використовуватися для генерування та відправки великої кількості HTTP запитів до цілі.. Ці запити можуть включати GET або POST запити до сервера.

Більше того, нападник може сконструювати особливий запит для збільшення сили атаки або для того, щоб уникнути виявлення. Наприклад, видати боту інструкцію надіслати HTTP запит на завантаження великого файлу. Тоді комп'ютеру жертви необхідно зчитати цей файл з диску, завантажити у пам'ять, загрузити в пакети та надсилати їх у ботнет. Отже, навіть простий HTTP запит може привести до значного завантаження процесору, пам'яті, пристроїв вводу/виводу та лінії з'єднання з Інтернет.

Однак така поведінка трафіку є підозрілою. Повторні запити на завантаження можуть бути виявлені та блоковані. Нападник може вдатися до стратегії з імітації звичайного трафіку. Він задає ботам програму пересилки HTTP запитів, аналізу відповідей і рекурсивного відкриття посилань. В цьому разі HTTP запити від атакуючої мережі дуже подібні до нормального веб трафіку, що ускладнює фільтрацію атак цього типу.

1.5.2 UDP flood

UDP-flood– це одна з видів DDoS атак мета якої є перевантаження каналу передачі даних цілі великою кількістю UDP (протокол датаграм користувача) пакетів, щоб зробити її недоступною для нормального трафіку та користувачів.

UDP-флуд працює в декілька кроків, розглянемо їх зі сторони сервера, коли він отримує UDP-пакет. За звичайних умов, коли сервер отримує UDP-пакет на певному порту, він проходить два кроки у відповідь:

1. Сервер спочатку перевіряє, чи запущені будь-які програми, які зараз прослуховують запити на порту, який повинен отримати пакет.

2. Якщо жодна програма не отримує пакети на цьому порту, сервер відповідає пакетом ICMP (ping), щоб повідомити відправника про те, що адресат недоступний.

UDP-флуд можна розглядати в контексті маршрутизації дзвінків портьє готелю. Спочатку на портьє надходить телефонний дзвінок, у якому той, хто дзвонить, просить підключити його до конкретної кімнати. Потім портьє має переглянути список усіх кімнат, щоб переконатися, що гість вільний у кімнаті та бажає прийняти дзвінок. Коли адміністратор зрозуміє, що гість не приймає дзвінки, він повинен знову підняти трубку та повідомити абоненту, що гість не прийматиме дзвінок. Якщо раптом всі телефонні лінії засвітяться одночасно з подібними запитами, то вони швидко перевантажаться. [9] Схематична атака UDP Flood зображена на рис.1.5:

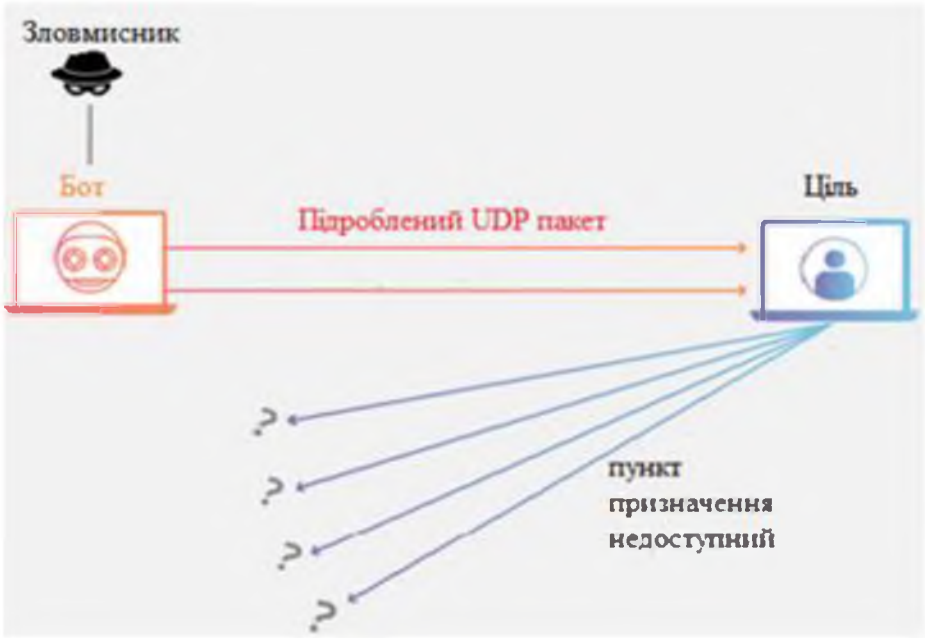

Рисунок 1.5 – Робота атаки типу UDP flood[9]

Оскільки кожен новий UDP-пакет отримує сервер, він проходить певні кроки для обробки запиту, використовуючи ресурси сервера в процесі. Під час передачі UDP-пакетів кожен пакет міститиме IP-адресу вихідного пристрою. Під час DDoS-атаки цього типу зловмисник, як правило, не використовує свою власну справжню IP-адресу, а натомість підробляє IP-адресу джерела UDP-пакетів, перешкоджаючи розкриттю справжнього місцезнаходження зловмисника та потенційному насиченню відповідними пакетами від цільової мережі.

Оскільки цільовий сервер використовує ресурси для перевірки та відповіді на кожен отриманий UDP-пакет, ресурси цільової цілі можуть швидко вичерпатися, коли надходить велика кількість UDP-пакетів, що призводить до відмови в обслуговуванні для нормального трафіку.

1.5.3 TCP SYN flood

Даний вид атаки, як видно з назви, використовує недолік роботу протоколу TCP, який направлений на процес встановлення TCP з'єднань, а саме трьох-фазного рукостискання (three-way handshake), між пристроями.

Цей процес протікає у 3 етапи та зображений на рисунку 1.6:

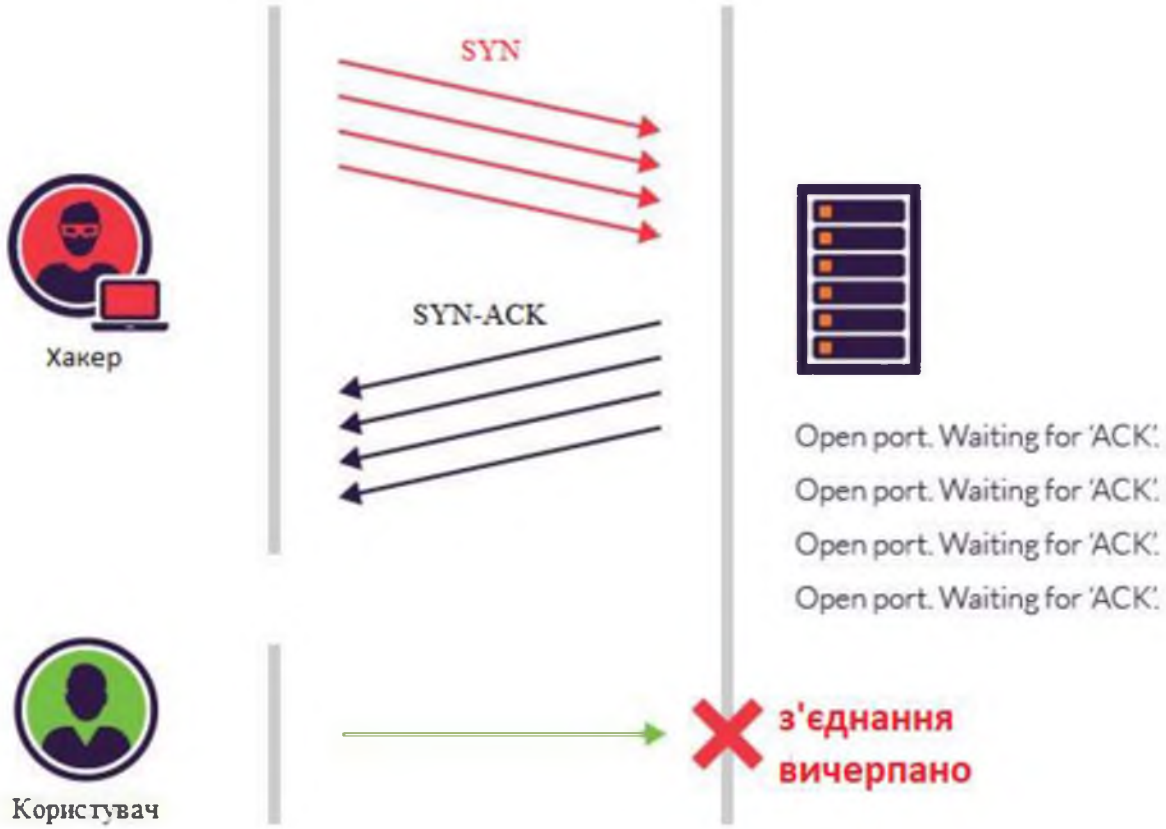

Рисунок 1.6 – Схема роботи атаки типу TCP SYN flood

- Зловмисник починає процес встановлення з'єднання, відправляючи TCP SYN (Syn – скорочено синхронізація) пакет до цільового сервера. Цей пакет сигналізує про бажання встановити з'єднання.
- Сервер обробляє SYN запит і відправляє назад SYN-ACK (SYN-ACK скорочено синхронізація-підтверджена) пакет, щоб підтвердити отримання SYN і продовжити процес встановлення з'єднання.

- Зазвичай, клієнт має відправити ACK пакет, щоб завершити процес встановлення з'єднання. Однак, під час SYN Flood атаки, ACK пакет ніколи не відправляється. Замість цього, зловмисник продовжує відправляти велику кількість SYN пакетів, часто з різних спотворених або підроблених IP-адрес.

Унаслідок даної атаки сервер зберігає недобудовані сесії в очікуванні закінчення з'єднання. Це веде до вичерпання ресурсів, таких як пам'ять та процесорний час, оскільки сервер намагається обробити всі ці напіввідкриті з'єднання. У результаті, законні запити від користувачів направленні для встановлення з'єднань можуть бути відхилені або ігноровані через недостатність ресурсів, що призводить до недоступності сервісів.

1.5.4 PING ICMP flood

Зловмисник в цьому виді атак, використовує ICMP протокол для створення великого навантаження на цільову систему або мережу. Ця атака використовує простий, але ефективний механізм для перевантаження ресурсів цілі, роблячи її недоступною для легітимного трафіку.

ICMP - це фундаментальний протокол, використовуваний в Інтернеті для надсилання повідомлень про помилки та оперативного контролю. Однією з основних функцій ICMP є можливість відправки "echo request" та отримання "echo reply" для перевірки доступності мережевих ресурсів.

Атака передбачає заповнення мережі жертви великою кількістю пакетів запитів типу “echo request” використовуючи команду “ping”. Ці пакети направляються на IP-адресу цільової системи. Тоді сервер або мережевий пристрій відповідає на кожен "echo request" відповідним "echo reply". Це стандартна поведінка мережі для перевірки доступності та зв'язку. Це напружує як вхідні, так і вихідні канали мережі, споживаючи значну пропускну здатність і призводячи канали зв’язку до відмови в обслуговуванні.

Оскільки "echo request" пакетів надходить значно більше, ніж система може ефективно обробити, вона починає витрачати свої ресурси (обчислювальну потужність, мережеву пропускну спроможність) на їх обробку.

Це може призвести до значного сповільнення в роботі системи, затримок в обробці легітимного трафіку або навіть повного збою.

Ефективність ping flood залежить від того, чи знають зловмисники IP-адресу своєї мети. Таким чином, атаки можна розділити на три категорії залежно від цілі та способу розв'язання її IP-адреси.

- Націлений на один комп'ютер у локальній мережі. Зловмиснику потрібен фізичний доступ до комп'ютера, щоб дізнатися його IP-адресу. Успішна атака призведе до зняття цільового комп'ютера. [11]
- Націлений на маршрутизатори, щоб порушити зв'язок між комп'ютерами в мережі. Він залежить від того, що зловмисник знає внутрішню IP-адресу локального маршрутизатора. Успішна атака призведе до зняття всіх комп'ютерів, підключених до маршрутизатора. [11]
- Сліпа атака передбачає використання зовнішньої програми для виявлення IP-адреси цільового комп'ютера або маршрутизатора перед виконанням атаки. [11]

Важливо зазначити, що для оптимальної підтримки ping flood атакуючий комп'ютер повинен мати доступ до більшої пропускної здатності, ніж жертва. Атака даного виду буде не ефективна проти великої мережі.

1.5.5 Атака фрагментованими пакетами

Атаки з фрагментацією IP-адресів є поширеною формою атаки типу «відказу в обслуговуванні», при якому зловмисник контролює мережу, використовуючи механізми фрагментації – датаграми.

Щоб зрозуміти як працює дана атака необхідно зрозуміти процес IP фрагментації, це процедура зв'язку при якій датаграми IP розбиваються на невеликі пакети, передаються по мережі, а потім знову збираються в вихідну інформацію.

Фрагментація необхідна для передачі даних, оскільки кожна мережа може обробляти свій розмір датаграм. Цей розмір відомий як максимальна одиниця передачі (MTU). Якщо відправляється датаграмма перевищує MTU, що приймає сервер, вона повинна бути фрагментована для повної передачі. Побачити як відбувається фрагментація пакетів можна на рис. 1.7:

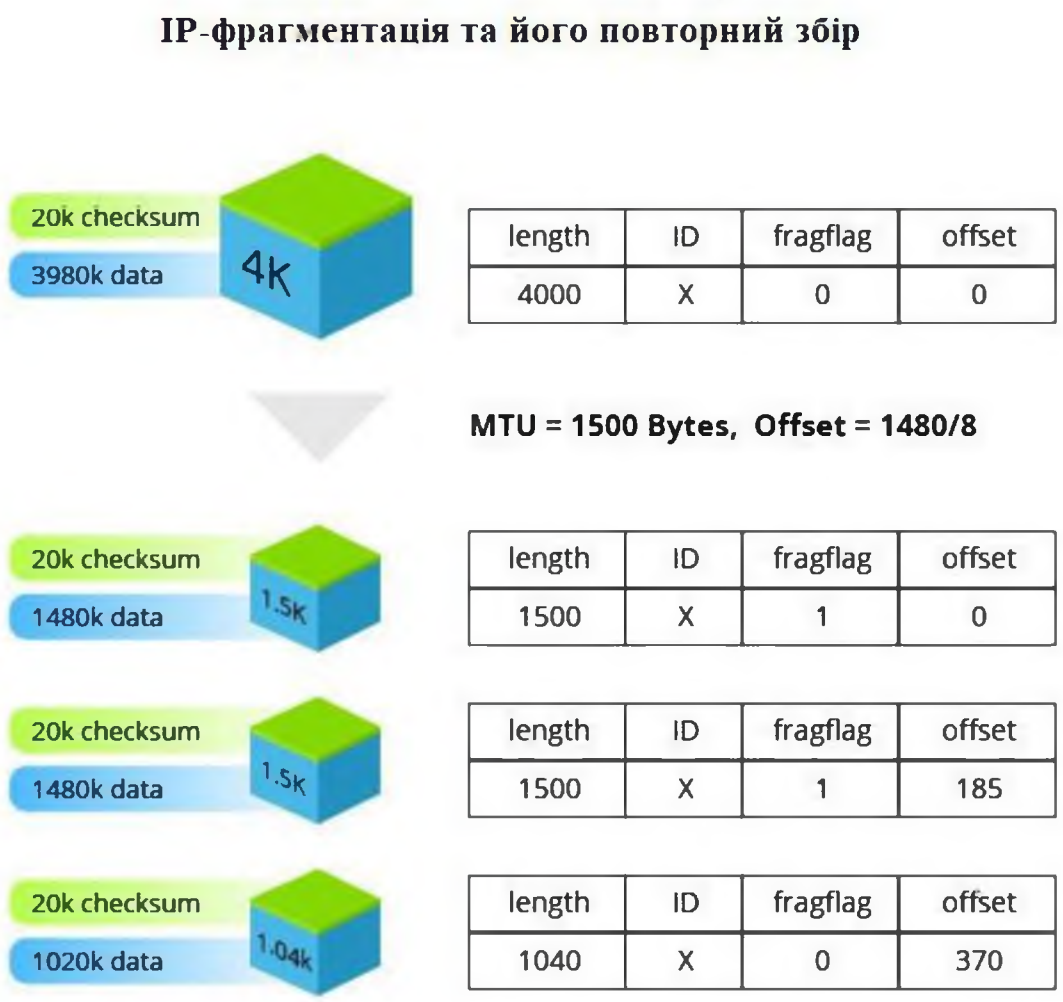

Рисунок 1.7 – Приклад IP фрагментациї [11]

Length – розмір фрагментованої датаграми.

ID – ID датаграми яку фрагментували.

Fragflag – показує чи є ще фрагменти.

Offset – Показує в якому порядку фрагменти повинні розташовані під час зборки.

Заголовок IP у кожній датаграмі містить прапорці, які вказують, чи дозволяється фрагментація. У випадках, коли до IP-заголовка додається прапор «не фрагментувати», пакет відкидається, і сервер надсилає повідомлення про те, що датаграма ICMP завелика для передачі.

Атаки фрагментації IP можуть мати кілька форм. Хоча всі вони використовують ділення датаграм, щоб пересилати цільові мережі, існують деякі помітні відмінності в тому, як виконуються різні вектори атаки.

Підвидом даної атаки є TCP fragmentation attacks (TearDrop)

Серверу-жертві відправляються фрагментовані пакети, які він не може розібрати через вразливість TCP/IP. Пакети накладаються один на одного, перевантажуючи сервер-жертву. Одне з полів IP-заголовка “Fragment Offset” вказує на початкову позицію чи усунення даних у фрагментованому пакеті щодо реального пакета. Якщо сума зсувів та розміру одного фрагментованого пакета відрізняється від наступного фрагментованого пакета, пакети накладаються. Коли це відбувається, сервер не в змозі розібрати вхідні пакети і виходить з ладу.

1.6 Правова база

Захист інформації є критично важливим аспектом для будь-якої організації, що використовує цифрові технології та інтернет, але навіщо захищатися саме від DDoS атаки. Звернемося до закону України “Про інформацію” та дамо визначення терміну захист інформації. Захист інформації - сукупність правових, адміністративних, організаційних, технічних та інших заходів, що забезпечують збереження, цілісність інформації та належний порядок доступу до неї;[12]

Виходячи з визначення, DDoS атака, у розрізі даного закону, має великий вплив саме на порядок доступу (або доступність), оскільки під час даного типу атак зловмисники вдаються до перевантаження інфраструктури цілі, що може вивести з ладу критичні для бізнесу сервіси які не можуть надавати доступ до своїх інформаційних ресурсів законним користувачам.

Крім того DDoS атаки хоч і переважно спрямовані на порушення доступності, вони також можуть вплинути на цілісність даних. У ситуаціях, коли системи перевантажені, може статися помилкова обробка даних, що може призвести до їх спотворення або втрати. Крім того, під час атаки ресурси IT команди часто зосереджені на відновленні доступності, що може зменшити увагу до моніторингу цілісності даних.

Якщо продовжити розбирати закон України "Про інформацію" то можна побачити що за статтею 2: "Основними принципами інформаційних відносин є: гарантованість права на інформацію, відкритість, доступність інформації, свобода обміну інформацією, достовірність і повнота інформації…."[12]. З чого можна зробити висновок, що як керівника підприємства, збереження трьох принципів захисту інформації є важливим завданням.

Також якщо подивитися на статтю 9 закону України " Про захист інформації в інформаційно-комунікаційних системах" то можна дізнатися що: " Відповідальність за забезпечення захисту інформації в системі покладається на власника системи. Про спроби та/або факти несанкціонованих дій у системі щодо державних інформаційних ресурсів або інформації з обмеженим доступом, вимога щодо захисту якої встановлена законом, власник системи повідомляє відповідно спеціально уповноважений центральний орган виконавчої влади з питань організації спеціального зв'язку та захисту інформації або підпорядкований йому регіональний орган"[13]. З чого можна зробити висновок, що буде відкрите кримінальне впровадження на власника системи якщо він не буде підтримувати надійний захист системи.

1.7 Методи захисту від DDoS атак

Глобально методи захисту можна поділити на дві групи: методи для запобігання і активні методи захисту.

До методів запобігання можна віднести розробку політик і процедур. Даний метод допомагає у структуруванні підходів захисту інфраструктури та забезпечує чіткі кроки для реагування на інциденти. В даному методі

аналізуються ризики, розробляються політики захисту цілей та процедури які будуть виконуватися під час атаки.

Також, на стадії запобігання, відбувається ліквідація вразливостей, підтримка обладнання та програмного забезпечення в оновленому стані. Багато мережевих атак цілеспрямовано використовують різні види вразливостей, наприклад, в програмному забезпеченні серверів або у використанні неефективних програмних скриптів, що можуть надмірно навантажувати ресурси сервера.

Під час атаки починають використовувати активні методи захисти одним із основних методів захисту є нарощування ресурсів та фільтрація трафіку.

Цей підхід полягає в збільшенні того елемента обробляючого елемента системи куди планується атак. Це може бути: обчислювальна потужності, пропускної спроможність мережі.

Підвищення пропускної спроможності мережі може допомогти витримати збільшений об'єм трафіку, який є характерним для DDoS атак. Більша пропускна спроможність дозволяє обробляти легітимні запити, навіть коли система зазнає атаки. Для даного методу потрібно мати вже заготовлені засоби розширення. Наприклад якщо відбувається атака на сервер на якому розміщена деяка база даних то потрібно створити заздалегідь сервери з розміщених на ній базою даних, які можна підключити в момент атаки.

У даного методу захисту є свої недоліки:

- нарощування ресурсів пов'язано зі зміною апаратного комплексу тому потрібна підготовка для оперативного підключення в момент початку атаки;
- підтримання надлишкових ресурсів економічно недоцільно в період очікування атаки.
- нарощувати ресурси можуть і зловмисники підключивши більший ботнет, використовуючи різноманітні скрипти для більш ефективного завантаження системи;

- в деякий момент час використання цього методу стане економічно недоцільним.

Для подолання недоліків оперативного підключення та підтримки ресурсів може допомогти Хмарні технології. Нам не потрібно буде займатися підготовкою для підключення додаткових ресурсів, оскільки хостинг провайдер буде надавати їх у разі збільшення навантаження на свій ресурс. Це можуть бути як легітимні користувачі так і шкідливі запити. Не вирішує це тільки економічної проблеми, оскільки зі збільшенням трафіку буде рости і ціна за хмарні технології.

Нарощування ресурсів не вирішує проблему DDoS атак безпосередньо, але створює запас часу, що дозволяє інфраструктурі протриматися деякий час витримуючи все більші та більші навантаження, щоб забезпечити безперервність бізнес-процесів під час атак.

Фільтрація трафіку направлена на нейтралізації атак у реальному часі. Вона включає в себе використання різних інструментів та стратегій, які діють безпосередньо на мережевий трафік, щоб ідентифікувати та блокувати атаки.

Для побудови ефективної системи фільтрації трафіку потрібно вирішити дві проблеми: дізнатися коли починається атака та визначити джерело атаки.

Для виявлення атаки застосовують моніторинг і аналіз даних та трафіку. Це допомагає збирати дані про обсяги трафіку, джерело, призначення, типи, протоколів і т.д. Збирають дану інформацію з журналів серверів, фаєрволів, маршрутизаторів та інших мережевих пристроїв.

Щоб локалізувати джерело атаки використовують два підходи: аналіз зловживань та аналіз аномалій. Метод аналізу зловживань, для виявлення атаки, здійснює порівняння поточних системних даних з даними, характерними для звичайних атак. Альтернативно, аналіз аномалій порівнює поточний стан системи з її звичайним станом.

Порівняння даних підходів зображені у таблиці 1.1:

Таблиця 1.1 – Порівняння критеріїв двох підходів захисту

Критерій	Підхід зловживань	Підхід аналізу аномалій
Виявлення відомих атак	Висока точність	Може пропустити без змін у поведінковому шаблоні
Виявлення нових атак	Неефективний	Ефективний
Швидкість виявлення	Швидкий	Повільний
Помилкові спрацьовування	Низькі	Високі у порівнянні з підходом до зловживань
Адаптивність	Обмежена	Висока
Оновлення обслуговування	Потрібні регулярні оновлення сигнатур	Потребує постійного налаштування та моніторингу

Як видно з таблиці 1.7 для оптимального захисту потрібно застосовувати обидва підходи.

Пристрій який може дізнаватися коли починається атака та визначати джерело переглядаючи увесь трафік системи є IDS система.

IDS – система виявлення вторгнень, призначена для реєстрації підозрілих дій у мережі, яка повідомляє про них відповідального за інформаційну безпеку співробітника за допомогою передачі повідомлення на консоль управління, надсилання електронного листа тощо

Традиційна IDS складається з сенсорів, які переглядають мережевий трафік чи журнали та передають аналізаторам, аналізатори шукають в отриманих даних шкідливий характер і у разі успішного виявлення – надсилає результати до адміністративного інтерфейсу. Залежно від розташування IDS поділяються на мережеві NIDS і хостові HIDS. За назвою зрозуміло, що одна відстежує весь мережевий трафік того сегмента, де вона встановлена, а інша в межах єдиного комп'ютера.[14]

Шкідливу активність у аналізованому трафіку можна виявляти різними способами. Тому в IDS існують такі характеристики, що відрізняються один від одного різні типи технологій IDS і описати їх можна так:

Сигнатурні IDS. Відстежують певні шаблони в трафіку і працюють подібно до антивірусного програмного забезпечення. Є два недоліки цього підходу: сигнатури повинні бути в актуальному стані і IDS такого типу не здатні виявити незнайомі атаки. Дану категорію можна розділити на два типи, як саме вони працюють: сигнатурні IDS, що відстежують шаблони – порівнюють мережні пакети з сигнатурами, а відстежуючи стан – порівнюють дії з шаблонами.

IDS, засновані на аномаліях. Цей тип заснований на поведінці системи та перед початком роботи відбувається етап навчання «нормальної» діяльності системи. Тому вона здатна виявляти незнайомі атаки. Аномалії, у свою чергу, в даній категорії діляться на три типи: статистичні - IDS створює профіль штатної діяльності системи і порівнює весь трафік і діяльність з цим профілем; аномалії протоколів – IDS аналізує трафік для виявлення фрагментів нелегітимного використання протоколів; аномалії трафіку – IDS виявляє нелегітимні дії у мережевому трафіку.

Після того як відбулася ідентифікація атаки і локалізоване джерело можна перейти до блокування трафіку. Створюються чорні та біли списки користувачів.

- Чорні списки дозволяють блокувати певний трафік, який вважається шкідливим.
- Білі списки блокує весь трафік крім включених в нього трафік.

1.8 Машинне навчання

1.8.1 Чому машинне навчання є актуальним для протидії DDoS атакам

З попереднього пункту можна зробити висновок що основною задачею в виявленні DDoS атаки є локалізація джерела та перегляд великої інформації, щоб знайти це джерело. А за допомогою машинного навчання можна вирішити ці обидві проблеми. Оскільки машинне навчання може автоматизувати

процеси: моніторингу, аналізу трафіку також аналізує тенденції мережевому трафіку, що забезпечує автоматичне оновлення системи аналізу.

1.8.2 Загальні відомості про машинне навчання

ML() - це галузь штучного інтелекту (AI), яка фокусується на розробці алгоритмів та технологій, що дозволяють комп'ютерам вчитися з даних та використовувати їх для навчання та прийняття рішень або прогнозів. Прогнози в даному випадку генеруються без додавання даних до системи, явним програмуванням, а отриманих системою в ході її роботи. На начальному етапи коли система ще не навчена їй можуть надаватися датасети – набори даних, які використовується для тренування, тестування та валідації. Вирішальну роль у виборі датасету грають розмір та якість даних, оскільки від них залежить які рішення буде приймати модель.

Також є різні методи навчання та функції за якими машина буде навчатися. Доречі, при використанні алгоритмів використовують засоби чисельних методів, статистики, теорію графів, теорію ймовірності, різноманітні оптимізаційні методи, а також методики для роботи цифровими даними.

Розрізняють такі методи навчання:

- Навчання з вчителем.

Навчання з учителем (Supervised Learning) є одним із основних напрямків у галузі машинного навчання. Цей підхід передбачає використання розмічених даних для тренування моделі, де кожному вхідному елементу даних відповідає заздалегідь відомий вихідний результат або мітка. Цей метод є особливо ефективним для задач, де потрібно здійснювати прогнозування або класифікацію на основі існуючих прикладів.

- Навчання без вчителя.

Відрізняється від навчання з вчителем відрізняється тим, що дані, які використовуються для тренування, не мають заздалегідь визначених міток або відомих відповідей. Замість цього, алгоритми навчання без учителя намагаються виявити приховані шаблони, структури або взаємозв'язки безпосередньо з нерозміченими даними.

- Напівнаглядове навчання.

Це підвид навчання де алгоритми тренуються на основі взаємодії з довкіллям і зосереджені на виборі послідовності дій, які максимізують сумарну винагороду. Машина не має попередньої інформації про середовище, але має можливість здійснювати в ній будь-які дії. Середовище реагує на ці дії і тим самим надає агенту дані, які дозволяють йому реагувати на них і вчитися. Фактично машина і середовище утворюють систему зі зворотним зв'язком. Навчання з підкріпленням дуже схоже на реальне навчання людей – машину карають за помилки і заохочують за правильні вчинки. Його використовують там, де перед машиною стоїть завдання – правильно виконати поставлені завдання у зовнішньому середовищі маючи безліч можливих варіантів дії.

1.8.3 Типові завдання для машинного навчання

- Регресія

Моделі регресійного аналізу зазвичай використовують, щоб показати або передбачити взаємозв'язок між процесом та тим, що цей процес може спровокувати.

- Класифікація

Це процес пошуку моделі, яка допомагає розділити дані на різні категоріальні класи. У цьому процесі дані класифікуються під різними мітками відповідно до деяких параметрів, наведених на вході, і тоді мітки прогнозуються для даних. Вихідною змінною в цьому разі є категорія, тобто алгоритми класифікації дозволяють розділити об'єкти відповідно до зазначених заздалегідь класів.

- Кластеризація

Розподіл на однорідні підмножини за деякою властивістю, коли дані неможливо точно класифікувати.

- Прогнозування

Вирішують завдання передбачення часових рядів – знаходження майбутніх значень залежно від часу

- Виявлення аномалій

Процес пошуку експериментальних даних, які суттєво відрізняються від інших. Вони виникають як наслідок збою вимірювальних приладів, описок операторів, помилкової конвертації даних тощо.

1.8.4 Загальні етапи побудови ML

- Збір та Підготовка Даних

Дані є основою машинного навчання. На цьому етапі збираються дані, які будуть використовуватися для тренування моделі. Дані можуть бути зібрані з різних джерел і можуть вимагати очищення, оскільки вони можуть містити шуми та не мати чіткої структури.

- Вибір ознак

На цій стадії з великої кількості доступних ознак вибираються лише ті, що важливі для навчання. Датасети часто включають багато ознак, іноді їх число досягає сотень або навіть тисяч. У створенні моделей машинного навчання не завжди очевидно, які ознаки є значущими (мають зв'язок з цільовою змінною) та які є зайвими або носіями шуму. Видалення непотрібних параметрів допомагає дослідникам краще розуміти дані, що скорочує час налаштування моделі та покращує точність її прогнозів. Часом, знаходження оптимального набору ознак може бути ключовим завданням, оскільки це сприяє розумінню механізмів, що лежать в основі досліджуваної проблеми.

- Вибір Моделі

На основі зібраних даних та типу задачі вибирається підходяща модель або декілька моделей для експериментів. Вибір моделі може залежати від розміру даних, точності, що потрібна, та обчислювальних ресурсів, які є у розпорядженні. Для будь-якої задачі машинного навчання можна застосувати численні алгоритми та створити декілька моделей. Наприклад, проблему класифікації для виявлення спаму можна вирішити за допомогою різноманітних моделей, таких як: наївний баєс, логістичну регресію та інші.

- Тренування Моделі

На цьому етапі модель тренується на підготовлених даних. Під час тренування алгоритм машинного навчання намагається знайти шаблони або закономірності, які відповідають за прогнозування або класифікацію.

- Оцінка Моделі

Після тренування модель оцінюється на тестовому наборі даних, який не був використаний під час тренування. Це допомагає зрозуміти, наскільки добре модель узагальнює дані та працює з новими, невідомими прикладами. На рисунку 1.8 зображена схема побудови моделі ML

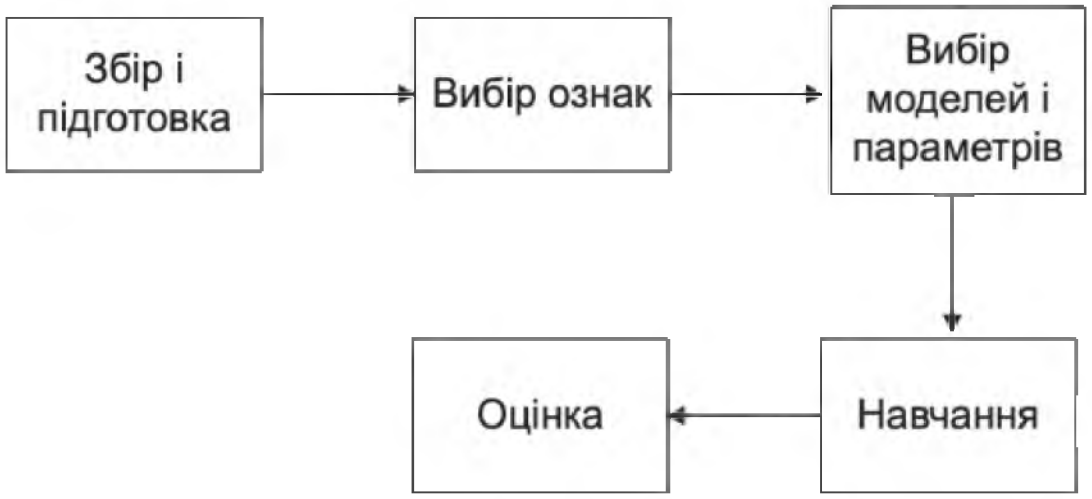

Рисунок 1.8 – Модель побудови ML

1.9 Висновки

В цьому розділі було розібрана природа DDoS атаки, та основні методи, які використовуються для їх реалізації. Було детально розглянуто їх визначення, ключові відмінності та способи реалізації. Основна увага приділялася методам, якими зловмисники використовують численні компрометовані хости для створення масованого трафіку, оскільки це головна відмінність DDoS атаки від DoS.

Була розглянута модель OSI, тому що в контексті DDoS атак, вона визначає стандартні рівні комунікації в мережах, які зловмисники використовують для своїх атак.

Далі було деталізовано різні види DDoS атак, а саме: HTTP flood, UDP flood, TCP SYN flood та PING ICMP flood, розглянуто механізми їх дії та потенційний вплив на цільові системи.

Крім того, був описаний botnet - ключовий інструмент для проведення DDoS атак. Починаючи з базового botnet Eggdrop, можна прослідити як вони стають все більш комплексними та витонченими, застосовуючи різні способи інфікування та контролю, що дало зловмисникам нові можливості у створенні нових ботів

Машинне навчання полегшує процес виявлення, класифікації та кластеризації інтернет трафіку системи. Надано його поверхневий аналіз який надає розуміння механізмів його роботи, Що є критично важливим для розробки ефективних стратегій їх виявлення та запобігання DDoS атак.

1.10 Постановка задачі

Метою даного дослідження є створення системи, яка аналізує запити до сервера для ідентифікації DDoS-атак. Виходячи з цього, можна визначити наступні завдання для дослідження в рамках спеціальної частини:

- Ознайомлення з алгоритмами машинного навчання для задачі класифікації.
- Зібрання тренувального та тестувальних наборів даних для навчання.
- Обрати засоби для програмної реалізації.
- Розроблення архітектуру програми, в якій будуть реалізовані обрані алгоритми ти проаналізовано отримані результати.
- Спираючись на отриманні результати проаналізувати їх та зробити висновки про проведену роботу.

РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА

2.1. Задача класифікації

Для ідентифікації трафіку підходить задача класифікації машинного навчання, оскільки класифікувати об'єкт – означає надати об’єкту ідентифікаційний номер класу, до якого це об’єкт відноситься. Нам дано кінцевий набір об'єктів і для якого ми знаємо класи, до яких належить кожен член цього набору. В нашому випадку він називається датасетом.

Далі за допомогою цього датасету потрібно побудувати певний алгоритм, який буде здатний виділяти за певним набором ознак, який знаходиться в датасеті, зловмисний трафік. Для цього нам потрібно зробити математичну модель ознак за якими ми зможемо створити датасет.

2.2. Математична модель

Для ефективної розробки алгоритму виявлення DDOS атак на основі машинного навчання спершу необхідно розглянути математичну модель виявлення із якою необхідно буде мати справу.

Нехай дано деяку множину вхідних даних X і є множина відповідей Y. Множина відповідей Y позначає приналежність даного прикладу до атаки. Є деяка навчальна вибірка $\{x1, \ldots, xn\} \subset X$ та вибірка відповідей $\{y1, \ldots, yn\} \subset Y$.

Задачею виявлення атак є знаходження такого визначаючого правила(, що дає найкраще наближення до правильних відповідей на всій множині X:

$a: X \rightarrow Y$

$xi \subset X$ в даному випадку набором ознак є дані про трафік,

Ознаки поділяються на типи:

булеві, область значень яких {0, 1};

номінальні, область значень яких – скінченна підмножина N;

порядкові, що представляють собою номінальні ознаки, для яких визначено лінійний порядок;

кількісні, значенням яких є дійсне число

Прикладом булевої ознаки є належність певного прикладу до вхідного(1) або вихідного(0) трафіку.

номінальні, область значень яких – скінченна підмножина N;

Номінальною ознакою є протокол, до якого належить надісланий пакету.

порядкові, що представляють собою номінальні ознаки, для яких визначено лінійний порядок;

В якості прикладу порядкової ознаки можна привести кількість флагів у заголовку пакету.

кількісні, значенням яких є дійсне число.

Кількісною ознакою є розмір надісланого пакету.

yi ⊂ Y позначає приналежність даного прикладу до атаки.

2.3. Відомості про датасет

Щоб створити алгоритм аналізування трафіку необхідно створити певну вибірку та структурувати її, в результаті отримують набір даних на яких вчитися ML. Для цього використовують датасети.

Датасет – це колекція даних, яка використовується для аналізу, обробки інформації та навчання штучного інтелекту. Датасети можуть містити різні типи даних, такі як текст, зображення, числа, аудіозаписи тощо. Датасети забезпечують необхідні дані для тренування моделей, дозволяючи алгоритмам вчитися на прикладах і вдосконалювати свою здатність прогнозувати або класифікувати.

Для задачі класифікації DDoS атак існує багато датасетів, ось деякі з них: KDD Cup 99, CICDDoS2019, ISCX IDS Dataset 2012, CAIDA UCSD "DDoS Attack 2007" Dataset.

Розглянемо їх ближче:

1) KDD Cup 99

Цей набір даних, використовувався для Третього міжнародного конкурсу інструментів виявлення знань і аналізу даних, який проводився разом із П'ятою міжнародною конференцією KDD-99 з відкриття знань і аналізу даних. Конкурсне завдання полягало в створенні детектора мережевого вторгнення,

прогнозної моделі, здатної розрізняти «погані» з'єднання, які називаються вторгненнями або атаками, та «добрі» нормальні з'єднання. Ця база даних містить стандартний набір даних, які підлягають перевірці, включаючи широкий спектр вторгнень, змодельованих у військовому мережевому середовищі. [18]

2) CICDDoS2019

CICDDoS2019 містить безпечні та найсучасніші поширені DDoS-атаки. Він також містить результати аналізу мережевого трафіку за допомогою CICFlowMeter-V3 із позначеними потоками на основі міток часу, IP-адрес джерела, портів джерела, протоколів і атак (поширюється в файлах формату CSV).

Створення реалістичного фонового трафіку було головним пріоритетом у зібранні цього набору даних. Для цього датасету була створена абстрактна поведінку 25 користувачів на основі протоколів HTTP, HTTPS, FTP, SSH та електронної пошти.

У цьому наборі даних є різні сучасні рефлексивні DDoS-атаки, такі як PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS і SNMP.[19]

3) ISCX IDS Dataset 2012

Даний датасет був створений Інститутом Інформаційних Систем Безпеки Канади (ISCX) в Університеті Нью-Брансвіку. Він був розроблений для дослідження та оцінки систем виявлення вторгнень.

Однією з основних переваг ISCX IDS Dataset 2012 є те, що він містить дані, які намагаються імітувати реальні мережеві умови. Це включає нормальний мережевий трафік, а також широкий спектр атак, таких як DDoS, атаки на веб-сервери, атаки на авторизацію тощо. Різноманітність Атак: Датасет включає різні типи атак, що дозволяє оцінювати ефективність IDS в різних сценаріях. Він включає зразки DDoS атак, атак з використанням ін'єкції SQL, атак "man-in-the-middle", і багато інших.

4) CAIDA UCSD "DDoS Attack 2007

Набір даних, який був зібраний під час реальної DDoS атаки, яка відбулася в серпні 2007 року. Цей датасет був створений та опублікований Центром Прикладних Інтернет-Досліджень Даних, який знаходиться в Каліфорнійському університеті

Для навчання класифікації я буду використовувати дані датасету CICDDoS2019, оскільки це останній оновлений відкритий датасет, оскільки який містить багато різноманітних типів DDoS атак, має великий розмір його можна знайти у відкритому доступі.[25]

2.3.1. Опис ознак датасету CICDDOS2019

Усього даний датасет має 84 ознаки, остання з них відповідає за належність до атаки або відсутності. Датасет має велику кількість характеристик яку потрібно скоротити, оскільки багато з них не мають великого значення для опису трафіку. В статті Machine Learning DDoS Detection Using Stochastic Gradient Boosting, в якій використовувалася модель машинного навчання на основі стохастичного градієнтного підсилення. Великим плюсом даного алгоритму є те що можна подивитися, які ознаки краще вплинули на навчання. В усякому разі в статті сказано що при навчанні даної моделі на неї сильніше вплинули характеристики зазначені на рисунку 2.1:

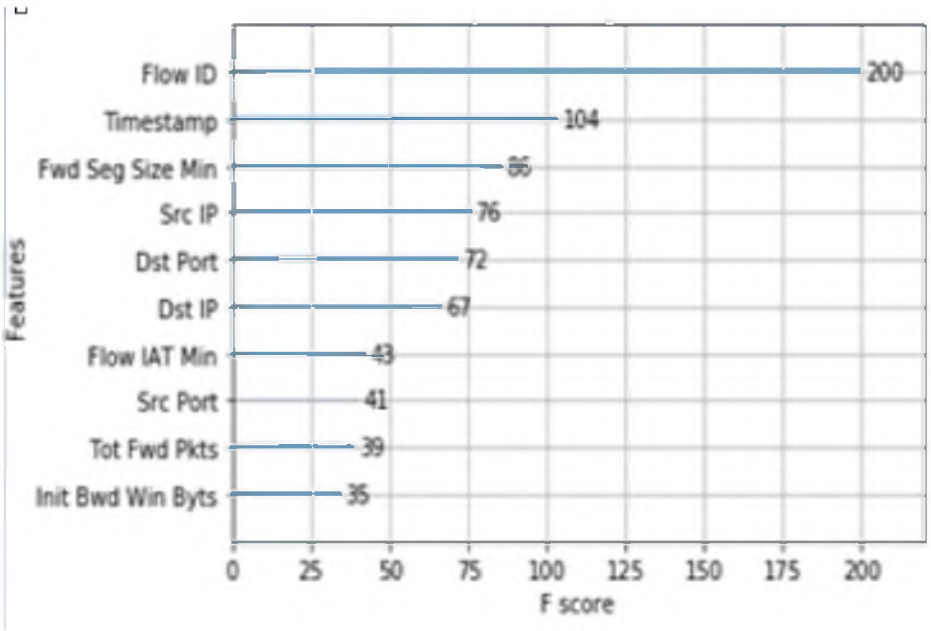

Рисунок 2.1 – Оцінка важливості ознак [20]

На основі цих даних обмежимся такими ознаками для економії часу навчання, вони зазначені в таблиці 2.1:

Таблиця 2.1 – Ознаки та їх опис для машинного навчання

Назва ознаки	Опис
Pkt Size Avg	Середній розмір пакету
Fwd seg size min	Мінімальний розмір сегменту
Src IP	IP відправника
Dst port	Порт отримувача
Dst IP	IP отримувача
Flow IAT Min	Мінімальний час між двома потоками
Src Port	Порт відправника
Tot Fwd Pkts	Загальна кількість вхідних пакетів
Init Bwd Win Byts	ініціалізаційного розміру вікна відповіді в байтах
Label	DDoS атака або звичайний трафік

Також для покращення швидкодії та покращення результату машинного навчання були виконанні дії над стовпцями даних в датасеті. Після цих дій розмір датасету повинен зменшитися, та підвищиться чистота даних.

Це були такі дії: Стовпці з одним значенням були видалені;

- Стовпці, у яких відсутні значення не перевищують 40 відсотків, вилучено;
- Рядки, у яких відсутні значення стовпця становили не більше 5 відсотків;
- Оброблено помилкові дані шляхом їх заміни, жоден екземпляр не був числом

Після виконання цих дій було помічено, що використання оперативної пам’яті програмою зменшилося на 500 мегабайт.

2.3.2. Як збираються ознаки в датасетах

Усі ознаки беруться з реального трафіку, який можна отримати з сервера у форматі access.Log, або за допомогою спеціалізованого програмного забезпечення такого як Wireshark. Для базової інформації використовують access.log, а для більш детального огляду пакетів застосовують Wireshark. На рисунку 2.4 та 2.5 зображені приклади з представленням пакетів в accesslog та Wireshark-у відповідно:

Рисунок 2.4 – представлення пакетів у acces.log

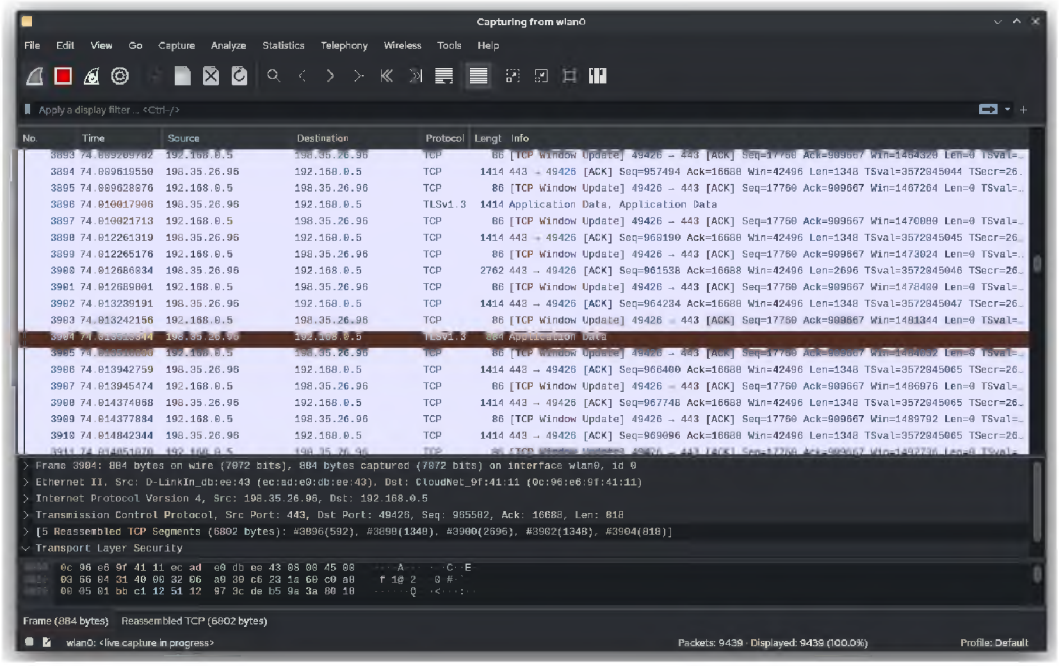

Рисунок 2.5 – Представлення пакетів у Wireshark

Далі для побудови датасету визначаються ознаки, а саме за якими буде відбуватися класифікація, до якого класу належить зразок. Цей набір ознак, буде використовуватися для ML, щоб розпізнавати запити на наявність притаманності к DDoS-атакам. При складанні датасету очікується, що виділені ознаки містять доречну інформацію. Записуватися ці ознаки в датасеті можуть в різному форматі:

- Булевому – приймати значення 1 або 0.

- Номінальному – приймаються дійсні числові значення.
- Порядкові - що представляють собою номінальні ознаки, для яких визначено лінійний порядок;
- Кількісні - значенням яких є дійсне число

Після такої класифікації маємо на руках список ознак (маркерів) які будуть притаманні кожному запиту відправленому задля проведення DDoS-атаки. Приклад такої підбірки зображений на рисунку 2.6:

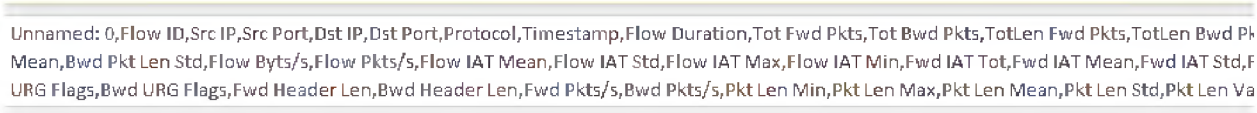

Рисунок 2.6 – ознаки датасету CICDDoS2019

2.4. Інструменти для алгоритма класифікації

Google Colaboratory – це інтерактивне середовище, з відкритим доступом яке ще також називається блокнотом, в якій можна: писати та компілювати програмний код мови Python, форматувати текст, зображення, а також підвантажувати відкриті приклади з github. Доступ до цього ресурсу можна отримати через браузер. Весь програмний код, який буде написаний в даному середовищі, буде використовувати хмарні ресурси Google, це означає, що можна використовувати апаратне забезпечення Google, у тому числі графічні процесори та TPU, незалежно від потужності машини. Основною властивістю Colab це те, що весь програмний код може бути представлений у вигляді блоків змінні в яких можуть бути змінені незалежно один від одного.

Python є універсальною мовою програмування високого рівня, призначеною для різноманітних цілей. Вона підтримує ряд функцій, як включають в себе: структуроване, об'єктно-орієнтоване програмування. Запроваджений ще наприкінці 80-х, Python стабільно розроблявся та став входити у число найпопулярніших мов програмування.[21]

Популярність Python забезпечила велику кількість навчальних матеріалів, які допомагають у вирішенні різних задач. Крім того, Python має сильну активну спільноту, що підтримує його великою кількістю бібліотек та фреймворків, особливо в областях машинного навчання та штучного інтелекту.

Наприклад, NumPy та SciPy використовуються для математичних та інженерних розрахунків, Pandas - для обробки даних, а Scikit-Learn для машинного навчання та штучного інтелекту.

Python вирізняється стабільністю, гнучкістю та наявністю великої кількості інструментів, що дозволяє розробникам писати надійне програмне забезпечення, навіть коли вони працюють із складними алгоритмами та універсальними робочими процесами. Лаконічний і читабельний код, а також інтеграція з іншими інженерними середовищами, як R через бібліотеки типу RPy2, роблять Python ідеальним вибором для проектів, пов'язаних з AI та ML. Використання Python значно скорочує час розробки, оскільки розробники можуть зосередитися на вирішенні проблем машинного навчання, не вдаючись в технічні подробиці.

2.4.1. Бібліотеки Python, які були використані

- Pandas - бібліотка для обробки та аналізу даних, шляхом створення таблиць таких як DataFrame та Seriies.
- Numpy - бібліотека для обчислень, яка використовується багатовимірних масивів та матриць, разом із великим набором математичних функцій для роботи з цими масивами.
- Picle - модуль для серіалізації та десеріалізації об'єктів. використовується для збереження складних об'єктів у файлі та відновлення їх з файлу пізніше. Це дозволяє легко зберігати конфігурації, стани моделей або будь-які інші об'єкти Python для подальшого використання.
- Sklearn - це бібліотека для машинного навчання. Дана бібліотека надає прості та ефективні інструменти для аналізу даних та машинного навчання. Включає алгоритми класифікації, регресії, кластеризації та зниження розмірності, а також засоби для вибору моделі, обробки даних та оцінки моделей.

2.5. Методи машинного навчання які використовуються для класифікації

2.5.1. DecisionTree

Дерева рішень є досить універсальними та гнучкими моделями навчання з учителем, які мають важливу властивість – простоту інтерпретації. Дерево рішень – це структура даних у вигляді бінарного дерева, що використовується для прийняття рішень. Ця структура надає можливість прогнозування як категоріальних значень (дерева класифікації), і значень у форматі дійсних чисел (дерева регресії), так навіть здатні містити і числові, і категоріальні дані без будь-яких операцій нормалізації чи створення фіктивних змінних.

Розглянемо детальніше процедуру створення звичайного навчального дерева рішень (структури, що формується зверху донизу):

1. Починаючи з кореня (root) дерева увесь набір даних поділяється на дві під-множини нащадки на основі бінарної умови. Наприклад, якщо визначено умову «f1 $\geq$ 0.45», то всі елементи даних, для яких ця умова істинна, направляються в лівий набір-нащадок, а елементи, для яких умова хибна, – у правий набір-нащадок.

2. Далі підмножини-нащадки продовжують рекурсивно розділятися на більш дрібні підмножини на основі інших умов. Умови поділу автоматично вибираються на кожному кроці в залежності від того, яка умова найкраще поділяє поточний набір елементів. Існує кілька загальноприйнятих метрик, якими кількісно оцінюється якість поділу:

- міра неоднорідності Джині (Gini impurity): якщо елементи в підмножині були довільним чином позначені відповідно до розподілу міток у наборі, то відносна частка неправильно позначених елементів має бути визначена як міра неоднорідності Джині. Наприклад, якщо у підмножині 25 % елементів мають мітку 0 (відповідно 75 % елементів з міткою 1), то присвоювання мітки 0 випадково вибраним 25 % від усіх елементів (іншим присвоюється мітка 1) має давати 37,5 % неправильних міток: 75 % елементів з міткою 0 та 25 % елементів з міткою 1 повинні бути некоректними. Операція поділу дерева рішень вищої якості повинна розділяти набір даних на підмножини, що чітко визначаються за їх мітками, отже, в результаті неоднорідності Джині виходить більш низькою. Таким чином, коефіцієнт

неправильної класифікації повинен бути низьким, якщо більшість елементів у наборі належить до того самого класу;

- зменшення дисперсії (variance reduction) часто використовується в деревах регресії з залежною змінною, що постійно присутня (поширюється по дереву). Зменшення дисперсії визначається як сумарне зменшення дисперсії в наборі, розділеному на два підмножини-нащадка. Найкращим варіантом поділу у вузлі дерева рішень має вважатися поділ, результатом якого є найбільше зниження дисперсії;
- приріст інформації (information gain) – це міра однорідності (purity) підмножин, отриманих у результаті поділу. Приріст інформації обчислюється за допомогою віднімання виваженої суми ентропії кожного вузла-нащадка дерева рішень з ентропії батьківського вузла. Що менше ентропія нащадків, то більше вписувалося приріст інформації, отже, якісніше виконано поділ.

3. Існує кілька різних методів для визначення моменту зупинення процедури поділу вузлів:

- коли чергова гілка дерева досягає деякої попередньо визначеної максимальної глибини (maximum depth), подальший поділ гілок припиняється;
- коли все листя дерева є однорідними (pure), тобто кожен лист-вузол містить тільки елементи, що належать одному і тому ж класу, отже, поділ припиняється;
- коли один (будь-який) із вузлів нащадків містить кількість елементів, меншу, ніж мінімальна кількість елементів вибірки (minimum number of samples), такий вузол більше не підрозділяється.

При завершенні процесу алгоритм виводить структуру дерева, в якій кожен вузол представляє бінарне рішення, нащадки кожного вузла представляють два можливі вихідні результати цього рішення, а кожен лист представляє класифікацію елементів даних відповідно від кореня дерева до цього листа. (Для неоднорідного (impure) листя рішення визначається більшістю голосів у тренувальній вибірці даних у відповідному аркуші.) На малюнку 2.1 показано, умовна структура дерева:

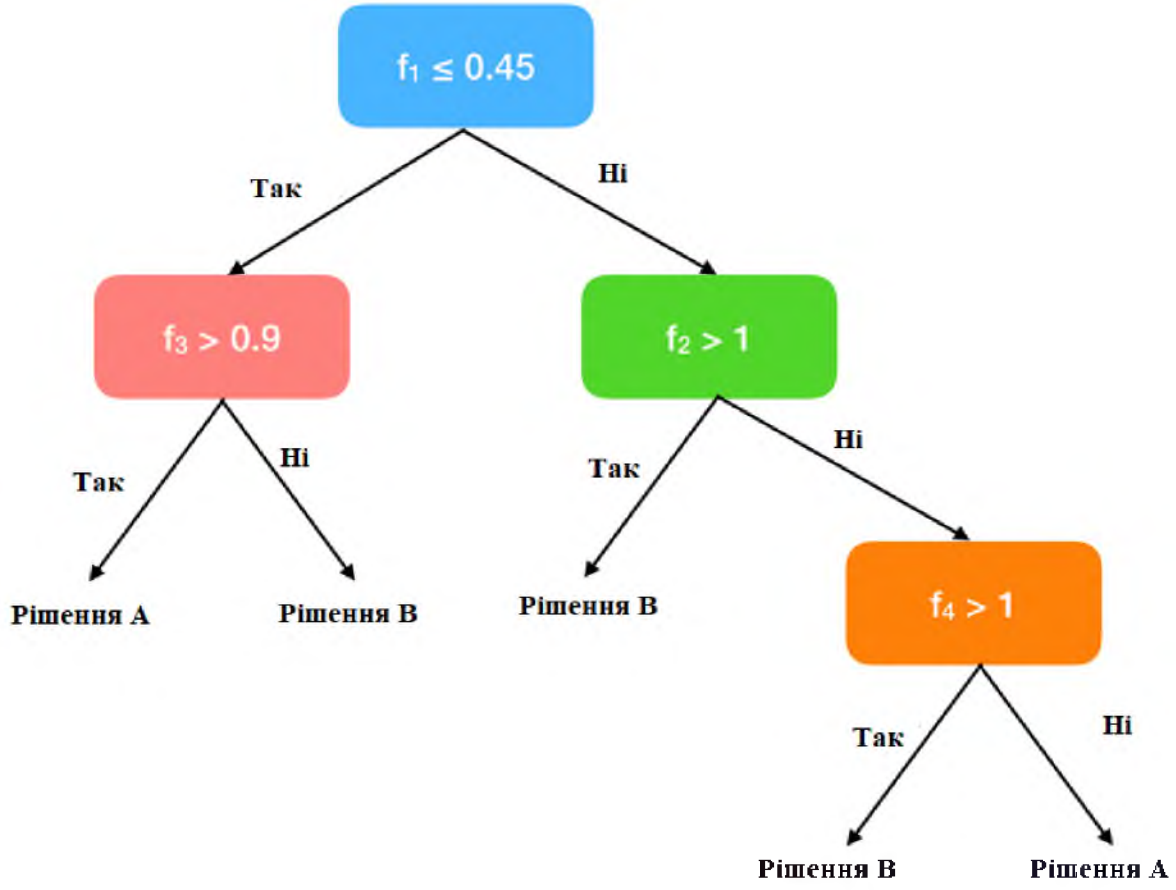

Рисунок 2.1 – Дерево рішень

Є декілька недоліків при використанні дерев'їв рішень:

- у деревах рішень часто виникає проблема перепідгонки (overfitting), коли дерева стають надмірно складними і не здатні правильно узагальнити тренувальний набір даних. Для зменшення складності дерев як регулюючої методики застосовується відсікання (pruning) (малоерспективних гілок);

- дерева рішень іноді менш точні та надійні (стійкі), ніж інші методики навчання з учителем. Незначні зміни в тренувальному наборі даних можуть призводити до істотних змін у дереві, що, в свою чергу, тягне за собою зміни в моделі прогнозів.

- метрики якості поділу для категоріальних змінних у деревах рішень мають тенденцію зміщуватися у бік змінних з більш ймовірними значеннями, тобто поділ за безперервними змінними або категоріальними змінними з трьома і більше категоріями призведе до вибору з більш високою ймовірністю, порівняно з бінарними змінними.

2.5.2. GaussainNB

Наївний байєсовський класифікатор (Naive Bayes classifier) є одним із найстаріших статистичних класифікаторів. Цей класифікатор називається наївним», тому що ґрунтується на вельми строгих статистичних вихідних передумовах, а саме: ознаки вибираються незалежно з деякого (невідомого заздалегідь) розподілу. Наївний класифікатор Байєса використовує наведене далі рівняння (теорема Байеса) для обчислення ймовірності настання конкретної події з урахуванням певного набору значень ознак:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad (1)$$

Де А та В є подіями

- P(A) та P (B) є ймовірностями А та В безвідносно одна до одної

- P(A|B), умовна ймовірність, є ймовірністю події А за умови істинності В.

- P(B|A) є ймовірністю В за умови істинності А [17]

Цей метод є одним з найпростіших та найбільш ефективних підходів до класифікації, що дозволяє розробляти швидкі моделі машинного навчання, здатні проводити оперативні прогнози.

Наївний байєсівський класифікатор має ряд переваг: він є одним з найшвидших і найдоступніших методів машинного навчання для визначення класу даних. Його можна застосовувати як для бінарної, так і для багатокласової

класифікації, при цьому він показує високу ефективність у багатокласових задачах порівняно з іншими алгоритмами.

Проте, серед недоліків наївного байєсівського класифікатора - припущення про незалежність ознак, тобто він не може враховувати кореляції між ними, що може впливати на точність моделі в деяких випадках.

2.5.3. Random forest

Випадкові ліси (random forests) формуються як прості ансамблі з кількох дерев рішень, які зазвичай містять від десятків до тисяч таких дерев. Ансамбль - це об'єднання кількох класифікаторів, у якому створюється складніший і найчастіше ефективніший класифікатор. Такі ансамблі часто називають лісами дерев рішень (decision forests)

Після тренування кожного окремого дерева рішень узагальнені прогнози випадкових лісів виконуються з урахуванням статистичної моди (типовості) окремих прогнозів від класифікаційних дерев (та статистичного середнього значення (меани) прогнозів окремих дерев для регресійних дерев.

Слід зазначити, що просте наявність безлічі дерев рішень у лісі призведе до того, що дерева будуть дуже схожі один на одного і виникатиме величезна кількість повторюваних поділів у різних деревах, особливо для тих ознак, які є найсуворішими критеріями прогнозу для залежної змінної.

Алгоритм формування випадкового лісу вирішує проблему, використовуючи наступний алгоритм тренування:

1. Для тренування кожного окремого дерева випадковим чином вибирається підмножина N елементів (вибірка) із загального тренувального набору даних.

2. У кожній точці поділу випадково вибирається m ознак з p доступних ознак, при цьому $m \leq p$. Далі вибирається оптимальна точка поділу з цих m ознак

3. Крок 2 повторюється доти, доки не завершиться тренування окремого дерева

4. Кроки 1, 2 та 3 повторюються доти, доки повністю не завершиться тренування всіх дерев у лісі.

В одиночних деревах рішень часто виникає проблема перепідгонки, при роботі з тренувальними наборами даних, а випадкові ліси пом'якшують цей небажаний ефект, середня характеристика безлічі дерев рішень, що дозволяє покращити ефективність моделі. Крім того, оскільки кожне дерево у випадковому лісі може тренуватися незалежно від усіх інших дерев, стає очевидною можливість розпаралелювання тренувального алгоритму. Таким чином, випадкові ліси здатні виконувати процес тренування дуже ефективно. Але складність випадкових лісів, що зростає зі збільшенням розміру, може істотно збільшити обсяг ресурсів, необхідних для зберігання даних, і значно утруднити пояснення прогнозів, порівняно з одиночними деревами рішень.

2.5.4. MLPClassifier

Штучні нейронні мережі представляють собою математичні моделі та їх програмні втілення, які створені на основі принципів структури та функціонування біологічних нейронів, а саме нервові клітини в організмах живих істот. Різноманітні архітектури цих мереж знаходять застосування у багатьох областях, включаючи системи виявлення вторгнень, де однією з найбільш базових є архітектура MLP (багатошарового перцептрона).

MLP складається з послідовності перцептронів - елементарних обчислювальних одиниць, що імітують діяльність біологічних нейронів, та включає мінімум три шари: вхідний, один або кілька прихованих шарів, та вихідний.

Для тренування MLP типово використовується метод зворотного поширення помилок, який дозволяє адаптувати мережу на основі розбіжностей між фактичними та прогнозованими результатами. Помилки "повертаються" назад до мережі для коригування ваг, що забезпечує оптимізацію вихідних даних. З достатньою кількістю нейронів і налаштованих прихованих шарів, MLP може ефективно апроксимувати функціональні залежності між вхідними та вихідними даними.

Тим не менш, існують певні обмеження та виклики у використанні нейронних мереж. Зокрема, тренування може потрапити в тупикові ситуації або вимагати значного часу, що обмежує їхнє використання в додатках реального часу. Крім того, поведінка навченої мережі може бути непередбачуваною, що створює ризики при її використанні для управління складними технічними системами. Оскільки нейронні мережі імітують складність біологічних систем, вони можуть демонструвати високу ефективність в розв'язанні складних задач, але це також робить їх поведінку менш інтуїтивно зрозумілою та контрольованою. На рисунку 2.2 зображено структура нейронної мережі.

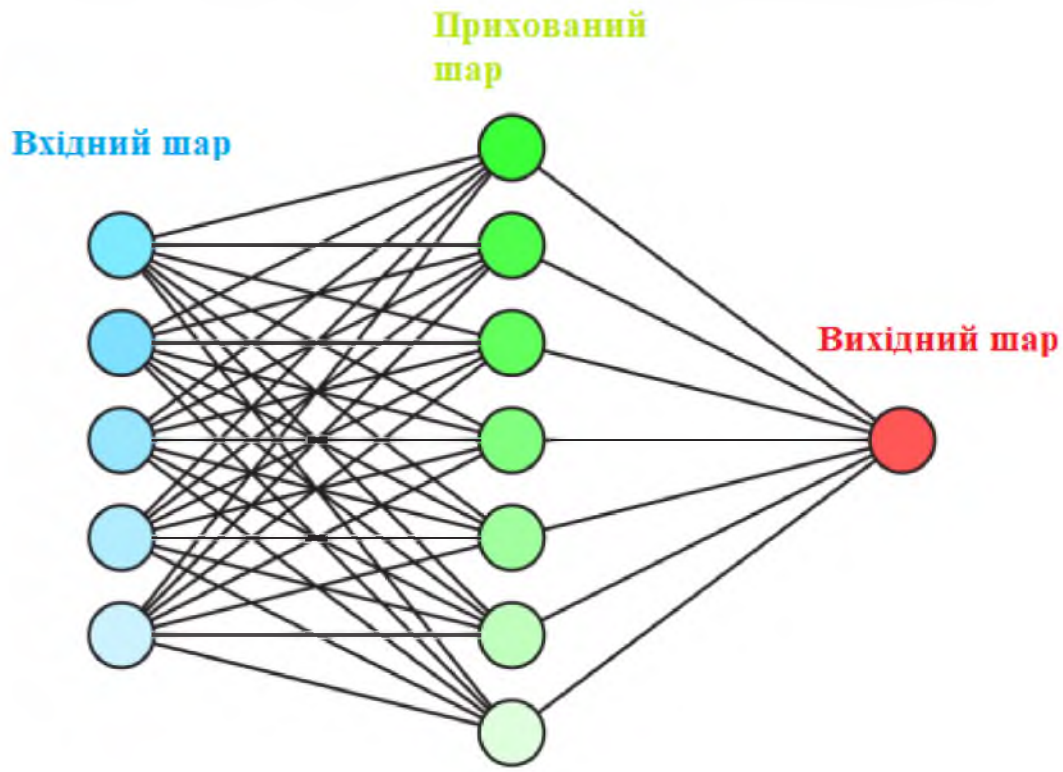

Рисунок 2.2 – Структура нейронної мережі

2.6. Як оцінюються алгоритми

ROC аналіз (Аналіз робочих характеристик приймача) - техніка в області машинного навчання, яка використовується для оцінки ефективності класифікаційних моделей. ROC аналіз дозволяє визначити, наскільки добре

модель відрізняє справжні позитивні від ложно-позитивних результатів. Цей аналіз виглядає у вигляді кривої яка відображає залежність кількості вірно класифікованих позитивних випадків від кількості невірно класифікованих негативних. У випадку виявлення DDOS атак позитивною вважається подія атаки

Модель розбиває, порогом прийняття рішень вхідні дані на позитивні або негативні. Ці позитивні створюють структуру помилок I та II родів.

Структуру помилок визначають з таблиці спряженості (confusion matrix) яка будується на основі результатів класифікації і фактичної (об'єктивної) приналежності прикладів до класів.

Таблиця 2.2 Confusion matrix

Модель	True positive	True Negative
Predicted positive	TP	FP
Predicted negative	FN	TN

- TP (True positive) - Кількість правильно визначених позитивних прикладів.
- TN (True Negative) - Кількість правильно визначених негативних прикладів.
- FN (False Negative) - Кількість неправильно визначених негативних прикладів.
- FP (False Positive) - Кількість неправильно визначених позитивних прикладів.

При аналізі результатів оперують відносними показниками вираженими у відсотках а саме:

- (Accuracy) – відсоток правильно класифікованих даних моделью

$$A = \frac{TPR + TNR}{TPR + TNR + FPR + FNR} \quad (2.2)$$

- Precision – відсоток істинно позитивних результатів від загальної кількості істинно позитивних випадків даних. Тобто це міра того,

наскільки багато з тих, кого модель класифікувала як позитивні, справді є позитивними.

$$P = \frac{TP}{TP + FP}$$

- Recal (Повнота) – відсоток позитивних результатів від загального числа істинно позитивних даних. Тобто це міра того, скільки зі всіх позитивних випадків модель змогла виявити. (2.1)

$$R = \frac{TP}{TP + FN} \quad (2.4)$$

- F1-score - це показник точності моделі на наборі даних. Це гармонічне середнє значення точності та запам’ятовування

$$F1 = \frac{2PR}{P + R} \quad (2.5)$$

- Чутливість (TPR) - Це міра здатності класифікатора правильно ідентифікувати позитивні випадки.[23]

$$TPR = \frac{TP}{TP + FN} \quad (2.6)$$

- Специфічність (TNR) - Це міра здатності класифікатора правильно ідентифікувати негативні випадки.[23]

$$TNR = \frac{TN}{TN + FP} \quad (2.7)$$

- FPR - неправильно ідентифікованих негативних випадків.

$$FPR = \frac{FP}{FP + TN} \quad (2.8)$$

2.7. Програмна реалізація.

Спочатку визначаємо, яку максимальну кількість стовпців та рядків будуть відображатися при виводі DataFrame. Значення None означає, що обмеження на кількість стовпців не існує.

```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

Далі в dtypes визначається усі ознаки які є в датасеті та його тип даних, далі ці дані будуть використовуватися для машинного навчання.

```
dtypes = {
'Src IP': 'category',
'Src Port': 'uint16',
'Dst IP': 'category',
'Dst Port': 'uint16',
'Protocol': 'category',
'Flow Duration': 'uint32',
'Tot Fwd Pkts': 'uint32',
....
'Label': 'category'
}
```

Після цього зчитуються дані з шляхом до вказаного датасету, де dtype є параметром, який вказує на тип даних для стовпців у DataFrame, а dtypes словник, де ознаки - це назви стовпців, а значення - типи даних цих стовпців. Вказую на C парсер, що ускорює обробку даних, та використовую параметр для оптимізації використання пам’яті

```
df = pd.read_csv('/content/drive/MyDrive/CIC-DDoS2019.csv',
dtype=dtypes,
engine='c',
low_memory=True
    )
```

Відкидються стовпці (з переліком) в яких переглядається тільки 1 значення

```
colsToDrop = np.array(['Fwd Byts/b Avg', 'Fwd Pkts/b Avg', 'Fwd Blk Rate Avg', 'Bwd Byts/b Avg', 'Bwd Pkts/b Avg', 'Bwd Blk Rate Avg'])
```

Обчислює кількість відсутніх (NaN) значень у кожному стовпц датасету. Визначаються стовпці які мають 40% або більше відсутніх значень та стовпці, у яких відсоток відсутніх значень більше 0%, але менше або дорівнює 5%

```
missing = df.isna().sum()
missing = pd.DataFrame({'count': missing, '% of total': missing/len(df)*100}, index=df.columns)
colsToDrop = np.union1d(colsToDrop, missing[missing['% of total'] >= 40].index.values)
dropnaCols = missing[(missing['% of total'] > 0) & (missing['% of total'] <= 5)].index.values
```

Замінюються всі значення нескінченності на (NAN), та додаються до масиву dropnaCols.

```
df['Flow Byts/s'].replace(np.inf, np.nan, inplace=True)
df['Flow Pkts/s'].replace(np.inf, np.nan, inplace=True)
dropnaCols = np.union1d(dropnaCols, ['Flow Byts/s', 'Flow Pkts/s'])
```

Видалення стовпців.

```
df.drop(columns=colsToDrop, inplace=True)
df.dropna(subset=dropnaCols, inplace=True)
print("Процес завершився успішно.\n")
```

Задається список назв стовпців, які будуть використовуватися як ознаки для моделі та задається назва стовпця, який використовується як цільова змінна.

```
features = ["Fwd Seg Size Avg", "Flow IAT Min", "Flow Duration", "Tot Fwd Pkts", "Pkt Size Avg", "Src Port", "Init Bwd Win Byts"]
target = "Label"

x = df.loc[:,features]
y = df.loc[:,target]
```

Вибір моделі

```
if (str(choice) == "0"):
```

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
modelname = "DecisionTree"
elif (str(choice) == "1"):
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
modelname = "GaussianNB"
elif (str(choice) == "2"):
from sklearn.ensemble import RandomForestClassifier
 model = RandomForestClassifier()
modelname = "RandomForest"
elif (str(choice) == "3"):
from sklearn.neural_network import MLPClassifier
model = MLPClassifier()
modelname = "MLP"
else:
print("Ви ввели від неможливу зміну. Вибрано модель DecisionTreeClassifier.")
 model = DecisionTreeClassifier()
```

Далі відбувається розділення даних на тренувальний та тестовий набір

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_size=1.0-float(80/100))
```

Відбувається тренування моделі по заданим вище ознакам та цільової змінної, після чого розраховується Accurasy, precision, recall, f1 score, та будується матриця помилок

```
model.fit(x_train, y_train)
print("Model fitted. \n")
   #Отримання результатів
y_pred = model.predict(x_test)
score = accuracy_score(y_test, y_pred)*100
```

```
print("Accuracy of the model "+modelname+" is: ", score)
precision = precision_score(y_test, y_pred, pos_label='Benign')*100
print("Precision of the model "+modelname+" is: ", precision)
recall = recall_score(y_test, y_pred, pos_label='Benign')*100
print("Recall of the model "+modelname+ " is: ", recall)
f1 = f1_score(y_test, y_pred, pos_label='Benign')*100
print("F1 score of the model "+modelname+ " is: ", f1)
matrix = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:")
print( matrix)
```

Візуалізується ROC – крива та матриця помилок

```
# Побудова ROC-кривої
def plot_roc_curve(model, x_test, y_test):
    y_pred_proba = model.predict_proba(x_test)[:,1]
    fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba, pos_label='ddos')
    roc_auc = auc(fpr, tpr)

    plt.figure()
    plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)'
% roc_auc)
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic')
    plt.legend(loc="lower right")
    plt.show()

# Виклик функції для побудови ROC-кривої
```

```
plot_roc_curve(model, x_test, y_test)

# Візуалізація Confusion Matrix
def plot_confusion_matrix(matrix):
    plt.figure(figsize=(10, 7))
    sns.heatmap(matrix, annot=True, fmt='g', cmap='Blues')
    plt.xlabel('Predicted label')
    plt.ylabel('True label')
    plt.title('Confusion Matrix')
    plt.show()

# Виклик функції для візуалізації Confusion Matrix
plot_confusion_matrix(matrix)
```

2.7.1. Розгляд та результат кожного з алгоритмів

Якщо не задавати ніяких параметрів при побудові моделі при використанні DecisionTreeClassifier, то за замовчуванням він буде мати максимальну можливу глибину та максимальну кількість листків, що приводить до складності в розумінні та проблеми перепідгоники, таке дерево зображене на рис. 2.3:

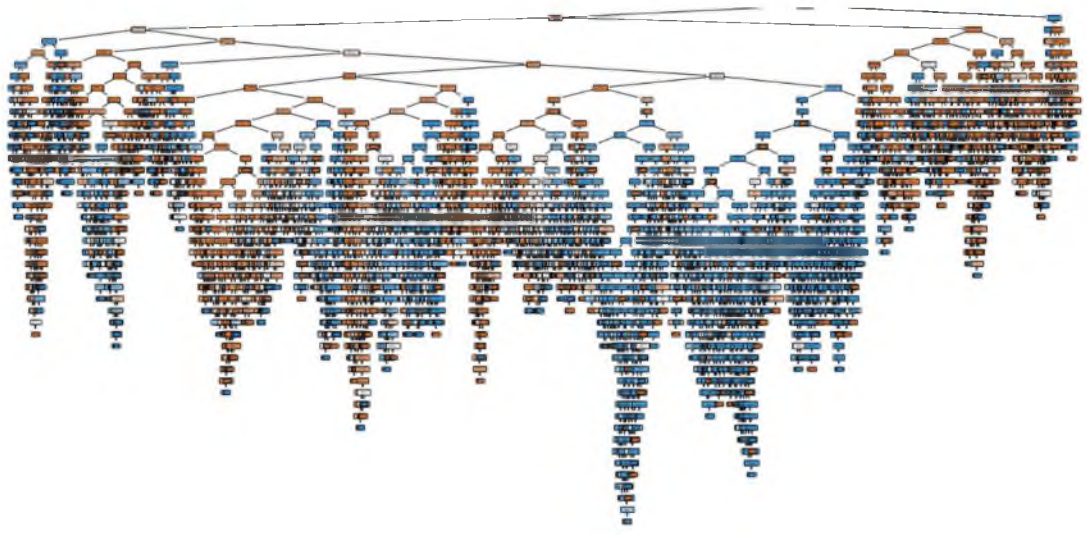

Рисунок 2.3 – Дерево з максимальною глибиною

Для запобігання цього в аргументах DecisionTree потрібно вказувати параметр max_depth = , який вказує наскільки буде ділитися дерево.

Критерій за яким будуть розділятися гілки в дереві – Entropy, оскільки за цим критерієм будується більш вірне формування матриці помилок

Таблиця 2.3 Різні критерії ділення та матриці помилок

Функція розбиття	Gini	Entropy	Log_loss
Матриця помилок	[[992476; 11025] [14609; 244721]]	[[1002699; 802] [4884; 25446]]	[[1002699; 802] [4884; 25446]]

За задаванням інших критерій таких як max_leaf_nodes, min_samples_split, min_samples_leaf, покращити результат не вдалося тому було вирішено лишити за їх замовчуванням.

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(max_depth=5,criterion='entropy')
modelname = "DecisionTree"
```

Після було побудовано Roc – криву цього алгоритма за його confusion matrix. Відповідно рис. 2.4. та таблиця 2.4:

Таблиця 2.4 Ітогова матриці помилок алгоритма DecisionTreeClassifier

	Справжні позитивні	Справжні негативні
Прогнозовано позитивні	1002699	802
Прогнозовано негативні	4884	25446

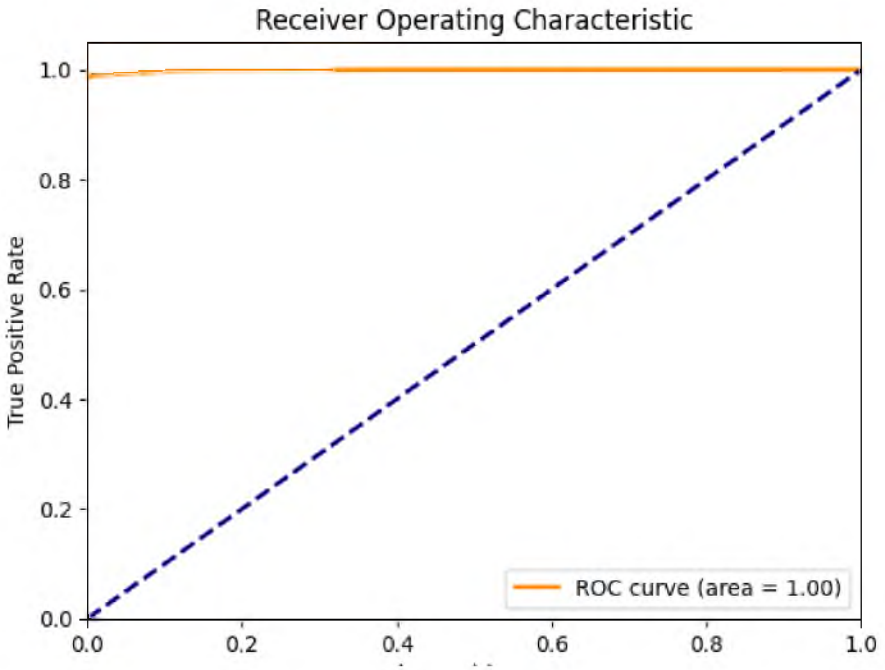

Рисунок 2.4 – ROC крива DecisionTreeClassifier

Оцінки алгориму DecisionTreeClassifier на основі матриці помилок:

Accuracy of the model DecisionTree is: 99.54974181026599

Precision of the model DecisionTree is: 99.51527566463507

Recall of the model DecisionTree is: 99.92007980061803

F1 score of the model DecisionTree is: 99.71726690680251

Час навчання алгориитму: 58 секунд

Задамо значення для уникнення проблем з обчисленнями, коли дисперсія дуже мала або рівна нулю.

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB(var_smoothing=1e-9)
modelname = "GaussianNB"
```

Після його роботи було побудовано Roc – криву цього алгоритма за його confusion matrix. Відпровідно рис. 2.5. та таблиця 2.5:

Таблиця 2.5 Матриці помилок алгоритма GaussianNB

	Справжні позитивні	Справжні негативні
Прогнозовано позитивні	622771	380730
Прогнозовано негативні	14828	244502

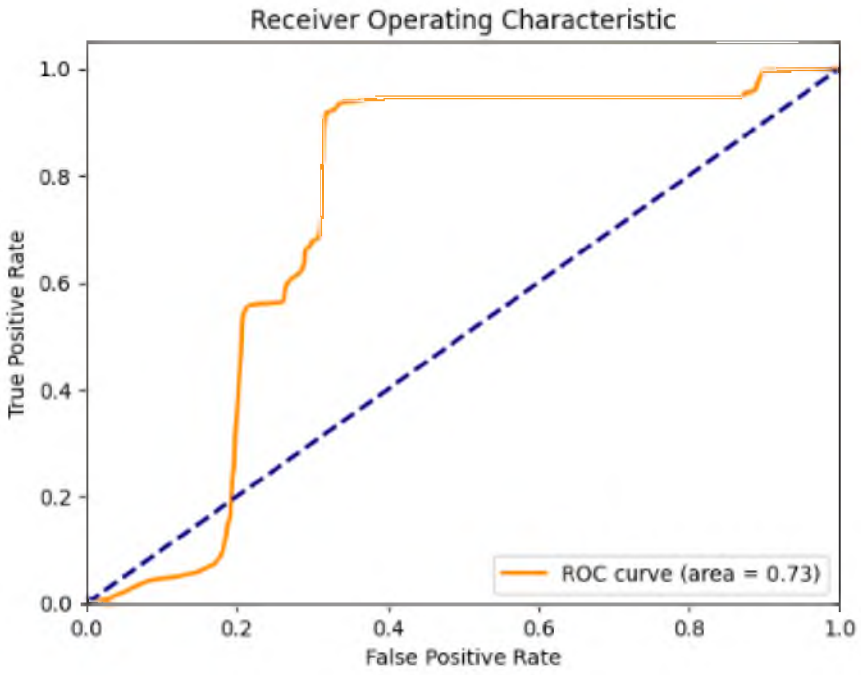

Рисунок 2.5 – ROC крива GaussianNB

Оцінки алгориму GaussianNB на основі матриці помилок:

Accuracy of the model GaussianNB is: 68.67688550566149

Precision of the model GaussianNB is: 97.67440036762919

Recall of the model GaussianNB is: 62.059828540280485

F1 score of the model GaussianNB is: 75.89677655231247

Час навчання алгориитму: 47 секунд

Головною відмінністю алгоритма RandomForest від DecisionTreeClassifier є те що будується декілька дерев а не одне

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=10,max_depth=5,criterion='entropy')
modelname = "RandomForest"
```

Після його роботи було побудовано Roc – криву цього алгоритма за його confusion matrix. Відпровідно рис. 2.6. та таблиця 2.6:

Таблиця 2.6 Матриці помилок алгоритма RandomForest

	Справжні позитивні	Справжні негативні
Прогнозовано позитивні	999125	4376
Прогнозовано негативні	29133	230197

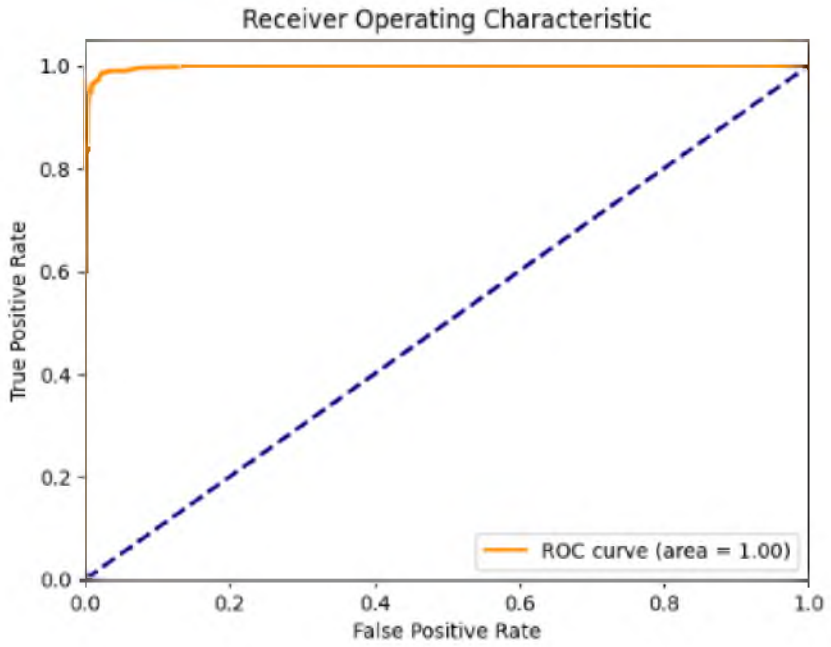

Рисунок 2.6 – ROC крива алгоритма RandomForest

Оцінки алгориму RandomForest на основі матриці помилок:

Accuracy of the model RandomForest is: 97.34651746749961

Precision of the model RandomForest is: 97.1667616493137

Recall of the model RandomForest is: 99.56392669264903

F1 score of the model RandomForest is: 98.35073943317096

Час навчання алгориитму: 3 хвилини

```
from sklearn.neural_network import MLPClassifier
        model = MLPClassifier(max_iter=50)
modelname = "MLP"
```

Після його роботи було побудовано Roc – криву цього алгоритма за його confusion matrix. Відпровідно рис. 2.6. та таблиця 2.6:

Таблиця 2.6 Матриці помилок алгоритма MLPClassifier

	Справжні позитивні	Справжні негативні
Прогнозовано позитивні	991542	11959
Прогнозовано негативні	2883	256447

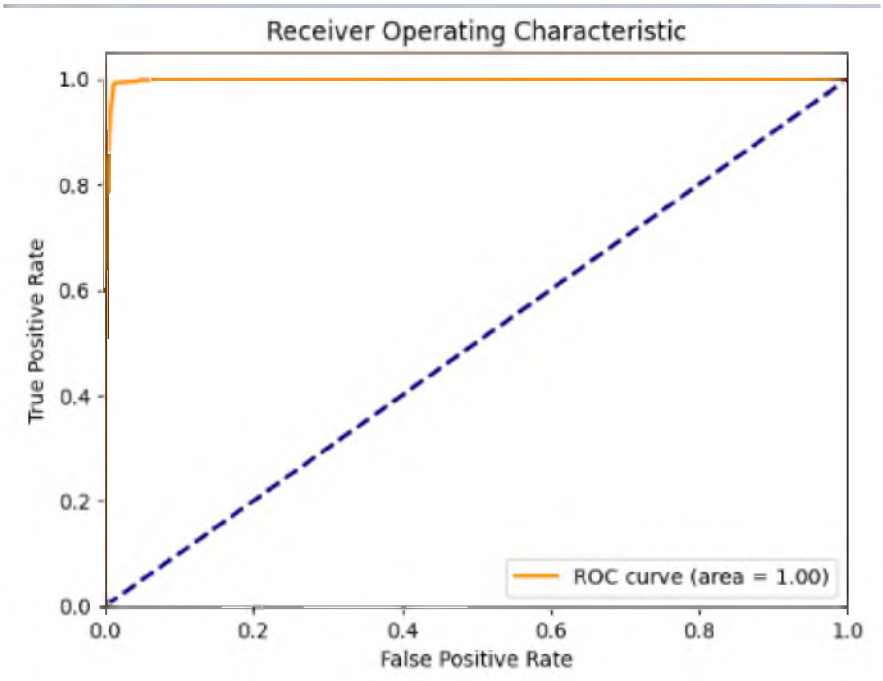

Рисунок 2.6 – ROC крива алгоритма MLPClassifier

Оцінки алгориму MLPClassifier на основі матриці помилок:

Accuracy of the model MLP is: 98.82470417656836

Precision of the model MLP is: 99.71008371672072

Recall of the model MLP is: 98.80827223889163

F1 score of the model MLP is: 99.25712964344025

Час навчання алгориитму: 37 хвилин

2.7.2. Оцінка та вибір кращого алгоритму який підходить для задачі класифікації

Для точної оцінки алгоритмів побудуємо таблицю порівняння кожного параметру кожного алгоритма

Таблиця 2.7 порівняння оцінок кожного з алгоритмів

	DecisionTree	GausainNB	RandomForest	MLPClassifier
Accuracy	99.59	68.67	97.35	98.82
Precision	99.51	97.67	97.16	99.71
Recall	99.92	62.05	99.56	98.80
F1 score	99.71	75.89	98.35	99.25
Час навчання	58 секунд	47 секунд	2 хвилини	37 хвилин

Отже, з таблиці можна побачити, алгоритми, які мали найменьшу та найбільшу ефективність у порівнянні з іншими, самим малоефективним алгоритмом показав себе GaussianNB. Байєвий алгоритм має такі оцінки: 68,67 за accuaracy, 97,67 за precision, 62,02 за recall і 75,89 за F1. Такий результат не виходить за межі очікуваного через те, що ця модель простіша за інші, швидша та потребує менше ресурсів обробки. Наївний баєвський класифікатор також не

ідеальний для роботи з великими обсягами даних. Щоб удосконалити результат цього класифікатора, можна, наприклад, виконати нормалізацію даних.

З іншого боку, кращим був алгоритм Decision Tree (зелений), який трохи кращий за Random Forest. Класифікатор Decision Tree має такі оцінки: 99.59 за accuaracy і, 99.51 за precision, 99.92 за Recall та 99,71 за F1. Оцінки були високі, що свідчить про те, що запропонований алгоритм може бути успішно реалізований для виявлялення DDoS-атаки з високою якістю.

Що стосується випадкового лісу, то воні показав також чудові результати. Це пов'язано з тим фактом, що цей класифікатор працює подібно до Decision Tree, що призводить до подібних результатів.

Щодо класифікатора MLP, який використовує нейронні мережі, ця модель потребує значно більше часу для навчання хоча й отримала добрі результати.

2.8. Використання алгоритма для навчання моделі виявлення DDoS атак

Далі з обраним алгоритмом можна створити модель, яка зможе ідентифікувати вхідний трафік на звичайний та ворожий. Модель може опрацьовувати мережеві пакети у реальному часі використовуючи набір ознак зазначених при побудові алгоритму, або на розгляд захисника системи, це

можуть бути: обсяги трафіку, частота запитів, використані порти, і відповідні IP-адреси Схема роботи моделі зображена на рис. 2.7:

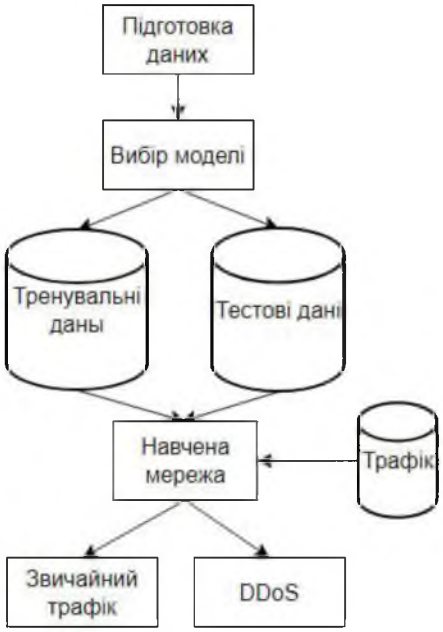

Рисунок 2.7 – Схема роботи створеної моделі

Після ідентифікації можна DDoS трафіку система може надавати додаткову про нападників: IP-адреси, порти та ін. . Ця інформація може бути використана для створення динамічних чорних і білих списків, які дозволяють миттєво блокувати ворожий трафік, не порушуючи обслуговування легітимних користувачів. Окрім того, інтелектуальна система може автоматизувати процес визначення маршрутів для відкидання трафіку (blackhole routing), які спрямовують шкідливий трафік у "чорну діру", звідки він не може завдати шкоди цільовим системам.

Інтеграція такої моделі з існуючими системами безпеки може значно підвищити ефективність заходів захисту від DDoS. Наприклад, відомості, отримані моделлю, можуть бути використані для налаштування систем фільтрації мережевого трафіку, таких як брандмауери та системи виявлення та запобігання вторгнення (IDS/IPS), що дозволяє їм автоматично адаптуватися до мінливих патернів атак.

Додатково, аналітична здатність моделі може бути використана для передбачення та запобігання майбутнім атакам, аналізуючи тенденції та виявляючи нові вектори атак, що ще не були зафіксовані. Завдяки машинному навчанню, IDS стає не тільки інструментом для виявлення, але й для прогнозування та стратегічного планування захисту мережевої інфраструктури.

2.9. Висновок

Було детально розглянуто використання алгоритмів машинного навчання для розробки моделі, здатної виявляти DDoS атаки на мережевій інфраструктурі. Було здійснено огляд різних класифікаційних методів, включаючи Decision Tree, Gaussian Naive Bayes, Random Forest та MLP, та оцінено їх ефективність для ідентифікації зловмисного трафіку. По таким параметри як: точність, відгук та F1-бал, були зроблена оцінка кожного з алгоритмів, в результаті чого найкращим алгоритмом для виявлення DDoS атак став алгорим DecisionTree.

Модель, що була розроблена в ході дослідження, продемонструвала здатність ефективно відокремлювати легітимний трафік від потенційно шкідливого, що є ключовим для забезпечення безпеки мережі. Це дозволяє не тільки виявляти атаки в реальному часі, але й швидко реагувати на них, що є важливим для запобігання та мінімізації можливих збитків від DDoS інцидентів.

За результатами дослідження, можна зробити висновок, що використання машинного навчання для виявлення DDoS атак є перспективним напрямком і може стати в нагоді для розробників сучасних IDS систем. Однак, потрібно враховувати, що постійна еволюція та зміна паттернів атак вимагатимуть регулярного оновлення та доопрацювання моделей, щоб забезпечити їх актуальність та ефективність.

РОЗДІЛ 3. ЕКОНОМІЧНА ЧАСТИНА.

3.1. Розрахунок витрат

Створення системи захисту від DDoS атак є край важливою, оскільки від нього залежить наскільки підприємство буде знаходитися в стані простою, а отже не буде отримувати прибуток, але потрібно не забувати про економічну доцільність створення захисту, щоб витрати на захист були доцільними. Розраховується ціна для підприємства, яке має 1 сервер з підключенням до інтернету та для 5 комп'ютерів. 3 з них також мають вихід в інтернет.

Нормування праці в процесі створення системи захисту від DDoS атак істотно ускладнено через творчий характер праці спеціалістів з інформаційної безпеки. Проте трудомісткість розробки даної системи може бути розрахована на основі трудомісткості робіт, які виконуються.

3.1.1. Трудомісткість

Трудомісткість створення ПЗ визначається тривалістю кожної робочої операції, починаючи з складання технічного завдання і закінчуючи оформленням документації (за умови роботи одного програміста): Трудомісткість створення ПЗ визначається тривалістю кожної робочої операції, починаючи з складання технічного завдання і закінчуючи оформленням документації (за умови роботи одного програміста):

$$t = t_{тз} + t_{в} + t_{а} + t_{пр} + t_{опр} + t_{д}, \text{годин}, \quad (3.1)$$

де $t_{тз}$ – тривалість складання технічного завдання на розробку ПЗ,

де $t_{тз} = 7$ годин

$t_{в}$ – тривалість вивчення ТЗ, літературних джерел за темою тощо;

$t_{а}$ – тривалість розробки блок-схеми алгоритма;

$t_{пр}$ – тривалість програмування за готовою блок-схемою;

$t_{опр}$ – тривалість опрацювання програми на ПК;

$t_{б}$ – тривалість підготовки технічної документації на ПЗ.

Складові трудомісткості визначаються на підставі умовної кількості операторів у програмному продукті Q (з урахуванням можливих уточнень у процесі роботи над алгоритмом і програмою)

Умовна кількість операторів у програмі:

$$Q = q * c(1 + p), \text{штук} \quad (3.2)$$

Де q – очікувана кількість операторів;

c – коефіцієнт складності програми. Коефіцієнт визначає відносну складність програми щодо типового завдання. Діапазон його зміни: 1.25…2

p – коефіцієнт корекції програми в процесі її опрацювання. Коефіцієнт визначає збільшення обсягу робіт за рахунок внесення змін в алгоритм або програму внаслідок уточнення технічного завдання. Діапазон его зміни 0,05…0,1

Очікувана кількість операторів (q): Код містить багато операторів, включаючи присвоєння, умовні конструкції, цикли, виклики функцій тощо. Усього їх приблизно 200.

Коефіцієнт складності програми (c): Код є досить складним, оскільки він включає обробку даних, машинне навчання та різні умовні конструкції. Візьмемо середнє значення: 1.5.

Коефіцієнт корекції програми (p): Припустимо, що зміни будуть помірними, та оберемо середнє значення 0.075.

За формулою (3.2) Розрахуємо кількість операторів у програмі

$$Q=200*1.5*1.075=322.5$$

Тривалість вивчення технічного завдання складається з опрацювання довідкової літератури з урахуванням уточнення ТЗ і кваліфікації програміста можливо оцінити за формулою:

$$\text{tв} = \frac{Q * B}{(75 \ldots 85) * k}, \text{годин} \quad (3.3)$$

де B – коефіцієнт збільшення тривалості етапу внаслідок недостатнього опису завдання, B = 1,2…1,5;

k – коефіцієнт, що враховує кваліфікацію програміста і визначається стажем роботи за фахом:

- До 2 років – 0,8;
- Від 2 до 3 років 1,0;

- від 3 до 5 років 1,1…1,2;
- від 5 до 7 років - 1,3…1,4;
- понад 7 років - 1,5…1,6.

Внаслідок того що опис завдання буде дороблятися візьмемо середнє значення В = 1.3

Програміст, який зможе розробляти даний програмний код має мати кваліфікацію від 2 до 3 років, тому k = 1,0.

Розрахуємо тривалість вивчення ТЗ, літературних джерел за темою використовуючи формулу (3.3)

$t_в$= 6 годин

Тривалість розробки блок-схеми алгоритму:

$$ta = \frac{Q}{(20 \dots 25) * k}, \text{годин} \qquad (3.4)$$

Розрахуємо тривалість розробки блок-схеми алгоритму за формулою (3.4):

$t_а$=14 годин

Тривалість складання програми за готовою блок-схемою:

$$t\text{пр} = \frac{Q}{(20 \dots 25) * k}, \text{годин} \qquad (3.5)$$

Розрахуємо тривалість складання програми за готовою блок-схемою за формулою (3.5):

$t_{пр}$=13 годин

Тривалість опрацювання програми на ПК:

$$t\text{опр} = \frac{1{,}5Q}{(4 \dots 5) * k}, \text{годин} \qquad (3.6)$$

Розрахуємо тривалість опрацювання програми на ПК за формулою (3.6):

$t_{опр}$ = 95 годин

Тривалість підготовки технічної документації на ПЗ:

$$t\text{д} = \frac{Q}{(15 \dots 20) * k} + \frac{Q}{(15 \dots 20)} * 0{,}75, \text{годин} \qquad (3.7)$$

Розрахуємо тривалість підготовки технічної документації на ПЗ (3.7):

$t_д$ = 24 годин.

Підрахуємо трудомісткість використовуючи формулу (3.1):

де $t_{тз}$ = 7 годин, $t_в$ = 6 годин, $t_а$ = 14 годин, $t_{пр}$ = 13 годин, $t_{опр}$ = 95 годин, $t_д$ = 24 годин.

$$t = 7 + 6 + 14 + 13 + 95 + 24 = 159 \text{ годин.}$$

3.1.2. Розрахунок витрат на створення програмного захисту від DDoS атак

Витрати на розробку програмного захисту від DdoS-атак $К_{пз}$ складаються з витрат на заробітну плату спеціаліста з інформаційної безпеки $З_{зп}$ і вартості витрат машинного часу, що необхідний для розробки системи захисту $З_{мч}$

$$Кпз = Ззп + Змч \tag{3.8}$$

Заробітна плата виконавця враховує основну і додаткову заробітну п а також відрахування на соціальне потреби (пенсійне страхування, страхування на випадок безробіття, соціальне страхування тощо) и визначається за формулою:

$$Ззп = t * Зіб, грн \tag{3.9}$$

де *t* – загальна тривалість розробки системи захисту, годин;

Зіб – середньо-годинна заробітна плата спеціаліста з інформаційної безпеки з нарахуваннями, грн/годину.

Заробітна плата в місяць спеціаліста інформаційної безпеки становить 28800, відповідно щоб отримати час за годину потрібно цю суму розділити на 160 годин.

$$З_{ЗП} = 159 * 180 = 28\,620$$

Вартість машинного часу для розробки політики безпеки інформації на ПК визначається за формулою:

$$Змч = tопр * Смч + tд, грн \tag{3.10}$$

де *t* – трудомісткість розробки політики безпеки інформації на ПК, годин;

$C_{мч}$ – вартість 1 години машинного часу ПК, грн./година.

Вартість 1 години машинного часу ПК визначається за формулою:

(3.11)

$$\text{Смч} = \text{P} * \text{t нал} * \text{Се} + \frac{\text{Фзал} * \text{На}}{Fp} + \frac{\text{Клпз} * \text{Напз}}{Fp}, \text{грн}$$

де P =– встановлена потужність ПК, кВт;

$t_{нал}$ – річний час використання ПК на рік у годинах

C_e – тариф на електричну енергію, грн/кВт година;

$Ф_{зал}$ – залишкова вартість ПК на поточний рік, грн.;

H_a – річна норма амортизації на ПК, частки одиниці;

$H_{апз}$ — річна норма амортизації на ліцензійне програмне забезпечення, частки одиниці;

$K_{лпз}$ – вартість ліцензійного програмного забезпечення, грн.;

F_p – річний фонд робочого часу (за 40-годинного робочого тижня F_p = 1920).

$С_{мч}$ = 0.6 * 3 *2.64 + ((9000*0.25)/1920) + ((7000*0.2)/1920) = 6,65 грн

$З_{мч}$= 95*6.65 + 24 =656 грн

$К_{пз}$ = 656+28 620=29276 грн

Залишкова вартість ПК визначається виходячи з фактичного терміну його експлуатації як різниця між первісною вартістю та зносом за час використання.

Визначена таким чином вартість розробки політики безпеки *Крп* є частиною одноразових капітальних витрат разом з витратами на придбання і налагодження апаратури системи інформаційної безпеки.

Таким чином, капітальні (фіксовані) витрати на проектування та впровадження проектного варіанта системи інформаційної безпеки складають:

$$\text{К} = \text{Крп} + \text{Кзпз} + \text{Кпз} + \text{Каз} + \text{Кнавч} + \text{Кн} \qquad (3.12)$$

де $К_{рп}$ – вартість розробки проекту інформаційної безпеки та залучення для цього зовнішніх консультантів, тис. гри. Зовнішні консультанти не наймалися ($К_{рп}$=1500)

$К_{зпз}$ —закупівельна вартість ліцензійного основного й додаткового програмного забезпечення (ПЗ), тис. грн;

Таблиця 3.1 – Перелік придбаного Пз

№	Назва	Кількість	Вартість за 1 шт
1	Microsoft Windows 10	1	1050 грн
2	Microsotf visual studio	1	1000 грн
Всього: $К_{зпз}$ = 2050 грн			

$К_{рп}$ – вартість розробки системи захисту від DDoS-атак, тис. гри; ($К_{пз}$ =29276)

$К_{аз}$ – Створюється своє апаратне забезпечення яке буде продивлятися вхідний трафік та відфільтровувати його. Приблизна ціна 300грн

$К_{навч}$ — витрати на навчання технічних фахівців і обслуговуючого персоналу, тис. гри; ($К_{навч}$ =1200грн)

$К_{н}$ – витрати на встановлення обладнання та налагодження системи інформаційної безпеки, тис. гри. ($К_{н}$ = 800)

За формулою (3.12) розрахуємо капітальні витрати:

К =1500+2050+29276+1200+800 = 34826 грн

3.1.3. Розрахунок поточних (експлуатаційних) витрат

Експлуатаційні витрати – це поточні витрати на експлуатацію та обслуговування об’єкта проектування за визначений період (наприклад, рік), що виражені у грошовій формі.

$$С = Св + Ск + Сак, тис. грн \quad (3.13)$$

Де $С_{в}$ – Витрати на Upgrade-відновлення й модернізацію системи інформаційної безпеки.

$С_{к}$ – витрати на керування системою в цілому;

$С_{ак}$ – «активність користувача», витрати викликані активністю користувачів системи інформаційної безпеки.

$С_{в}$ – орієнтовно визначимо виходячи із вагової частки статей витрат (21%) з сукупної вартості інформаційної системи

$$С_в = 34826 * 0.21 = 7313 \text{ грн.}$$

Витрати на керування системою інформаційної безпеки ($С_к$) складають:

$$Ск = Сн + Са + Сз + Сєв + Сел + со + Стос, грн \quad (3.14)$$

Витрати на навчання адміністративного персоналу и кінцевих користувачів визначаються за даними організації з проведення тренінгів персоналу, курсів підвищення кваліфікації тощо (Сн=3000 грн).

Річний фонд амортизаційних відрахувань ($С_а$) визначається у відсотках від суми капітальних інвестицій за видами основних фондів і нематеріальних активів (ПЗ); Вартість купівлі ліцензійного ПЗ – 2050 грн., мінімальний строк дії користування - 2 роки

$$С_а = 2050/2 * 100 = 1025 \text{ грн/рік}$$

Річний фонд заробітної плати інженерно-технічного персоналу, що обслуговує систему інформаційної безпеки ($С_З$), складає:

$$Сз = Зосн + Здод, грн \quad (3.15)$$

Де $З_{осн}$ $З_{дод}$ – основна і додаткова заробітна плата відповідно, грн на рік.

Основна заробітна плата визначається, виходячи з місячного посадового окладу, а додаткова заробітна плата – в розмірі 8-10% від основної заробітної плати.

$$З_{дод} = 345\ 600 * 9\% = 7906 \text{ грн}$$

Основна Заробітна плата в місяць спеціаліста інформаційної безпеки становить ($З_{осн}$ = 28880 грн/місяць)

$$С_з = 345\ 600 + 27648 = 373248 \text{ грн}$$

До річного фонду заробітної плати додається єдиний внесок на загальнообов'язкове державне соціальне страхування – консолідований страховий внесок, збір якого здійснюється відповідно до класів професійного ризику виробництва, до яких віднесено платників єдиного внеску, з урахуванням видів їх економічної діяльності.

Розмір єдиного внеску на загальнообов'язкове державне соціальне страхування визначається на підставі встановленого чинним законодавством

відсотка від суми основної та додаткової заробітної плати (за узгодженням з консультантом економічної частини дипломного проекту)

$$С_{ссв}= 28800 *0.22 = 6336 \text{ грн}$$

Вартість електроенергії', що споживається апаратурою системою інформаційної безпеки протягом року ($С_{ел}$), визначається за формулою:

$$Сел = Р * FP * Це, грн \quad (3.16)$$

де Р – встановлена потужність апаратури інформаційної безпеки, кВт;

$$Р = 0,6 * 4 кВТ + (1 сервер + 1 комп.) * 0,7 Квт = 4 кВт$$

$F_р$ – річний фонд робочого часу системи інформаційної безпеки (працює кожен день с 9:00 до 18:00 10 годин на добу);

FP = 365 днів * 24 годин * (сервер та комп'ютер) + 1920 * 2 комп'ютера = 21 360 годин.

Це – тариф на електроенергію, грн/кВт-годин. (Це = 2.64)

$$С_{ел} = 4 *2.64 * 21 360 =225 561 \text{ грн}$$

Витрати на залучення сторонніх *організацій* для виконання деяких видів обслуговування, навчання та сертифікацію обслуговуючого персоналу визначаються за даними організації. ($С_о = 0$)

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки ($С_{тос}$) визначаються за даними організації або у відсотках від вартості капітальних витрат (1-3%).

$$К = 34826 \text{ грн}$$

$$С_{тос} = 34826 * 2 \% = 697 \text{ грн}$$

Витрати, викликані активністю користувачів системи інформаційної безпеки ($С_{ак}$) можна орієнтовно визначити, користуючись даними табл. 1 про вагові частки статей витрат у сукупній вартості системи інформаційної безпеки.

$$С_{ак} =34826 * 46\% = 16019 \text{ грн}$$

Розрахуємо витрати на керування системою інформаційної безпеки за формулою (3.14):

$$С_к = 3000 + 1025 + 373248+ 6336 + 225 561 + 0 + 697 = 609 867 \text{ грн}$$

У кожному конкретному випадку можуть бути враховані й інші види поточних витрат, що визначаються специфікою експлуатації проектованої системи інформаційної безпеки.

3.2. Оцінка Величини збитку

Для розрахунку вартості збитку необхідно застосувати спрощену модель оцінки

Необхідні такі дані для розрахунку:

$t_п$ - час простою вузла або сегмента корпоративної мережі внаслідок атаки, 4 годин;

$t_в$ - час відновлення після атаки персоналом, що обслуговує корпоративну мережу, 3 годин;

$t_{ви}$ - час повторного введення загубленої інформації співробітника атакованого вузла або сегмента корпоративної мережі, 3 годин

$З_с$ - заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі, 20000 грн на місяць

$З_о$ - заробітна плата обслуговуючого персоналу (адміністраторів та ін.) 28000 грн на місяць

$Ч_о$ - чисельність обслуговуючого персоналу (адміністраторів та ін. осіб) 1 особи

$Ч_с$ - чисельність співробітників атакованого вузла або сегменту корпоративної мережі, 3 осіб

О - обсяг продажів атакованого вузла або сегмента корпоративної мережі, 650 000 тис. грн у рік;

$П_{зч}$ - вартість заміни встаткування або запасних частин, 3000 грн;

I – число атакованих вузлів або сегментів корпоративної мережі 2;

N – середнє число атак на рік. 50

Упущена вигода від простою атакованого вузла або сегмента корпоративної мережі становить:

$$U = Пп + Пв + V \quad (3.17)$$

де $П_п$ – оплачувані втрати робочого часу та простої співробітників атакованого вузла або сегмента корпоративної мережі, грн;

$П_в$ – вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн,

V – втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності співробітників атакованого вузла або сегмента корпоративної мережі являють собою втрати їхньої заробітної плати (оплата непродуктивної праці) за час простою внаслідок атаки:

$$Пп = \left(\frac{\sum Зс}{F}\right) * tп, грн \qquad (3.18)$$

де F – місячний фонд робочого часу (при 40-а годинному робочому тижні становить 176 ч).

$З_с$ – заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі, 20000 грн на місяць;

tп — час простою вузла або сегмента корпоративної мережі внаслідок атаки, 4 годин;

Пп= ((20000/176)*3)*4= 1363 грн

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі включають кілька складових:

$$Пв = Пви + Ппв + Пзч \qquad (3.19)$$

де $П_{ви}$ – витрати на повторне уведення інформації, грн;

$П_{пв}$ – витрати на відновлення вузла або сегмента корпоративної мережі,

$П_{зч}$ – вартість заміни устаткування або запасних частин, грн.

Витрати на повторне введення інформації $П_{ви}$ розраховуються виходячи з розміру заробітної плати співробітників атакованого вузла або сегмента корпоративної мережі $З_с$, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ви}$: ($t_{ви}$ = 2)

$$Пви = \frac{\sum Зс}{F} * t_{ви} \quad (3.20)$$

Пви = ((20000/176)*3)*3 = 1022 грн

Витрати на відновлення вузла або сегмента корпоративної мережі Ппв визначаються часом відновлення після атаки $t_В$ і розміром середньо-годинної заробітної плати обслуговуючого персоналу (адміністраторів):

$$Ппв = \frac{\sum Зо}{F} * tв \quad (3.21)$$

$П_{ПВ}$ = (28000/176)*3 = 477 грн

$П_в$ = 1022 + 477 + 3000 = 4499 грн

Втрати від зниження очікуваного обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі визначаються виходячи із середньо-годинного обсягу продажів і сумарного часу простою атакованого вузла або сегмента корпоративної мережі: (О=750 000)

$$V = \frac{O}{Fr} * (tп + tв + tви) \quad (3.22)$$

Де Fr – річний фонд часу роботи організації (52 робочих тижні, 5-ти денний робочий, тиждень 8-ми годинний робочий день) становить близько 2080 ч.

V = (650000/2080) *(4+3+3) = 3125 грн

Упущена вигода від простою атакованого вузла або сегмента мережі становить:

U = 1363 + 4499 + 3125 = 8993 грн

Таким чином загальний збиток від атаки на вузол або сегмент корпоративної мережі організації складе:

$$B = \sum i \sum n\, U \quad (3.23)$$

Де і – число атакованих вузлів, 2 комп'ютера;

N – середнє число атак на рік: 30 разів.

В = 8993 * 2 * 50 =899 370грн

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної безпеки і становить:

$$E = B * R - C \quad (3.24)$$

де B – загальний збиток від атаки на вузол або сегмент корпоративної мережі, тис. гри;

R – очікувана імовірність атаки на вузол або сегмент корпоративної мережі, 0,7 частки одиниці;

C – щорічні витрати на експлуатацію системи інформаційної безпеки, тис. гри.

$$E = 899\,370 * 0.7 - 609\,867 = 19643 \text{ грн}$$

3.3. Визначення та аналіз показників економічної ефективності

Показник сукупної вартості володіння (ТСО) використовується, якщо величину відверненого збитку від атаки на вузол або сегмент корпоративної мережі важко або неможливо визначити у вартісній формі.

У цьому випадку необхідно порівняти сукупну вартість володіння, розраховану для двох варіантів проектного рішення щодо створення або удосконалення системи інформаційної безпеки, і вибрати варіант із найменшою з них.

Коефіцієнт повернення інвестицій ROSI показує, скільки гривень додаткового прибутку приносить одна гривня капітальних інвестицій на впровадження системи інформаційної безпеки.

Щодо до інформаційної безпеки говорять не про прибуток, а про запобігання можливих втрат від атаки на сегмент або вузол корпоративної мережі, а отже:

$$\text{ROSI} = \frac{E}{K}, \text{частка одиниці} \qquad (3.25)$$

де Е – загальний ефект від впровадження системи інформаційної безпеки (розділ 3.2 методичних вказівок, формула 3.8), 19643 грн;

К – капітальні інвестиції за варіантами, що забезпечили цей ефект, тис. гри.

Якщо порівнюється два варіанти системи інформаційної безпеки, то обирається варіант з більшим значенням ROSI.

$$\text{Rosi} = 19643/34826 = 0.55$$

Для остаточної оцінки варіантів і вибору найбільш ефективного з них необхідно порівняти розрахункове значення ROSI з бажаним значенням показника ефективності Ен.

Проект системи інформаційної безпеки визнається доцільним за умови

$$ROSI > EH$$

При $ROSI<E_H$ варіант є збитковим і більш економічним визнається відмова від його реалізації.

Для вибраного варіанта визначається розрахунковий строк окупності капітальних інвестицій T_p.

Термін окупності капітальних інвестицій T_p показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки:

$$To = \frac{K}{E} = \frac{1}{ROSI} \text{ років} \qquad (3.26)$$

Якщо варіанти економічно рівноцінні, то приймається варіант, що забезпечує більш високу надійність, поліпшення умов праці.

$$T_o = 34826/19643 = 1{,}81 \text{ років}$$

3.4. Висновок

Згідно з наведеними розрахунками можливо зробити висновок, що розробка підходу до виявлення атак DDoS атак в мережі підприємства з використанням алгоритмів машинного навчання є економічно доцільною.

Капітальні витрати, які складають 34826 грн, дозволяють отримати ефект величиною 19643 грн. Відповідно до отриманих значень показників економічної можна зазначити, що запропонований підхід дозволить отримувати 0,55 економічного ефекту на 1 грн. капітальних витрат (коефіцієнт повернення інвестицій ROSI складає 0,55 грн.). Термін окупності при цьому складатиме 1,81 років.

ВИСНОВКИ

Було проведено огляд та оцінка ефективності різних класифікаційних алгоритмів машинного навчання, таких як Decision Tree, Gaussian Naive Bayes, Random Forest та MLP, для ідентифікації зловмисного трафіку. Серед цих алгоритмів, Decision Tree виявився найбільш ефективним для виявлення DDoS атак.

Розроблена модель демонструє високу здатність до ефективного відокремлення легітимного трафіку від потенційно шкідливого, що є ключовим аспектом для забезпечення безпеки мережі. Ця модель дозволяє не тільки виявляти атаки в реальному часі, але й швидко реагувати на них, запобігаючи та мінімізуючи можливі збитки від DDoS інцидентів.

Дане дослідження підтверджує, що використання машинного навчання для виявлення DDoS атак є перспективним напрямком. Це може стати в нагоді для розробників сучасних IDS (Intrusion Detection Systems) систем. Важливо враховувати необхідність регулярного оновлення та доопрацювання моделей для забезпечення їх актуальності та ефективності у відповідь на постійно еволюціонуючи патерни атак.

Капітальні витрати на розробку системи захисту від DDoS атак складають 34826 грн, а коефіцієнт повернення інвестицій (ROSI) складає 0.55 грн., що свідчить про економічну доцільність запропонованого підходу. Окупність інвестицій у розробку системи інформаційної безпеки з використанням машинного навчання становить приблизно 1.81 року, що також підкреслює її економічну ефективність.

ПЕРЕЇЛК ПОСИЛАНЬ

1. Найбільші DDoS-атаки за перше півріччя 2020 [Електронний ресурс] // 10Guards. – 2020. – Режим доступу до ресурсу: https://10guards.com/ua/articles/ddos-attacks-doubled-in-q2-2020/.

2. Siddhesh J. A look at SPAM — the underworld of the internet! [Електронний ресурс] / Joglekar Siddhesh // Medium. – 2009. – Режим доступу до ресурсу: https://medium.com/siddheshj/a-look-at-spam-the-underworld-of-the-internet-c124ee0c80a4.

3. Воробієнко П. П. ТЕЛЕКОМУНІКАЦІЙНІ ТА ІНФОРМАЦІЙНІ МЕРЕЖІ / П. П. Воробієнко, Л. А. Нікітюк, П. І. Резніченко. – Київ: САММІТ-Книга, 2010. – 708 с.

4. Qijun G. Denial of Service Attacks [Електронний ресурс] / G. Qijun, L. Peng – Режим доступу до ресурсу: https://s2.ist.psu.edu/paper/DDoS-Chap-Gu-June-07.pdf.

5. About Eggdrop [Електронний ресурс] // Eggdrop Development. – 2022. – Режим доступу до ресурсу: https://www.eggheads.org/about/.

Kellermann T. BOTs: Cyber -zombies [Електронний ресурс] / Tom Kellermann // World Bank Treasury Security Team. – 2004. – Режим доступу до ресурсу: https://documents1.worldbank.org/curated/zh/173421468142165710/pdf/359060Bots0rev01PUBLIC1.pdf.

6. Zeus (malware) [Електронний ресурс] // Wikipedia. – 2016. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Zeus_(malware).

7. Mirai (malware) [Електронний ресурс] // Wikipredia. – 2018. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Mirai_(malware).

8. UDP flood attack [Електронний ресурс] // CloudFlare – Режим доступу до ресурсу: https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/.

9. Ping flood (ICMP flood) [Електронний ресурс] // Imperva – Режим доступу до ресурсу: https://www.imperva.com/learn/ddos/ping-icmp-flood/.

IP Fragmentation Attack [Електронний ресурс] // imperva – Режим доступу до ресурсу: https://www.imperva.com/learn/ddos/ip-fragmentation-attack-teardrop/.

10. ЗАКОН УКРАЇНИ \"Про інформацію\". // Верховної Ради України. – 1992. – №48. – С. 650.

11. ЗАКОН УКРАЇНИ \"Про захист інформації в інформаційно-комунікаційних системах\". // Верховної Ради України. – 1994. – №31. – С. 286.

12. Загальні поняття про системи виявлення та запобігання вторгненням [Електронний ресурс] // Habr. – 2019. – Режим доступу до ресурсу: https://habr.com/ru/companies/otus/articles/479584/.

13. Що таке SYN-flood атака? [Електронний ресурс] // StormWall – Режим доступу до ресурсу: https://stormwall.pro/knowledge-base/attack/syn-attack.

14. Chio C. Machine Learning and Security / C. Chio, D. Freeman. – Boston: O*Reilly, 2020. – 388 с.

15. Теорема Баєса [Електронний ресурс] // Wikipedia. – 2013. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%91%D0%B0%D1%94%D1%81%D0%B0.

16. KDD Cup 1999 Data [Електронний ресурс] // The UCI KDD Archive. – 1999. – Режим доступу до ресурсу: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Електронний ресурс]

17. DDoS Evaluation Dataset [Електронний ресурс] // UNB. – 2019. – Режим доступу до ресурсу: https://www.unb.ca/cic/datasets/ddos-2019.html.

18. Prasad D. Machine Learning DDoS Detection Using Stochastic Gradient Boosting / D. Prasad, P. Babu, A. C. // International Journal of Computer Sciences and Engineering. – 2019. – С. 10.

19. Why Python for Machine Learning? [Електронний ресурс] // Python. – 2021. – Режим доступу до ресурсу: https://pythonbasics.org/why-python-for-machine-learning/.

20. Google Colaboratory [Електронний ресурс] // Google – Режим доступу до ресурсу: https://colab.research.google.com/?hl=ru.

21. Receiver Operating Characteristic [Електронний ресурс] // Springer – Режим доступу до ресурсу: https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_569

22. Bishom C. M. Pattern Recognition and Machine learning / Christopher Bishom., 2006. – 758 с.

23. The CAIDA \"DDoS Attack 2007\" Dataset [Електронний ресурс] // CAIDA. – 2007. – Режим доступу до ресурсу: https://www.caida.org/catalog/datasets/ddos-20070804_dataset/.

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість	Примітки
Документація				
1	А4	Реферат	2	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	3	
4	А4	Вступ	1	
5	А4	Стан Питання. Постановка задачі	28	
6	А4	Спеціальна частина	31	
7	А4	Економічний розділ	13	
8	А4	Висновки	1	
9	А4	Перелік посилань	3	
10	А4	Додаток А	1	
11	А4	Додаток Б	1	
12	А4	Додаток В	1	
13	А4	Додаток Г	1	

ДОДАТОК Б. Перелік документів на оптичному носії

ДОДАТОК В. Відгук керівника кваліфікаційної роботи

В І Д Г У К

на кваліфікаційну роботу студента групи 125м-22-2 Явіра Ярослава Романовича

на тему: «Дослідження алгоритмів виявлення DDoS атак із використанням машинного навчання.»

Пояснювальна записка складається зі вступу, трьох розділів і висновків, розташованих на ___ сторінках.

Мета роботи є актуальною, оскільки вона спрямована на підвищення освітлення проблем захисту від DDdos атак які є великою проблемою сьогодення особливо для багатьох інтернет додатків.

При виконанні роботи автор продемонстрував добрий рівень теоретичних знань і практичних навичок. На основі аналізу різноманітних методів та алгоритмів машинного навчання для виявлення цих типів атак, був сформульований найбільш відповідний метод захисту від даних атак.

Практична цінність роботи полягає у тому, що представлений метод захисту є повноцінною можливістю уникнення атак та забезпечення доступності інформації.

Рівень запозичень у кваліфікаційній роботі не перевищує вимог «Положення про систему виявлення та запобігання плагіату».

В цілому робота задовольняє усім вимогам, а її автор Явір Я. Р. заслуговує на оцінку « » та присвоєння кваліфікації «Магіср з кібербезпеки» за спеціальністю 125 Кібербезпека.

ДОДАТОК Г. Відгук керівника економічного розділу

Економічний розділ виконаний відповідно до вимог, які ставляться до кваліфікаційних робіт, та заслуговує на оцінку 95 б. («відмінно»).

Керівник розділу _____ Доцент. к.е.н. Пілова Д.П.

(підпис) (ініціали, прізвище)