Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп’ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«___» _____ 2024 р

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: «**Комп'ютерна модель виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання**»

Захищено на засіданні
Атестаційної комісії № 42
протокол № __ від __.06.2024 р.
Оцінка _____ /_____
Голова Атестаційної комісії
_____ **СКОБ Ю.О.**

Виконав:
студент 4 курсу, групи КІ–41
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 – Комп’ютерна інженерія.
ЛАНІН Євген Сергійович_____

Керівник: доцент кафедри теоретичної та прикладної системотехніки, канд. техн наук, доц.
БАКУМЕНКО Ніна Станіславівна_____

Рецензент: д.т.н., доцент, професор закладу вищої освіти кафедри теоретичної та прикладної інформатики факультету математики і інформатики
РУККАС Кирило Маркович_____

Харків – 2024

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і чотирьох додатків. Загальний обсяг роботи складає 67 сторінок, із яких 43 сторінки основної частини з 14 рисунками, 3 таблицями, 13 найменуваннями списку використаних джерел та чотирьома додатками.

Метою кваліфікаційної роботи є розробка комп'ютерної моделі для виявлення атак в комп'ютерних мережах, використовуючи методи машинного навчання.

Об’єкт дослідження – процес виявлення атак в комп’ютерних мережах на підставі даних мережевого трафіку.

Предмет дослідження – методи машинного навчання, що застосовуються для виявлення аномальної активності в мережевому трафіку та ідентифікації атак.

Проблема, яка вирішується в кваліфікаційній роботі полягає в тому, щоб скориставшись існуючими програмними засобами машинного навчання підвищити рівень безпеки мереж та мінімізувати негативні наслідки атак.

Область застосування – захист комп'ютерних мереж. Розроблена комп'ютерна модель може широко використовуватися в сфері малого та середнього IT-бізнесу, а також у великих корпораціях та державних установах для забезпечення високого рівня захисту інформаційних систем кіберзагроз.

Ключові слова: ШІ, МШ, DDoS, комп'ютерна мережа, виявлення атак, аномальна активність, модель, мережевий трафік, класифікація.

ABSTRACT

The explanatory note to the bachelor's thesis consists of an introduction, three chapters, conclusions, a list of references and four appendices. The total volume of the work is 67 pages, of which 43 pages are the main part with 14 figures, 3 tables, 13 references and four appendices.

The problem that is solved in the qualification work is to use existing machine learning software tools to increase the level of network security and minimise the negative effects of attacks.

The field of application is the protection of computer networks. The developed computer model can be widely used in small and medium-sized IT businesses, as well as in large corporations and government agencies to ensure a high level of protection of information systems from cyber threats.

Keywords: AI, MS, DDoS, computer network, attack detection, anomalous activity, model, network traffic, classification.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

МШ - машинне навчання;

ШІ - штучний інтелект.

ВСТУП

У сучасному цифровому світі, де комп'ютерні мережі пов'язані з усіма сферами життя, забезпечення мережевої безпеки є надзвичайно важливим питанням. Експоненціальне зростання швидкості та обсягів обміну інформацією в мережах створює як нові можливості для розвитку, так і виклики у сфері кібербезпеки. Атаки на комп'ютерні системи та мережі стають все більш витонченими та складними, загрожуючи конфіденційності, цілісності та доступності даних, що має серйозні наслідки для організацій та окремих осіб.

Актуальність роботи. Постійне зростання обсягів даних і зростаюча складність атак на комп'ютерні мережі вимагають вдосконалення захисту. Використання методів машинного навчання для виявлення атак в комп'ютерних мережах є одним з найбільш перспективних напрямків в цій галузі. Актуальність даного дослідження полягає в тому, що воно спрямоване на розробку комп'ютерних моделей, які можуть своєчасно виявляти та реагувати на потенційно небезпечну поведінку в мережах, забезпечуючи тим самим надійний захист від кіберзагроз.

Основні виклики, що стоять перед дослідниками в цій галузі, включають розробку ефективних алгоритмів виявлення атак, збір та обробку великих обсягів даних для навчання моделей, а також забезпечення швидкості та точності виявлення небезпечної активності. Успіх у вирішенні цих завдань сприятиме побудові більш безпечних та надійних комп'ютерних систем та мереж, що є необхідним для подальшого розвитку сучасного суспільства.

Метою даної роботи є забезпечення більш стабільної роботи нейронних мереж шляхом виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання. Завданням даного дослідження є аналіз існуючих підходів до виявлення атак, розробка нових алгоритмів та їх впровадження на практиці з метою підвищення ефективності захисту комп'ютерних мереж від кіберзагроз.

Комп’ютерна модель для виявлення атак в комп’ютерних мережах сприятиме підвищенню безпеки комп'ютерних систем та мереж.

РОЗДІЛ 1
МЕТОДИ ВИЯВЛЕННЯ АТАК В КОМП'ЮТЕРНИХ МЕРЕЖАХ

1.1 Вступ до виявлення атак в комп'ютерних мережах

Оскільки залежність світу від цифрових технологій зростає, безпека комп'ютерних мереж стає все більш важливою. Комп'ютерні мережі можна атакувати різними способами, наприклад за допомогою вірусів, шпигунського програмного забезпечення, фішингу, атак типу «відмова в обслуговуванні» та розподілених атак «відмова в обслуговуванні». Ці атаки можуть знищити дані, спричинити перебої в роботі систем і фінансові збитки, а їх своєчасне виявлення та блокування важливо для забезпечення цілісності та доступності мережевих ресурсів.

Для запобігання таких випадків розроблено методи виявлення атак, метою яких є виявлення мережевої активності, що може потенційно становить загрозу. Ефективні системи виявлення атак здатні розпізнавати як відомі типи атак, так і адаптуватися до нових загроз. Базове виявлення атак використовує методи сигнатур для порівняння мережевої активності з базою даних відомих атак і методів виявлення аномалій, які шукають відмінності в мережевій активності від звичайної.

Машинне навчання та штучний інтелект дозволяють точніше виявляти атаки, дозволяючи системам пристосовуватися до нових і нових загроз, збільшуючи швидкість і точність виявлення та зменшуючи кількість помилкових спрацьовувань.

Деякі проблеми виявлення атак пов'язані з необхідністю обробки великих обсягів даних у режимі реального часу, а також тонким балансом між точністю виявлення та конфіденційністю користувача.

1.2 Традиційні методи виявлення атак у комп'ютерних мережах

Як випливає з назви, традиційні методи виявлення атак у комп'ютерних мережах базуються на перевірених технологіях і практиках. Вони являють собою набір підходів, які допомагають ідентифікувати та блокувати загрози для

обчислювальної інфраструктури. Серед таких методів – системи виявлення вторгнень і системи запобігання вторгненням, дві найважливіші технології, задіяні в безпеці мережі.

Перший тип технологій – це системи виявлення вторгнень, вони можуть бути двох типів:

- на основі сигнатур
- на основі аномалій

Перші порівнюють поточний мережевий трафік із набором шаблонів або «сигнатур», пов’язаних із відомими атаками. У результаті, хоча системи виявлення вторгнень виявилися дуже ефективними проти переважної більшості відомих загроз, вони майже марні проти нових або модифікованих загроз. Системи виявлення вторгнень на основі аномалій порівнюють поточні дані з історичними, щоб визначити їх відхилення. Отже, вони можуть бути більш ефективними, коли йдеться про виявлення нових типів загроз, але ризик помилкових спрацьовувань залишається високим.

Другий тип технології, системи запобігання вторгненням, розвинувся з систем виявлення вторгнень. Окрім виявлення атак, вони блокують відомі загрози та не дають шкідливим пакетам досягти мети.

1.3 Використання машинного навчання і штучного інтелекту у виявленні атак на комп'ютерні мережі

Новою перевагою в гонці озброєнь між зловмисниками та захисниками комп’ютерних мереж є машинне навчання та штучний інтелект, які допомагають ефективніше боротися з атаками. Нижче наведені основні переваги машинного навчання і, зокрема, штучного інтелекту в боротьбі з атаками.

Природа моделей, що самонавчається, дозволяє їм навчатися на історичних даних, що дає змогу ідентифікувати зловмисні дії, навіть якщо вони суттєво відрізняються від раніше відомих атак. Це включає виявлення нульових атак та швидку адаптацію до змінюваних векторів загроз.

Штучний інтелект може аналізувати великі обсяги даних, що дає змогу зменшити помилкові спрацьовування та підвищує ймовірність блокування

загроз. Прогнозні здібності аналітичних систем у сфері штучного інтелекту характеризують здатність не лише реагувати на поточні інциденти, а й передбачити подальші жертви та потенційні атаки.

Серед найпопулярніших підходів:

1. Навчання з вчителем: Модель навчається на попередньо позначених даних нормального та ненормального трафіку та здатна класифікувати нові дані як один із двох різновидів. Для класифікації використовуються такі алгоритми, як дерева рішень, випадкові ліси, логістична регресія.

2. Навчання без вчителя: Ці алгоритми дозволяють ідентифікувати складні шаблони в даних, які не були позначені до навчання моделі. Такий вид навчання має найкращу ефективність у виявленні атак, характеристики яких заздалегідь невідомі. Найчастіше це реалізується за допомогою кластеризації або алгоритмів виявлення аномалій.

Незважаючи на вищезазначені переваги, використання МШ та ШІ характеризується низкою проблем і викликів. Якість і доступність даних є першим із них, оскільки робота моделей значною мірою залежить від обсягу та якості даних – у разі неповної або нерелевантної вибірки модель може працювати неефективно. Можливість навмисного чи ненавмисного спотворення моделей машинного навчання, також слід враховувати.

Таким чином, розвиток ML та AI для виявлення атак продовжить зростати, створюючи умови для розробки більш автоматизованих та інтелектуальних систем. В результаті сектор захисту інформації стане більш ефективним і менш обтяжливим для аналітиків.

1.4 Хмарні та розподілені системи виявлення атак у комп'ютерних мережах

Концепція систем кіберзахисту була сформована розвитком хмарних і розподілених технологій. Концепція хмарного та розподіленого виявлення є прямим наслідком появи нових розподілених систем і надає інструменти для більш гнучкого та економічно доцільного підходу.

Переваги хмарних і розподілених систем виявлення

1. Масштабованість: Хмарні рішення можуть легко масштабуватися, щоб відповідати зростаючим потребам безпеки, дозволяючи швидко збільшувати або зменшувати ресурси в залежності від поточного навантаження або рівня загроз.

2. Гнучкість та доступність: Системи, розміщені в хмарі, доступні з будь-якої точки світу, що дозволяє централізовано управляти захистом для географічно розосереджених мереж.

3. Економічна ефективність: Хмарні сервіси знижують потребу в інвестиціях в обладнання та обслуговуванні інфраструктури, забезпечуючи організаціям засоби використання найсучасніших технологій без значних капіталовкладень.

Хмарні системи виявлення вторгнень використовують хмарні технології для збору та аналізу даних мережевого трафіку. Це дозволяє визначати потенційні атаки практично в реальному часі, значно скорочуючи час відповіді системи.

У розподілені системи виявлення вторгнень моніторинг і аналіз виконуються паралельно в різних точках мережі, що забезпечує глибший рівень нагляду і більш швидке реагування на інциденти.

Виклики і стратегії подолання:

- Управління даними та конфіденційність: робота з великими обсягами даних у хмарі загрожує конфіденційності користувачів, тому необхідно мати високий рівень захисту даних.
- Залежність від постачальників послуг: використання хмарних технологій для визначення атак також може призвести до надмірної залежності від постачальників хмарних послуг, особливо щодо конфігурації та інтеграції.
- Затримка та продуктивність: забезпечення високої продуктивності та мінімальної затримки для обробки всієї інформації про трафік, ймовірно, потребує високопродуктивних хмар і зручної конфігурації.

Подальший розвиток хмарних та розподілених систем виявлення атак залежить від інновацій в області штучного інтелекту та машинного навчання, а також від покращення технологій захисту даних і забезпечення приватності.

Інтеграція передових аналітичних інструментів і розвиток співпраці між організаціями та хмарними провайдерами можуть значно покращити ефективність і надійність цих систем, що зробить їх незамінними інструментами у забезпеченні кібербезпеки.

1.5 Майбутні тренди і виклики у виявленні атак на комп'ютерні мережі

Виявлення атак на комп'ютерні мережі постійно розвивається, адаптуючись до нових викликів та технологічних можливостей. Ось декілька ключових трендів, які можуть формувати майбутнє цієї галузі:

1. Інтеграція з ШІ та машинним навчанням: Адаптивні системи виявлення атак, які вже застосовують різноманітні компоненти штучного інтелекту та машинного навчання, стануть ще розумнішими та прогнозуватимуть, виявлятимуть різноманітні загрози та реагуватимуть на них у реальному часі. AI розширить можливості адаптивності в нових моделях, всебічно оновивши виявлення нових атак, зменшивши помилкові спрацьовування.

2. Зростання хмарних та гібридних систем безпеки: Оскільки більше компаній переходять до хмарних обчислень, хмарні та гібридні системи виявлення атак стануть нормою, надаючи гнучкість, масштабованість та ефективність управління безпекою.

3. Автоматизація реагування на інциденти: Автоматизація процесів реагування на загрозу, автоматичного блокування атаки та усунення вразливостей без втручання людини стане важливою частиною майбутніх систем безпеки.

4. Конфіденційність даних і управління: Із зростаючою кількістю нормативних актів для захисту персональних даних, таких як GDPR в ЄС, організаціям доведеться вирішувати дедалі складніші завдання щодо захисту особистої інформації після виявлення атаки.

Основними викликами у майбутньому стануть:

1. Забезпечення захисту від розширених постійних загроз (APT): APT стають складнішими і прихованими, що вимагає більш продуманих стратегій виявлення та реагування, що включають розширений моніторинг та аналітику поведінки.

2. Великі обсяги даних та їх аналіз: Збільшення обсягів даних ускладнює моніторинг та аналіз здатності організації виявляти загрози без додаткових капіталовкладень у обчислення.

Майбутнє виявлення атак комп'ютерних мереж видається досить захоплюючим і, водночас, складним, із неперервними інноваціями та адаптаціями, необхідними для захисту від постійно еволюціонуючих кіберзагроз.

Висновки за розділом 1

Зі зростанням залежності світу від цифрових технологій безпека комп'ютерних мереж стає критично важливою. Комп'ютерні мережі піддаються різноманітним атакам, таким як віруси, шпигунське програмне забезпечення, фішинг, атаки типу «відмова в обслуговуванні» та розподілені атаки «відмова в обслуговуванні». Ці атаки можуть мати серйозні наслідки, включаючи втрату даних, збої в роботі систем і фінансові збитки. Тому своєчасне виявлення та блокування атак є критично важливими для забезпечення цілісності та доступності мережевих ресурсів.

Об'єктом дослідження є процес виявлення атак в комп'ютерних мережах на підставі даних мережевого трафіку. Базове виявлення атак використовує методи сигнатур для порівняння мережевої активності з базою даних відомих атак і методи виявлення аномалій, які шукають відмінності між поточною мережею активністю та звичайною.

Предмет дослідження – методи машинного навчання, що застосовуються для виявлення аномальної активності в мережевому трафіку та ідентифікації атак. Машинне навчання та штучний інтелект дозволяють точніше виявляти атаки, дозволяючи системам адаптуватися до нових загроз, збільшуючи швидкість і точність виявлення та зменшуючи кількість помилкових спрацьовувань. Проблеми виявлення атак пов'язані з необхідністю обробки

великих обсягів даних у режимі реального часу, а також тонким балансом між точністю виявлення та конфіденційністю користувача.

Метою даної роботи є забезпечення більш стабільної роботи нейронних мереж шляхом виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання. Новітніми технологіями вважаються машинне навчання та штучний інтелект, які підвищують ефективність систем виявлення атак. Системи виявлення атак можуть розпізнавати як відомі типи атак, так і адаптуватися до нових загроз. Використання цих технологій дозволяє швидко ідентифікувати зловмисні дії та зменшувати кількість помилкових спрацьовувань. Безпека комп'ютерних мереж є ключовим елементом у сучасному цифровому світі. Ефективні методи виявлення атак, включаючи традиційні методи, машинне навчання та штучний інтелект, є критично важливими для захисту мережевих ресурсів від зловмисних дій. Розвиток цих технологій продовжуватиметься, сприяючи створенню більш автоматизованих і інтелектуальних систем виявлення атак.

РОЗДІЛ 2
МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АТАК В КОМП'ЮТЕРНИХ МЕРЕЖАХ

2.1 Вступ до машинного навчання

У сучасному світі важко уявити собі хоч один аспект життя, який не покращений хоча б частково завдяки машинному навчанню. Скрізь, де є завдання, що вимагають повторення, де потрібно аналізувати дані та робити висновки, машинне навчання може стати у пригоді. Особливо це стало помітно в останні кілька років, коли обчислювальні потужності значно зросли, а дані збираються й обробляються майже всюди. Найбільш поширені застосування машинного навчання включають рекомендаційні системи, розпізнавання зображень, обробку тексту, самокеровані автомобілі, виявлення спаму та багато іншого [11].

Машинне навчання найкраще порівнюється з імітацією людської «інтуїції». Інтуїція заснована на прийнятті рішення на основі минулого досвіду. У випадку машинного навчання, початковий набір даних слугує досвідом. Після обробки цього набору даних, система здатна приймати рішення на основі навчання. Таким чином, машинне навчання можна розглядати як спосіб навчити комп’ютер мислити подібно до людини.

Прикладом переваги машинного навчання над традиційними рішеннями може бути спам-фільтр. Якщо створюється спам-фільтр, використовуючи стандартні методи програмування, доведеться створити шаблони для ідентифікації спаму в електронних листах. Для правильної роботи програми необхідно розробити кілька алгоритмів. Однак такий підхід має значний недолік – фільтр постійно потребує оновлення та додавання нових правил. Це перетворює програму на довгий і складний список правил, який важко підтримувати.

На противагу, спам-фільтр, заснований на методах машинного навчання, автоматично аналізує, які слова і фрази є кращими предикторами спаму,

визначаючи часті шаблони слів у спамових повідомленнях порівняно з легітимними. Така програма ефективно виконує своє завдання, є коротшою, простішою в обслуговуванні та, ймовірно, точнішою [4].

Машинне навчання відмінно підходить для:

- Завдань, які вимагають безлічі дрібних налаштувань або довгих списків правил у традиційних рішеннях: один алгоритм машинного навчання часто може спростити процес і працювати ефективніше.
- Вирішення складних проблем, де традиційні методи не дають належних результатів: найсучасніші підходи машинного навчання можуть знайти оптимальні рішення.
- Ситуацій з мінливими умовами: система машинного навчання здатна адаптуватися до нових даних.
- Аналізу складних завдань та великих обсягів інформації.

Для кращого розуміння слід зазначити, що машинне навчання є складовою частиною штучного інтелекту. Основна мета штучного інтелекту – навчити комп’ютер приймати рішення на рівні людини. Існує два основних підходи для прийняття більшості рішень:

- За допомогою логіки та міркувань.
- Використовуючи власний досвід.

Обидва ці явища відображені в комп'ютерах, і в них є назва: штучний інтелект. Штучний інтелект – це назва процесу, в якому комп'ютер ухвалює рішення, наслідуючи людину. Тобто штучний інтелект поєднує описані вище два варіанти. Машинне навчання – це коли ми фокусуємося тільки на другому варіанті. А саме, коли комп'ютер ухвалює рішення, ґрунтуючись на досвіді. Цей досвід ми називаємо даними. Таким чином, машинне навчання – це коли комп'ютер приймає рішення, ґрунтуючись на попередніх даних [11].

Існує багато видів систем машинного навчання, і основні категорії базуються на двох ключових факторах:

- Типи навчання: Це залежить від типу змінної відгуку (або міток) у навчальних даних. Вони включають навчання з вчителем, навчання без вчителя, напівкероване навчання та навчання з підкріпленням.
- Суб'єктивне групування: Це групування визначається тим, що модель намагається досягти. Кожна група має схожий набір алгоритмічних підходів і принципів.

Існує багато перетинів, де алгоритми машинного навчання застосовуються до певної проблеми. Через це для однієї задачі може існувати безліч різних моделей машинного навчання. Створення найкращої моделі є мистецтвом, що вимагає багато терпіння та численних спроб і помилок.

2.2 Основні підходи в машинному навчанні

Навчання з вчителем

Клас алгоритмів машинного навчання, в якому дані включають змінну відгуку (або мітку), або ж її можна створити, називається навчанням з вчителем. Іншими словами, це набір даних, де кожен екземпляр має правильно ідентифіковану відповідь. Змінна відгуку може бути або неперервною, або категоричною. Алгоритм навчається на змінній відгуку, використовуючи заданий набір предикторних змінних. Наприклад, якщо набір даних стосується групи пацієнтів, кожен екземпляр буде містити змінну відгуку, яка вказує, чи має пацієнт рак (категорична змінна). Завдання є класифікаційним, якщо змінна відгуку категоріальна, і регресійним, якщо вона неперервна. Деякі алгоритми, призначені для регресії, можуть використовуватися для класифікації, і навпаки. Наприклад, логістичну регресію часто застосовують для класифікації, оскільки вона здатна визначати ймовірність належності до певного класу.

Навчання без вчителя

З іншого боку, коли мітки недоступні, алгоритми машинного навчання називаються навчанням без вчителя. Навчання в цьому випадку відбувається на основі певної міри схожості або відстані між кожним рядком у наборі даних. Найбільш поширеною технікою навчання без вчителя є кластеризація. Інші

методи, такі як видобуток асоціативних правил (Association Rule Mining, ARM), засновані на частоті подій.

Наприклад, є велика кількість даних про відвідувачів блогу. Можливо, захочеться запустити алгоритм кластеризації, щоб визначити групи схожих відвідувачів. Не потрібно вказувати алгоритму, до якої групи належить кожен відвідувач: він самостійно знайде ці зв'язки. Наприклад, можна виявити, що 40% відвідувачів — це чоловіки, які люблять комікси і зазвичай читають блог ввечері, тоді як 20% — це молоді любителі фантастики, які заходять на вихідних. Використовуючи ієрархічний алгоритм кластеризації, можна розбити кожну групу на менші підгрупи. Це допомагає краще таргетувати контент для кожної групи.

Напівкероване навчання

У попередніх двох типах або всі спостереження в наборі даних мають мітки, або мітки відсутні взагалі. Напівкероване навчання знаходиться між цими двома крайнощами. У багатьох практичних ситуаціях створення міток є досить витратним процесом, оскільки для цього потрібні кваліфіковані експерти. Отже, коли мітки наявні лише для невеликої частини спостережень, напівкеровані алгоритми стають оптимальним вибором для побудови моделі. Прикладом цього є деякі фото-хостинги. Після завантаження великої кількості фотографій сервіс може автоматично визначити, що одна й та сама людина присутня на різних зображеннях – це приклад неконтрольованого навчання. Потім користувачу потрібно лише додати мітки до кількох осіб, і система зможе розпізнати кожну людину на всіх фотографіях. Більшість алгоритмів напівкерованого навчання поєднують елементи як керованого, так і некерованого навчання.

Навчання з підкріпленням

Алгоритм навчання з підкріпленням (так званий агент) постійно навчається від взаємодії з навколишнім середовищем у повторюваному процесі. У ході цього процесу агент набуває досвіду через взаємодію з середовищем, поступово освоюючи всі можливі стани. Наприклад, багато роботів використовують алгоритми навчання з підкріпленням, щоб навчитися ходити, або безпілотні

автомобілі та дрони використовують навчання з підкріпленням для прийняття рішень під час руху.

Після аналізу основних типів машинного навчання, для розв'язання задачі виявлення атак у комп'ютерних мережах найбільше підходить навчання з вчителем.

2.3 Основні методи навчання з вчителем

Машина опорних векторів

Машини опорних векторів (SVM) широко застосовуються у задачах класифікації. Вони визначають два класи, знаходячи оптимальну гіперплощину, яка максимізує відстань між найближчими точками даних протилежних класів. Кількість ознак у вхідних даних визначає, чи є гіперплощина лінією у двовимірному просторі або площиною в n-вимірному просторі. Оскільки існує багато можливих гіперплощин для розрізнення класів, максимізація відстані між точками дозволяє алгоритму знайти найкращу межу між класами. Це забезпечує йому здатність добре узагальнювати нові дані та робити точні класифікаційні прогнози. Лінії, які прилягають до оптимальної гіперплощини, називаються опорними векторами, оскільки ці вектори проходять через точки даних, що визначають максимальну відстань між ними.

Алгоритм машини опорних векторів є популярним у машинному навчанні завдяки його здатності вирішувати як лінійні, так і нелінійні задачі класифікації. Якщо дані не можуть бути лінійно розділені, використовуються ядрові функції для перетворення простору даних у вищий вимірний простір, де можливе лінійне розділення. Це застосування ядрових функцій часто називається "ядровим трюком". Вибір ядрової функції, будь то лінійні ядра, поліноміальні ядра, ядра радіальної базисної функції або сигмоїдні ядра, залежить від характеристик даних та конкретного застосування [12].

Лінійні машини опорних векторів застосовуються до даних, які можуть бути лінійно розділені. Це означає, що для класифікації даних не потрібні додаткові перетворення. Границя рішення та опорні вектори утворюють аналогію вулиці, причому використовується поняття "вписування найширшої можливої

вулиці" для опису цієї задачі квадратичної оптимізації. Математично цю розділювальну гіперплощину можна представити так:

$$w * x + b = 0, \quad (2.1)$$

де w - ваговий вектор;

x - вхідний вектор;

b - член зсуву.

Існують два методи для визначення маржі, тобто максимальної відстані між класами: класифікація з жорсткою маржею та класифікація з м'якою маржею. Коли застосовується SVM з жорсткою маржею, точки даних будуть чітко відокремлені за межами опорних векторів [12]. Це описується формулою:

$$(w * x_j + b)\, y_j \geq a, \quad (2.2)$$

а потім максимізується маржа, яка представлена як:

$$max\ \gamma = a / ||w||, \quad (2.3)$$

де a - це маржа, спроектована на w.

Класифікація з м'якою маржею є більш гнучкою, оскільки допускає певну кількість помилкових класифікацій за допомогою змінних нахилу (ξ). Гіперпараметр C контролює цю гнучкість: високе значення C зменшує допустимість помилок, звужуючи маржу, тоді як низьке значення C дозволяє ширшу маржу, допускаючи більше неправильно класифікованих даних.

У більшості реальних сценаріїв дані не піддаються лінійному розділенню, і тут стають актуальними нелінійні машини опорних векторів. Щоб зробити дані лінійно роздільними, застосовуються методи попередньої обробки для перетворення їх у простір ознак вищої розмірності. Проте простори вищої розмірності можуть створювати додаткову складність, підвищуючи ризик надмірного навчання моделі та ускладнюючи обчислення. "Трюк з ядром" допомагає знизити частину цієї складності, роблячи обчислення більш ефективними шляхом заміни обчислення точкового добутку відповідною функцією ядра [12].

Існує декілька різних типів ядер, які можна використовувати для класифікації даних. Деякі з найпопулярніших функцій ядра включають:

- Поліноміальне ядро
- Ядро радіально-базисної функції (також відоме як ядро Гауса)
- Ядро сигмоїда

Логістична регресія

Логістична регресія використовується для моделювання зв'язку між змінними-предикторами та категоричною відповідною змінною. Існує три основні види логістичної регресії, що залежать від типу категоричності змінної:

- Біноміальна логістична регресія: має лише два можливих значення для залежної змінної (0 або 1). Ця модель зазвичай оцінює ймовірність того, що значення дорівнює 1, і на основі певного порогу прогнозує значення залежної змінної. Масова функція ймовірності для біноміального розподілу визначається формулою 2.4:

$$f(k;n.p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad (2.4)$$

де k - кількість успіхів;

n - загальна кількість випробувань;

p – одиниця ймовірність успіху.

- Багатофакторна логістична регресія: для категоріальної змінної відповіді існує три або більше рівнів. Зазвичай для кожного рівня обчислюється ймовірність, і на основі певного критерію класифікації (наприклад, найбільшої ймовірності) визначається значення змінної відповіді [9].

- Впорядкована логістична регресія: категоріальна змінна відповіді має певний внутрішній порядок. Цей метод схожий на мультиноміальну логістичну регресію, але з важливою відмінністю – він враховує внутрішній порядок рівнів змінної. Наприклад, змінна оцінки від 1 до 5 [9].

Випадковий ліс

Цей метод ґрунтується на поєднанні мішків та випадкового вибору ознак, що призводить до формування некорельованого лісу дерев рішень. Випадковий вибір ознак, також відомий як "пакування ознак" або "метод випадкових підпросторів", передбачає генерування випадкової підмножини ознак, що веде до

зниження кореляції між деревами рішень. Це ключова відмінність між класичними деревами рішень та випадковими лісами: перші досліджують всі можливі розбиття ознак, тоді як другі обирають лише їх підмножину.

Для налаштування алгоритму випадкових лісів перед навчанням існує три ключових гіперпараметри: розмір вузла, кількість дерев та кількість ознак, що обираються для вибірки. Після визначення цих параметрів класифікатор випадкового лісу може використовуватися для вирішення задач регресії або класифікації [13].

В основі даного алгоритму лежить множина дерев рішень. Кожне з них будується на основі вибірки даних, що випадковим чином береться з навчального набору з заміною (бутстреп-вибірка). При цьому треть даних відводиться під тестові дані, відомі як "позапакетна" вибірка. Додатковий рівень випадковості вноситься за допомогою пакування ознак. Цей метод додає різноманітності до набору даних, що зменшує кореляцію між деревами рішень.

Висновки за розділом 2

Машинне навчання широко використовується в багатьох сферах, де необхідне повторення операцій, аналіз даних та ухвалення рішень. Завдяки значному зростанню обчислювальних потужностей та накопиченню великих обсягів даних, машинне навчання стало невід'ємною частиною сучасного світу. Воно застосовується в рекомендаційних системах, розпізнаванні зображень, обробці тексту, самокерованих автомобілях, розпізнаванні спаму тощо.

Машинне навчання імітує людську інтуїцію, ухвалюючи рішення на основі минулого досвіду. Наприклад, спам-фільтри, що базуються на методах машинного навчання, аналізують слова і фрази, які часто зустрічаються в спам-повідомленнях, і таким чином ефективно відокремлюють спам від легітимних повідомлень. Ці фільтри є простішими в обслуговуванні та точнішими, ніж традиційні методи.

Машинне навчання особливо підходить для задач, де існуючі рішення вимагають великої кількості налаштувань або правил, для складних проблем, які не вирішуються традиційними методами, та для адаптації до мінливих умов.

Машинне навчання є частиною штучного інтелекту, який навчає комп'ютери ухвалювати рішення не гірше за людей.

Системи машинного навчання поділяються на контрольовані та неконтрольовані. Контрольоване навчання використовує дані з мітками (наприклад, категоричними або неперервними змінними відгуку) для створення моделей, що можуть класифікувати нові дані. Неконтрольоване навчання працює без міток, шукаючи закономірності або групи в даних (наприклад, кластеризацію). Існують також напівкероване навчання, яке поєднує елементи обох попередніх підходів, та навчання з підкріпленням, де агент навчається через взаємодію з навколишнім середовищем.

Машинне навчання має безліч застосувань, і його розвиток продовжуватиметься, забезпечуючи автоматизовані та інтелектуальні рішення для складних завдань.

РОЗДІЛ 3
РОЗРОБКА КОМП'ЮТЕРНОЇ МОДЕЛІ

3.1 Розвідувальний аналіз

Розвідувальний аналіз даних – це набір методів і технік, які використовуються для вивчення та розуміння структури, властивостей і моделей даних перед застосуванням більш формальних статистичних моделей або моделей машинного навчання. Мета полягає в тому, щоб визначити основні характеристики даних і вирішити, які моделі та методи аналізу найбільш підходять для подальшої роботи.

Ключові компоненти аналізу включають:

- Огляд даних, тобто початкове вивчення даних, щоб дізнатися їх первинну структуру та зміст, наприклад розмірність даних, типи змінних і кількість відсутніх значень;
- Візуалізація даних, яка передбачає використання різних типів графіків і діаграм для візуального представлення даних і виявлення прихованих закономірностей, тенденцій і аномалій. Найбільш популярними видами візуалізації є гістограми, діаграми розсіювання, коробкові діаграми та теплові карти;
- Описова статистика: обчислення основних статистичних даних, таких як середнє значення, медіана, мода, стандартне відхилення, квартилі та коефіцієнт кореляції для різних змінних у заданому наборі даних;
- Виявлення аномалій, що означає ідентифікацію та вивчення аномальних або нетипових значень, які можуть показати або помилки в зборі даних, або деякі цікаві явища, які потребують подальшого дослідження;

Розвідувальний аналіз є важливим етапом будь-якого аналітичного проекту, оскільки він закладає основу для визначення того, які подальші дії та методи необхідні для досягнення цілей аналізу.

3.1.1 Опис даних

CICIDS2017 - це набір даних, що містить записи про доброякісну та злоякісну мережеву активність, наближену до реальних умов. Він складається з результатів аналізу мережевого трафіку, проведеного за допомогою CICFlowMeter. Дані представлені у CSV-файлах, де кожен запис містить інформацію про часову мітку, IP-адреси джерела та призначення, порти джерела та призначення, протоколи та тип атаки. На основі даного набору даних було змодельовано 25 моделей поведінки користувачів, що ґрунтуються на використанні протоколів HTTP, HTTPS, FTP, SSH та електронної пошти. До складу набору даних включені найпоширеніші типи атак, зафіксовані у звіті McAfee за 2016 рік. До них належать веб-атаки, перебір паролів, DoS, DDoS, Heartbleed, бот-атаки та сканування мережі. Детальні описи цих атак можна знайти в документації до набору даних [2].

Короткий опис змінних датасету:

1. Flow Duration: Час тривалості кожного потоку. Можна аналізувати середню, максимальну та мінімальну тривалість.
2. Packet Counts (Total Fwd Packets, Total Backward Packets): Кількість пакетів у кожному напрямку.
3. Packet Lengths (Fwd Packet Length Max/Min/Mean/Std): Статистика щодо довжини пакетів у передньому напрямку. Це може вказувати на типи трафіку (наприклад, HTTP, FTP, тощо).
4. Flow Bytes/s, Flow Packets/s: Швидкість передачі даних та пакетів.
5. Flag Counts (SYN, ACK, FIN, PSH, RST тощо): Кількість пакетів з певними прапорцями. Це може вказувати на типи комунікації, такі як встановлення з'єднання, передача даних, завершення з'єднання тощо.
6. Active/Idle Times: Середні, максимальні та мінімальні часи активності та простою.
7. Label: Мітка або клас, до якого належить кожен запис.
8. Flow IAT (Inter Arrival Time) Mean/Std/Max/Min): Статистика про час між послідовними пакетами в кожному потоці.

9. Fwd/Bwd IAT Total/Mean/Std/Max/Min: Аналогічно Flow IAT, але розбито на передній (від клієнта до сервера) та зворотній (від сервера до клієнта) напрямки.

10. PSH/URG Flags: Кількість пакетів з встановленими флагами PSH (Push) або URG (Urgent).

11. Down/Up Ratio: Відношення кількості пакетів у напрямку вниз/у напрямку вгору. Це може вказати на типи трафіку: велику кількість пакетів у напрямку вниз може свідчити про великий потік завантаження даних.

12. Average Packet Size, Avg Fwd/Bwd Segment Size: Середній розмір пакета або сегменту у передньому та зворотньому напрямках. Це може дати інформацію про типи даних, які передаються.

13. Init_Win_bytes_forward/ backward: Розмір вікна TCP у вихідному та зворотньому напрямках. Це може бути важливим для аналізу пропускної спроможності мережі та управління витратами на мережу.

3.1.2 Описова статистика

Аналіз базується на наборі даних, що складається з 225745 записів мережевого трафіку. Набір даних включає кілька ключових характеристик, які описують властивості кожного з'єднання в мережі.

Кількісні показники можна побачити на табл. 3.1:

Таблиця 3.1

Таблиця основних кількісних показників набору даних

Змінна	Мінімум	Максимум	Середнє значення	Стандартне відхилення
Destination Port	0	65532	8879,62	19754,65
Total Fwd Packets	1	1932	4,87	15,42
Total Backward Packets	0	2942	4,57	21,76
Total Length of Fwd Packets	0	183012	939,46	3249,4
Total Length of Bwd Packets	0	5172346	5960,48	39218,34
Fwd Packet Length Max	0	11680	538,54	1864,13
Fwd Packet Length Min	0	3867	388,23	1490,32
Bwd Packet Length Max	0	1460	26,25	125,14
Bwd Packet Length Min	0	667	1,37	14,84
Flow Bytes/s	0	99860000	161600	1089000
Flow Packets/s	0	222000	57,89	676,47
Flow IAT Mean	0	77200000	358800	1440000
Fwd IAT Total	0	77200000	358900	1440000
Bwd IAT Total	0	77200000	358800	1440000

Розподіл категоріальних змінних:

Label: Показник, що вказує на тип трафіку

- Безпечний трафік (Label = 0): 97718 записів (43.3%)
- DDoS трафік (Label = 1): 128027 записів (56.7%)

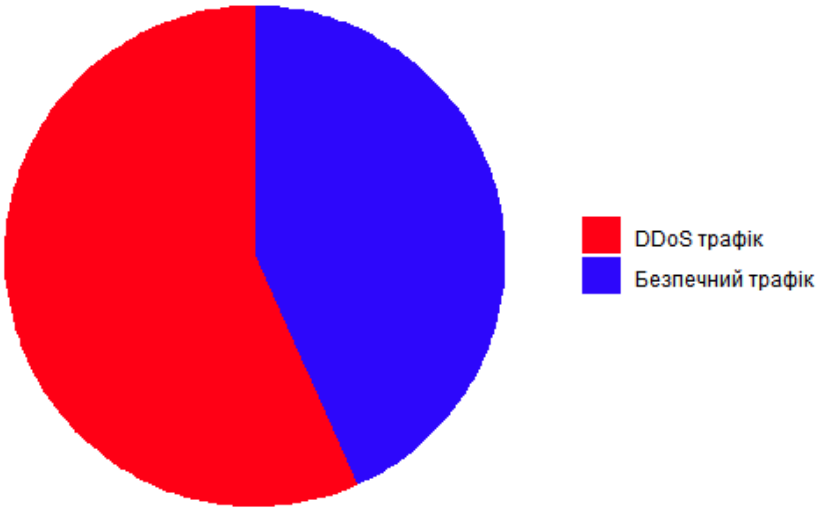


Рисунок 3.1 – Розподіл трафіку за змінною Label

3.1.3 Візуалізація даних

Візуальний аналіз даних є фундаментальною частиною нових підходів до науки про дані та аналітики. Він допомагає дослідникам та аналітикам вивчати більше інформації за менший час, ніж за допомогою традиційних методів. Візуалізація даних за допомогою різноманітних графічних представлень допомагає виявити приховані закономірності, аномалії та тенденції, які в іншому випадку відносно складно виявити, використовуючи лише табличні або текстові формати.

Основні принципи візуального аналізу даних полягають у виборі правильного типу графіків і діаграм, які відображають або відображають реальну природу і строгість даних. До них відносяться такі інструменти, як гістограми, секторні діаграми та теплові карти для оцінки розподілу даних і виявлення статистичних характеристик. Діаграми розсіювання та лінійні графіки можна використовувати для обговорення взаємозв'язку між змінними, а також динаміки змін.

Хороша візуалізація даних також враховує те, як люди сприймають речі, маючи можливість ефективно використовувати колір, форму і розмір, щоб привернути увагу глядачів саме там, де це потрібно. Нарешті, одним з найважливіших аспектів, про який варто згадати, є психологічні аспекти, пов'язані зі сприйняттям. Правильне використання візуальних елементів має надзвичайно позитивний вплив на чіткість та зрозумілість інформації, включеної в графіки з точки зору людини (лінійні діаграми).

Таким чином, візуальний аналіз даних є безцінним компонентом, який включає наукові методи і технології, доповнені творчими відкриттями, для отримання нових знань у процесі прийняття рішень у багатьох галузях науки і промисловості.

Вибір змінних для візуалізації даних мережевого трафіку є важливим кроком у процесі аналізу, який дозволяє виявити ключові характеристики та патерни, що можуть свідчити про наявність аномалій або загроз. У цьому дослідженні для візуалізації обрані наступні змінні: "Flow Duration", "Total Fwd Packets", "Total Backward Packets", "Total Length of Fwd Packets", "Total Length of Bwd Packets", "Fwd Packet Length Max", "Bwd Packet Length Max", "Fwd IAT Mean", "Bwd IAT Mean", "Label". Вибір цих змінних обґрунтований їхньою значимістю для розуміння поведінки мережевого трафіку та виявлення потенційних DDoS атак.

Обґрунтування вибору змінних

1. Flow.Duration - Тривалість потоку в мікросекундах.

Ця змінна дозволяє визначити тривалість з'єднань у мережі, що може бути індикатором аномальної активності. Короткотривалі з'єднання часто асоціюються з DDoS атаками.

2. Total.Fwd.Packets - Кількість пакетів, що були надіслані з ініціатора з'єднання.

Кількість пакетів, надісланих вперед, може відображати інтенсивність трафіку, що є ключовим фактором при аналізі DDoS атак, які часто характеризуються великою кількістю пакетів за короткий період часу.

3. Total.Backward.Packets - Кількість пакетів, що були надіслані у відповідь.

Ця змінна дозволяє оцінити реакцію системи на вхідний трафік. Висока кількість відповідних пакетів може свідчити про спроби обробити великий обсяг запитів, характерних для DDoS атак.

4. Total.Length.of.Fwd.Packets - Сумарна довжина всіх пакетів, надісланих з ініціатора з'єднання.

Аналіз загальної довжини пакетів допомагає визначити обсяг переданих даних, що може бути важливим показником для виявлення атак, спрямованих на перевантаження каналу зв'язку.

5. Total.Length.of.Bwd.Packets - Сумарна довжина всіх пакетів, надісланих у відповідь.

Ця змінна дозволяє оцінити загальний обсяг даних, що повертаються у відповідь на запити, що також може свідчити про інтенсивність обробки запитів системою.

6. Fwd.Packet.Length.Max - Максимальна довжина пакету, надісланого з ініціатора з'єднання.

Максимальна довжина пакету може бути показником для ідентифікації великих даних, що передаються у мережі, та виявлення можливих аномалій.

7. Bwd.Packet.Length.Max - Максимальна довжина пакету, надісланого у відповідь.

Аналіз цієї змінної допомагає виявити максимальний обсяг даних, що можуть бути передані у відповідь, що є важливим для оцінки здатності системи обробляти великі запити.

8. Fwd.IAT.Mean - Середній час між надсиланням пакетів з ініціатора з'єднання

Ця змінна дозволяє аналізувати частоту надсилання пакетів, що може бути індикатором аномальної активності, такої як DDoS атаки, які часто характеризуються високою частотою пакетів.

9. Bwd.IAT.Mean - Середній час між надсиланням пакетів у відповідь.

Аналіз інтервалів між відповідними пакетами дозволяє оцінити швидкість обробки запитів системою, що також може свідчити про наявність аномалій.

10. Label – Мітка.

Змінна "Label" є ключовою для класифікації та порівняння характеристик безпечного та DDoS трафіку, що дозволяє визначити патерни, характерні для кожного типу трафіку.

Вибір зазначених змінних базується на їхній здатності забезпечити глибоке розуміння характеристик мережевого трафіку, що є критично важливим для виявлення аномалій та загроз. Кожна з обраних змінних вносить значний вклад у процес аналізу та дозволяє ідентифікувати ключові відмінності між безпечним та DDoS трафіком, що є необхідним для побудови ефективних систем моніторингу та захисту мережі.

Проведемо для обраних даних візуалізацію та аналіз:

Графік на рисунку 3.2 зображає розподіл тривалості потоку (Flow.Duration) для двох категорій мережевого трафіку: безпечний трафік (позначений синім кольором) та DDoS трафік (позначений червоним кольором). Тривалість потоку вимірюється у мікросекундах по горизонтальній осі, а кількість потоків відображена по вертикальній осі.

Короткотривалі потоки (до 1e+07 мікросекунд) складають найбільшу частку для обох категорій, але їх значно більше серед DDoS трафіку. Це може бути пов'язано з тим, що DDoS атаки часто включають велику кількість короткотривалих з'єднань для створення перевантаження на цільовій системі.

Потоки з тривалістю понад 4e+07 мікросекунд спостерігаються рідше, однак їх наявність як у безпечному, так і в DDoS трафіку може вказувати на складніші сценарії взаємодії або атаки, що імітують легітимний трафік.

Порівняння тривалості потоків між безпечним і DDoS трафіком демонструє значну різницю в кількості короткотривалих з'єднань. Це може бути використано як критерій для автоматичного виявлення DDoS атак у реальних мережах.

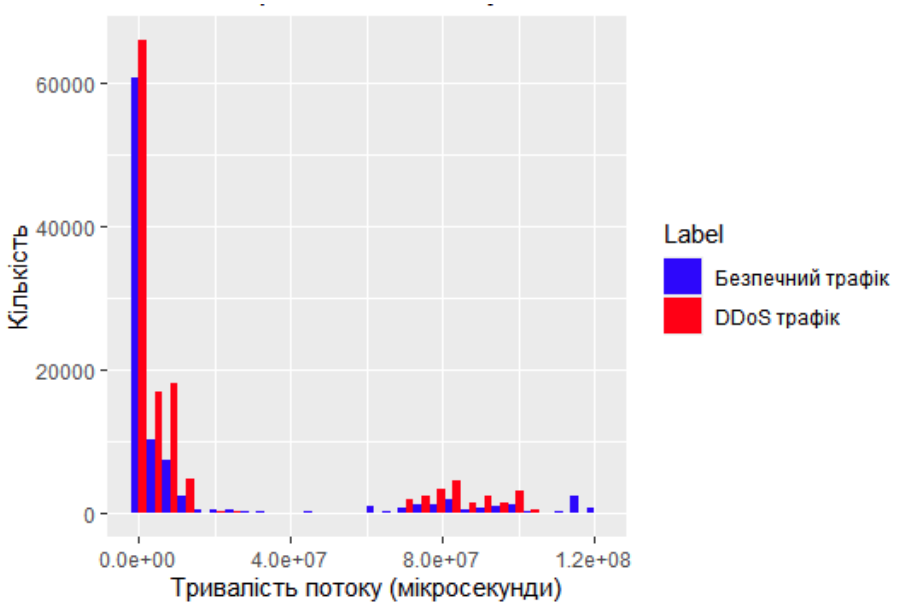


Рисунок 3.2 – Розподіл тривалості потоку(Flow.Duration) за змінною Label

Гістограма на рисунку 3.3 зображає розподіл кількості отриманих пакетів (Total.Fwd.Packets), для двох категорій мережевого трафіку: безпечний трафік (позначений синім кольором) та DDoS трафік (позначений червоним кольором). Кількість пакетів показана на горизонтальній осі у логарифмічній шкалі, а частота потоків — на вертикальній осі.

DDoS трафік переважає у діапазоні малої кількості пакетів (1-10), що може бути пов'язано з природою DDoS атак, де генерується велика кількість коротких запитів з метою перевантаження цільової системи.

Безпечний трафік демонструє ширший діапазон кількості пакетів, особливо у діапазоні понад 10 пакетів. Це свідчить про різноманітність легітимних з'єднань, які можуть включати передачу великих обсягів даних, таких як завантаження файлів або стримінг медіа.

Використання логарифмічної шкали для осі x дозволяє краще відобразити розподіл даних, які варіюються в широкому діапазоні, і робить графік більш читабельним. Це особливо корисно для виявлення відмінностей у розподілі кількості пакетів між двома категоріями трафіку.

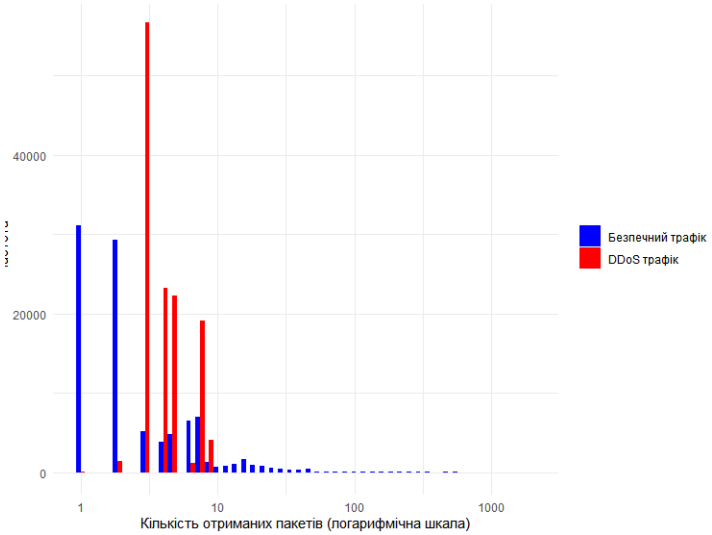


Рисунок 3.3 – Розподіл загальної кількості отриманих пакетів (Total.Fwd.Packets) за змінною Label

Гістограма на рисунку 3.4 зображає розподіл кількості надісланих у відповідь пакетів (Total.Bwd.Packets), для двох категорій мережевого трафіку: безпечний трафік (позначений синім кольором) та DDoS трафік (позначений червоним кольором). Кількість пакетів показана на горизонтальній осі у логарифмічній шкалі, а частота потоків — на вертикальній осі.

Як і у випадку з Total.Fwd.Packets, DDoS трафік переважає у діапазоні малої кількості пакетів (1-10), що може бути пов'язано з природою DDoS атак. Також є схожість і у безпечного трафіка, який має ширший діапазон кількості пакетів, особливо у діапазоні понад 10 пакетів.

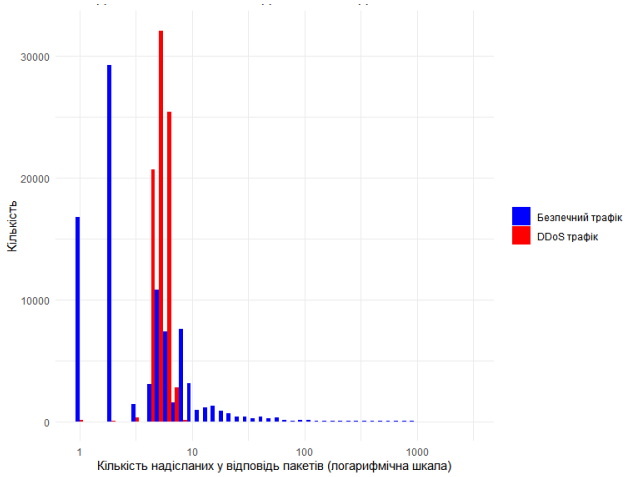


Рисунок 3.4 – Розподіл загальної кількості надісланих пакетів у відповідь (Total.Backward.Packets) за змінною Label

Для загального розподілу довжини надісланих пакетів зберігається тенденція високої щільності коротких з'єднань у DDoS трафіку спрямованих на створення перевантаження системи. А безпечний трафік так само має ширший діапазон значень.

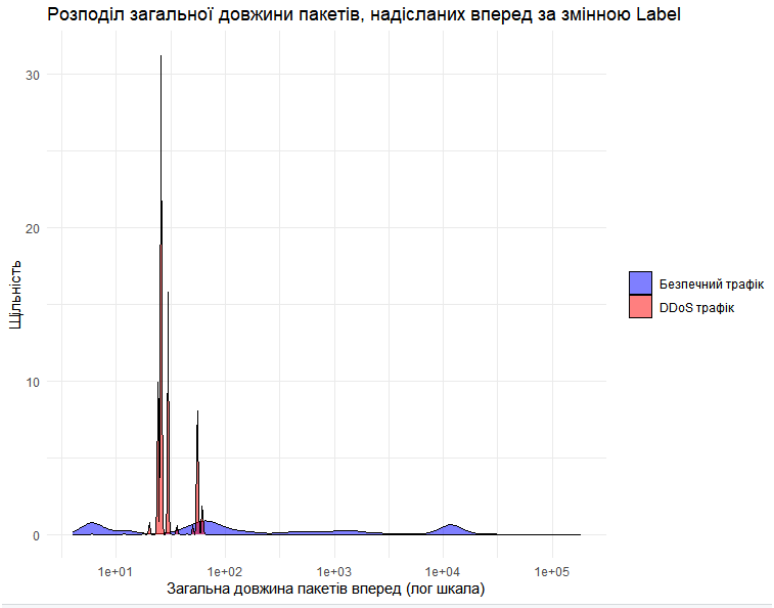


Рисунок 3.5 Розподіл загальної довжини отриманих пакетів (Total.Length.of.Fwd.Packets) за змінною Label

Графік розсіювання на рисунку 3.6 зображає розподіл загальної довжини надісланих у відповідь пакетів (Total.Length.of.Bwd.Packets), для двох категорій мережевого трафіку: безпечний трафік (позначений синім кольором) та DDoS трафік (позначений червоним кольором). Загальна довжина пакетів відображена на вертикальній осі у логарифмічній шкалі, а категорія трафіку (Label) — на горизонтальній осі.

Безпечний трафік демонструє значну мінливість: довжина пакетів розкидана по всьому діапазону осі Y. Є кілька горизонтальних смуг, де точки розташовані щільніше, що свідчить про загальні значення довжини пакета. Така мінливість передбачає, що безпечний трафік охоплює різні розміри пакетів, що може бути пов'язане із звичайними коливаннями активності мережі.

DDoS-трафік характеризується більш регулярним та передбачуваним характером із меншою мінливістю довжини пакета. Точки даних для DDoS-

трафіку в основному зосереджені в окремих смугах, що вказує на певну довжину пакетів, що повторюється. Ця закономірність передбачає, що в DDoS-атаках часто використовуються пакети однакового розміру, можливо, через автоматичні сценарії атак, що генерують однорідний трафік.

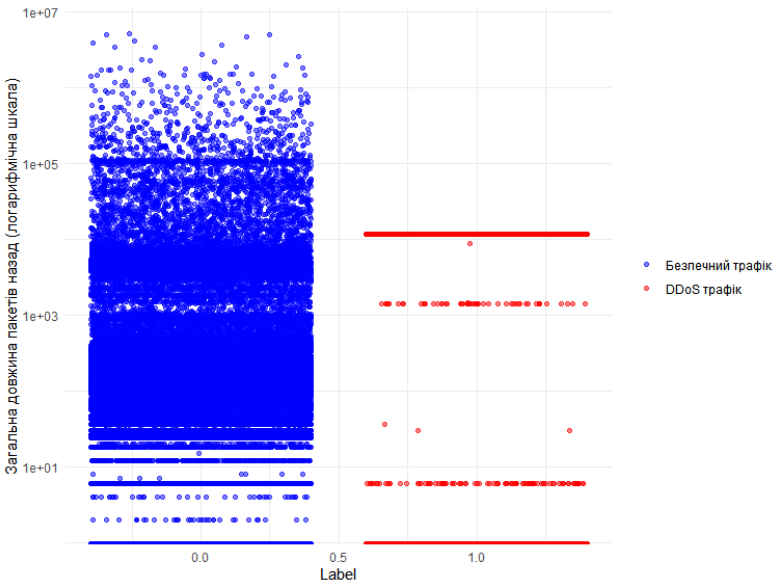


Рисунок 3.6 Розподіл загальної довжини надісланих у відповідь пакетів (Total.Length.of.Bwd.Packets) за змінною Label

За графіком розсіювання з рисунку 3.7 можна зробити висновки, що безпечний трафік має широкий і рівномірний розподіл загальної довжини пакетів, тоді як DDoS трафік характеризується скупченнями на певних рівнях, що може вказувати на типові патерни атак.

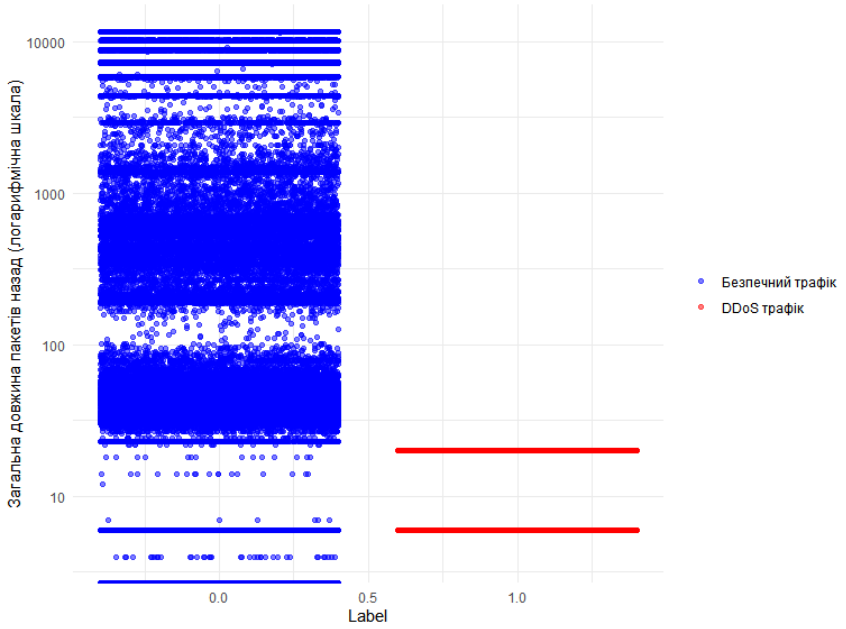


Рисунок 3.7 Розподіл максимальної довжини отриманого пакету (Fwd.Packet.Length.Max) за змінною Label

На рисунку 3.8 приведено графік змінної Bwd.Packet.Length.Max. Безпечний трафік характеризується високою концентрацією потоків з мінімальною довжиною пакету, тоді як DDoS трафік має чіткі піки на рівнях близько 4000, 6000, 8000, та 12000 байтів, що свідчить про специфічні патерни трафіку, характерні для атак.

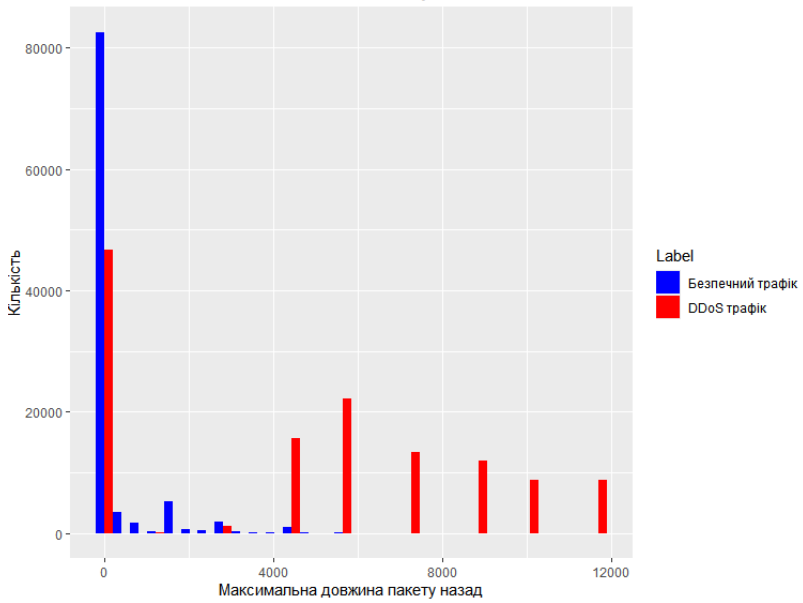


Рисунок 3.8 Розподіл максимальної довжини надісланих у відповідь пакетів (Bwd.Packet.Length.Max) за змінною Label

Рисунок 3.9 зображує графік змінної Fwd.IAT.Mean. Розподіл безпечного трафіку демонструє декілька піків щільності, що вказують на наявність різних типів легітимного трафіку з різними інтервалами між пакетами. Найвищі піки щільності спостерігаються в діапазонах близько 1e+02, 1e+04, та 1e+06 мікросекунд.

Розподіл DDoS трафіку також показує кілька піків щільності, з найвищими значеннями на рівнях приблизно 1e+02, 1e+04, та 1e+06 мікросекунд. Піки щільності для DDoS трафіку є більш вираженими та вузькими, що свідчить про більш одноманітний характер атак.

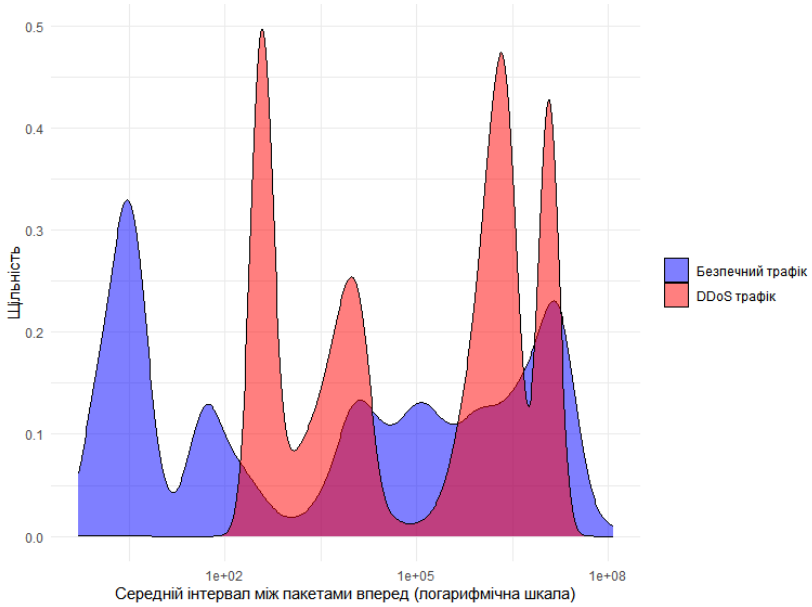


Рисунок 3.9 Розподіл середнього інтервалу між отриманими пакетами (Fwd.IAT.Mean) за змінною Label

На графіку для Bwd.IAT.Mean маємо приблизно схожу ситуацію, безпечний трафік має ширший діапазон та рівномірне розподілення, DDoS трафік демонструє чіткіші та вужчі піки, що може бути характерною ознакою атак з постійними інтервалами між пакетами.

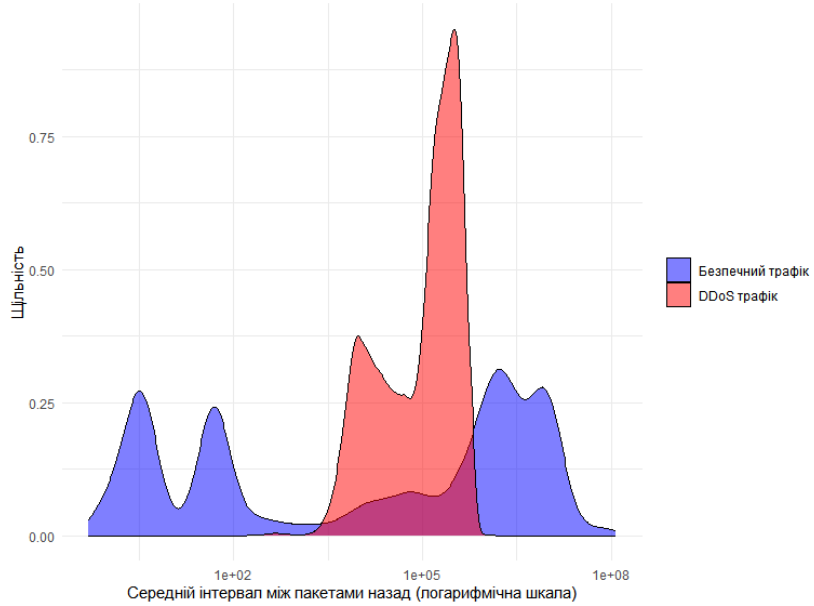


Рисунок 3.10 Розподіл середнього інтервалу між пакетами надісланими у відповідь (Bwd.IAT.Mean) за змінною Label

Отже, аналіз розподілу різних характеристик мережевого трафіку виявив суттєві відмінності між безпечним та DDoS трафіком. Безпечний трафік демонструє широкий і різноманітний розподіл значень, що відображає різні типи легітимних з'єднань. У той час, як DDoS трафік характеризується чіткими піками та вузькими розподілами на певних рівнях, що може бути ознакою атак.

Кореляційна матриця – це таблиця, яка показує коефіцієнти кореляції між змінними в наборі даних. Коефіцієнт кореляції варіюється від -1 до 1 і визначає напрямок та силу зв'язку між змінними:

- **«1»** означає ідеальну позитивну кореляцію (якщо одна змінна зростає, інша також зростає).
- **«-1»** означає ідеальну негативну кореляцію (якщо одна змінна зростає, інша зменшується).
- **«0»** означає відсутність кореляції.

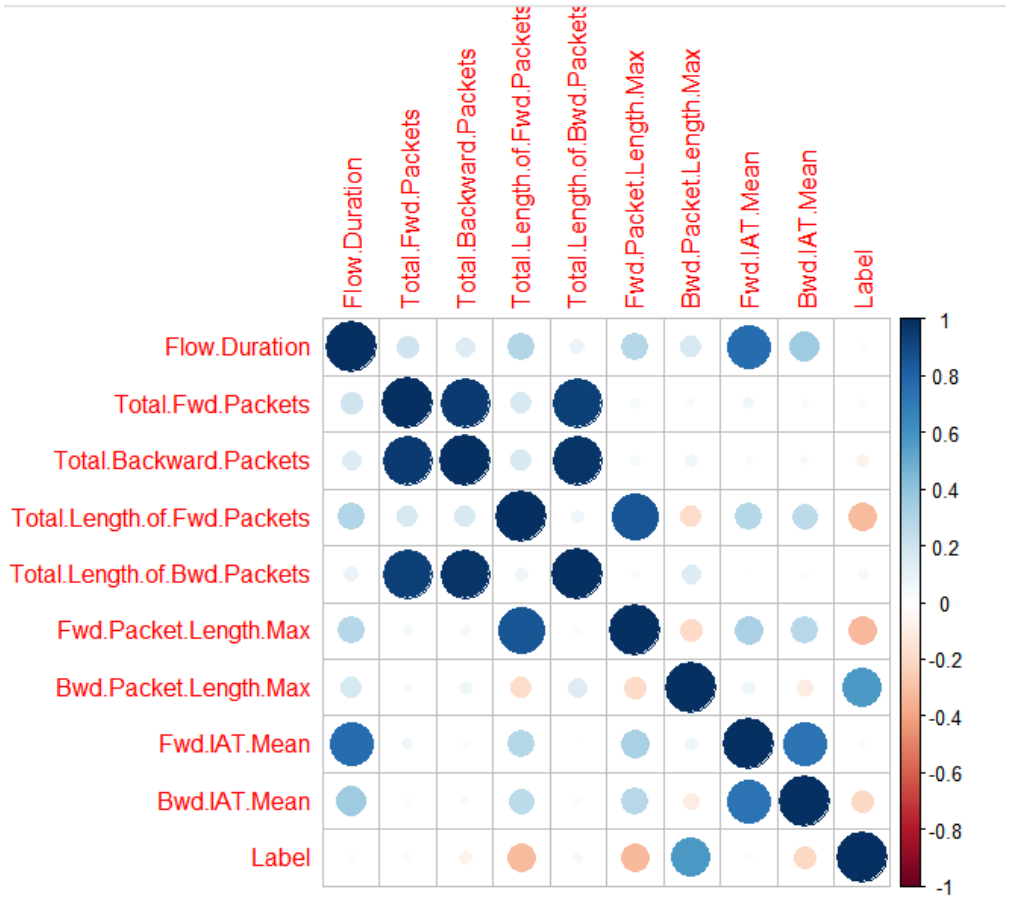


Рисунок 3.11 Кореляційна матриця

За кореляційною матрицею обрано такі змінні для побудови моделі: Bwd.Packet.Length.Max, Fwd.Packet.Length.Max, Total.Length.of.Fwd.Packets.

3.2 Побудова моделі за допомогою логістичної регресії

Метою попереднього аналізу було побудувати модель логістичної регресії для виявлення DDoS атак на основі мережевого трафіку. Було використано змінні, що включають максимальну довжину отриманого і надісланого у відповідь пакету, а також загальну довжину отриманих пакетів.

Для забезпечення відтворюваності результатів було встановлено фіксований початковий стан генератора випадкових чисел. Далі дані були розділені на навчальну та тестову вибірки, де 75% даних використовувалися для навчання моделі, а 25% — для тестування.

Було побудовано модель логістичної регресії з використанням мітки (Label) та трьох незалежних змінних: максимальна довжина отриманого пакету(Fwd.Packet.Length.Max), максимальна довжина надісланого у відповідь

пакету(Bwd.Packet.Length.Max) і загальна довжина отриманих пакетів(Total.Length.of.Fwd.Packets). Модель була навчена на навчальній вибірці.

Отримані передбачення були класифіковані у дві категорії: 1 (DDoS трафік) та 0 (безпечний трафік), з використанням порогу ймовірності 0.5.

Було проведено оцінку коефіцієнтів моделі, звіт можна побачити на рисунку 3.12, за допомогою матриці невідповідностей, яка включає такі показники, як точність, чутливість, специфічність, каппа-коефіцієнт та інші метрики. Для оцінки здатності моделі розрізняти класи було розраховано показник AUC (Area Under Curve) з використанням об'єкта передбачень та функцій з пакету ROCR.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 17170    10
         1  7259 31996

               Accuracy : 0.8712
                 95% CI : (0.8684, 0.874)
    No Information Rate : 0.5671
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7281

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7029
            Specificity : 0.9997
         Pos Pred Value : 0.9994
         Neg Pred Value : 0.8151
             Prevalence : 0.4329
         Detection Rate : 0.3042
   Detection Prevalence : 0.3044
      Balanced Accuracy : 0.8513

       'Positive' Class : 0
```

Рисунок 3.12 Матриця невідповідностей моделі логістичної регресії

Точність вказує на частку правильних передбачень серед усіх передбачень. В даному випадку модель правильно класифікувала 87.12% зразків.

Чутливість відображає здатність моделі правильно ідентифікувати негативні випадки (DDoS трафік). Чутливість моделі становить 70.29%.

Специфічність відображає здатність моделі правильно виявляти позитивні випадки (безпечний трафік). Модель виявила 99.97% таких випадків.

Збалансована точність є середнім арифметичним чутливості та специфічності, що дозволяє враховувати дисбаланс класів. В даному випадку цей показник становить 85.13%.

Показник AUC відображає здатність моделі розрізняти класи. Значення 0.9788549 вказує на високу дискримінаційну здатність моделі, що є дуже хорошим результатом.

Модель логістичної регресії показала непогану ефективність у виявленні DDoS атак, але можна добитись кращих показників. Загалом ці результати вказують на те, що модель може бути успішно застосована для реального моніторингу мережевого трафіку та виявлення потенційних атак.

Для покращення результатів виявлення DDoS атак на основі мережевого трафіку була розроблена нова модель логістичної регресії з використанням усіх змінних датасету, окрім тих, що мають високу кореляцію.

Дані були розділені на навчальну та тестову вибірки у пропорції 75% до 25%, забезпечуючи відтворюваність результатів за допомогою фіксованого початкового стану генератора випадкових чисел. Кореляційний аналіз був проведений для вибору релевантних змінних. Змінні з високим рівнем кореляції (понад 0.9) були видалені для уникнення мультиколінеарності. Модель логістичної регресії була побудована з використанням крос-валідації (метод 5-кратної крос-валідації) для забезпечення надійності результатів. Результати наведені на рисунку 3.13.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 23708    30
         1   721 31976

               Accuracy : 0.9867
                 95% CI : (0.9857, 0.9876)
    No Information Rate : 0.5671
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9728

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9705
            Specificity : 0.9991
         Pos Pred Value : 0.9987
         Neg Pred Value : 0.9779
             Prevalence : 0.4329
         Detection Rate : 0.4201
   Detection Prevalence : 0.4206
      Balanced Accuracy : 0.9848

       'Positive' Class : 0
```

Рисунок 3.13 Матриця невідповідностей

Модель правильно класифікувала 98.73% зразків.

Чутливість відображає здатність моделі правильно виявляти негативні випадки (DDoS трафік). Модель виявила 97.17% таких випадків.

Специфічність відображає здатність моделі правильно ідентифікувати позитивні випадки (безпечний трафік). Специфічність моделі становить 99.92%.

Збалансована точність є середнім арифметичним чутливості та специфічності, що дозволяє враховувати дисбаланс класів. В даному випадку цей показник становить 98.55%.

Показник позитивної прогностичної цінності (Pos Pred Value): 0.9989. Висока частка правильно виявлених випадків безпечного трафіку серед усіх випадків, класифікованих як безпечний трафік.

Показник негативної прогностичної цінності (Neg Pred Value): 0.9788. Висока частка правильно виявлених випадків DDoS трафіку серед усіх випадків, класифікованих як DDoS трафік.

Покращена модель логістичної регресії показала значно кращі результати у виявленні DDoS атак, демонструючи високу точність, чутливість та

специфічність. Високе значення каппа-коефіцієнта та збалансованої точності підтверджують надійність і узгодженість моделі.

3.3 Побудова моделі за допомогою випадкових лісів

Для більш обширного розуміння проблеми виявлення DDoS атак на основі мережевого трафіку була розроблена модель випадкових лісів. Було використано змінні, що включають максимальну довжину отриманого і надісланого у відповідь пакету, а також загальну довжину отриманих пакетів..

Для забезпечення відтворюваності результатів було встановлено фіксований початковий стан генератора випадкових чисел. Дані були розділені на навчальну та тестову вибірки у пропорції 75% до 25%.

Модель випадкових лісів була побудована з використанням мітки (Label) та трьох незалежних змінних: максимальна довжина отриманого пакету(Fwd.Packet.Length.Max), максимальна довжина надісланого у відповідь пакету(Bwd.Packet.Length.Max) і загальна довжина отриманих пакетів(Total.Length.of.Fwd.Packets). Було використано 500 дерев (ntree = 500) та 3 змінних для поділу на кожному вузлі (mtry = 3). Важливість змінних також була врахована під час навчання моделі.

Оцінка Моделі: Було проведено оцінку моделі за допомогою матриці суміщень, яка включає такі показники, як точність, чутливість, специфічність, каппа-коефіцієнт та інші метрики, які представлені на рисунку 3.14.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 24417    60
         1    12 31946

               Accuracy : 0.9987
                 95% CI : (0.9984, 0.999)
    No Information Rate : 0.5671
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9974

 Mcnemar's Test P-Value : 3.042e-08

            Sensitivity : 0.9995
            Specificity : 0.9981
         Pos Pred Value : 0.9975
         Neg Pred Value : 0.9996
             Prevalence : 0.4329
         Detection Rate : 0.4327
   Detection Prevalence : 0.4337
      Balanced Accuracy : 0.9988

       'Positive' Class : 0
```

Рисунок 3.14 Матриця невідповідностей для моделі з випадковими лісами

Модель правильно класифікувала 99.87% зразків.

Чутливість відображає здатність моделі правильно виявляти позитивні випадки (безпечний трафік). Модель виявила 99.95% таких випадків.

Специфічність відображає здатність моделі правильно ідентифікувати негативні випадки (DDoS трафік). Специфічність моделі становить 99.81%.

Збалансована точність є середнім арифметичним чутливості та специфічності, що дозволяє враховувати дисбаланс класів. В даному випадку цей показник становить 99.88%.

Інші Метрики:

Показник позитивної прогностичної цінності (Pos Pred Value): 0.9975. Висока частка правильно виявлених випадків безпечного трафіку серед усіх випадків, класифікованих як безпечні.

Показник Негативної Прогностичної Цінності (Neg Pred Value): 0.9996. Висока частка правильно виявлених випадків DDoS трафіку серед усіх випадків, класифікованих як негативні.

Модель випадкових лісів показала відмінні результати у виявленні DDoS атак, демонструючи гарні результати за всіма показниками. Ці результати вказують на те, що модель випадкових лісів є ефективним інструментом для реального моніторингу мережевого трафіку та виявлення потенційних атак.

Висновки за розділом 3

Для виявлення DDoS атак на основі мережевого трафіку були розроблені та оцінені три моделі: логістична регресія, покращена логістична регресія та модель випадкових лісів. Кожна модель пройшла тестування з використанням різних метрик оцінки, таких як точність, чутливість, специфічність та інші показники узгодженості, які можна побачити у табл. 3.2 та табл. 3.3.

Таблиця 3.2

Порівняльна таблиця отриманих результатів

	Точність	Чутливість	Специвічність
Логістична регресія	0.8712	0.7029	0.9997
Покращена логістична регресія	0.9867	0.9705	0.9991
Випадковий ліс	0.9987	0.9995	0.9981

Таблиця 3.3

Порівняльна таблиця отриманих результатів

	Збалансована точність	Pos Pred Value	Neg Pred Value
Логістична регресія	0.8513	0.9994	0.8151
Покращена логістична регресія	0.9848	0.9987	0.9779
Випадковий ліс	0.9988	0.9975	0.9996

Логістична регресія

Логістична регресія є базовою моделлю, яка показала добрі результати у виявленні DDoS атак. Вона демонструє високий рівень точності, здатність правильно ідентифікувати значну частину атак і легітимного трафіку. Ця модель має добру узгодженість з реальними даними, хоча й поступається за деякими показниками іншим моделям.

Покращена логістична регресія

Покращена логістична регресія включає додаткову обробку кореляційної матриці з видаленням змінних з високою кореляцією. Ця модель показує ще вищу точність та більш надійну здатність виявляти як DDoS атаки, так і легітимний трафік. Покращена логістична регресія демонструє високу узгодженість і може бути ефективним інструментом для аналізу мережевого трафіку.

Випадковий ліс

Модель випадкових лісів показала найкращі результати серед усіх трьох моделей. Вона відзначається надзвичайно високою точністю, чудовою здатністю ідентифікувати атаки, а також відмінною узгодженістю з реальними даними. Модель випадкових лісів демонструє високу ефективність у виявленні аномалій у мережевому трафіку, що робить її найкращим вибором для реального моніторингу та виявлення DDoS атак.

На основі проведеного аналізу рекомендується використовувати модель випадкових лісів для реального моніторингу мережевого трафіку та ефективного виявлення потенційних DDoS атак, оскільки ця модель демонструє найкращі результати за всіма ключовими показниками.

ВИСНОВКИ

У даній роботі проведено детальний аналіз методів виявлення атак у комп'ютерних мережах з використанням машинного навчання. В результаті дослідження розроблено та оцінено кілька моделей, включаючи логістичну регресію, покращену логістичну регресію та модель випадкових лісів. Кожна з моделей продемонструвала високий рівень точності та ефективності у виявленні DDoS атак, проте модель випадкових лісів показала найкращі результати за всіма ключовими показниками.

Основні результати дослідження включають:

- Розробку та впровадження моделей машинного навчання для виявлення атак на основі аналізу мережевого трафіку.
- Виявлення ключових характеристик, що дозволяють ефективно відрізняти безпечний трафік від атакуючого, таких як тривалість потоку, кількість пакетів, загальна довжина пакетів та інтервали між ними.
- Використання методів візуалізації даних для виявлення патернів та аномалій у мережевому трафіку.

Модель випадкових лісів виявилася найефективнішою для реального моніторингу мережевого трафіку та виявлення потенційних атак, демонструючи надзвичайно високу точність, чудову здатність ідентифікувати атаки та відмінну узгодженість з реальними даними. Це робить її найкращим вибором для практичного застосування в області кібербезпеки.

Успішна реалізація даного проекту сприятиме підвищенню рівня безпеки комп'ютерних мереж, забезпечуючи своєчасне виявлення та блокування потенційних кіберзагроз. Отримані результати можуть бути використані для подальших досліджень та розробок у сфері захисту інформаційних систем, а також для впровадження у реальних умовах з метою покращення захисту від кіберзагроз.

Робота підкреслює важливість використання сучасних методів машинного навчання для виявлення та запобігання атакам у комп'ютерних мережах, що є

критично важливим у сучасному цифровому світі. Подальший розвиток технологій у цій сфері відкриває нові можливості для створення більш безпечних та надійних комп'ютерних систем та мереж, що є необхідним для забезпечення стабільного функціонування сучасного суспільства.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2016. 758 p.
2. CICIDS2017 [Електронний ресурс]. - режим доступу: URL: https://www.kaggle.com/datasets/cicdataset/cicids2017/data (дата звернення - 20.02.2024).
3. Cyber-attack detection in network traffic using machine learning [Електронний ресурс]. - режим доступу: URL: https://repository.rit.edu/cgi/viewcontent.cgi?article=12453&context=theses (дата звернення - 17.03.2024).
4. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Incorporated, 2022.
5. Kubat M. Introduction to Machine Learning. Springer International Publishing AG, 2021. 432 p.
6. Mitchell T. M. Machine Learning: A Guide to Current Research. Boston, MA : Springer US, 1986. 429 p.
7. Network Anomaly Uncovering on CICIDS-2017Dataset: A Supervised Artificial Intelligence Approach [Електронний ресурс]. - режим доступу: URL: https://www.researchgate.net/publication/360773330_Network_Anomaly_Uncovering_on_CICIDS-2017_Dataset_A_Supervised_Artificial_Intelligence_Approach (дата звернення - 10.03.2024).
8. Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review [Електронний ресурс]. - режим доступу: URL: https://www.sciencedirect.com/science/article/pii/S1877050920311121 (дата звернення - 14.03.2024).

9. Ramasubramanian K., Singh A. Introduction to Machine Learning and R. *Machine Learning Using R*. Berkeley, CA, 2016. 566 p.
10. Raschka S., Julian D., Hearty J. Python: Deeper Insights into Machine Learning. Packt Publishing, 2017. 916 p.
11. Serrano L. Grokking Machine Learning. Manning Publications Co. LLC, 2021. 498 p.
12. What are support vector machines (SVMs)? [Електронний ресурс]. - режим доступу: URL: https://www.ibm.com/topics/support-vector-machine (дата звернення - 17.04.2024).
13. What is random forest? [Електронний ресурс]. - режим доступу: URL: https://www.ibm.com/topics/random-forest (дата звернення - 26.04.2024).

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **бакалавр**
Галузь знань: **12 – Інформаційні технології**
Спеціальність: **123 – Комп'ютерна інженерія.**

ЗАТВЕРДЖУЮ
Завідувач кафедри теоретичної
та прикладної системотехніки
_____ д.т.н., проф. Шматков С. І.
«21» грудня 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Ланіна Євгена Сергійовича

(прізвище, ім'я, по батькові студента)

1. Тема роботи **«Комп'ютерна модель виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання»**

керівник роботи Бакуменко Ніна Станіславівна, доцент кафедри ТПС, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «03» травня 2024року № 4101-5/909

2. Строк подання студентом роботи 31 травня 2024 року

3. Перелік питань, які потрібно розробити
1) Постановка задачі класифікації станів комп'ютерної мережі.
2) Аналіз існуючих методів машинного навчання для вирішення задачі класифікації.
3) Вибір та обґрунтування методів машинного навчання для виявленні атак в комп'ютерних мережах.
4) Розробка програмно-алгоритмічної моделі виявлення атак на основі обраного методу машинного навчання.
5) Тестування моделі та аналіз отриманих результатів.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Підбір наукової літератури	21.12.2023 - 25.01.2024
2	Огляд сучасних методів класифікації об’єктів	19.12.2023 - 2.01.2024
3	Аналіз інструментальних засобів для вирішення задач машинного навчання для виявлення атак в комп’ютерних мережах	2.01.2024 - 2.02.2024
4	Розробка комп’ютерної моделі виявлення атак в комп'ютерних мережах	2.01.2024 - 2.02.2024
5	Тестування та апробація розробленої системи	3.02.2024 - 30.03.2024
7	Оформлення пояснювальної записки та підготовка презентації	3.03.2024 - 31.05.2024
8	Представлення дипломного проекту керівнику дипломної роботи та рецензенту.	31.03.2024 - 31.05.2024
9	Оформлення звіту за результатами переддипломної практики.	15.05.2024 – 31.05.2024

5. Дата видачі завдання 21.12.2023

Студент Є. С. Ланін _____
ініціали, прізвище підпис

Керівник роботи Н. С. Бакуменко _____
ініціали, прізвище підпис

Додаток Б

Затверджую

_____

«_____»_____2024 р.

Технічне завдання

на розробку прототипу «Комп'ютерна модель виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання».

1.	Введення	1.1 Назва роботи – «Комп'ютерна модель виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання». 1.2. Галузь застосування: Інформаційні технології , комп’ютерні системи, кібербезпека.
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп’ютерна інженерія. 2.2. Завдання на кваліфікаційну роботу бакалавра № 4101-5/909 від «03» травня 2024 року (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).

3.	Призначення розробки	3.1. Мета розробки: забезпечення більш ефективного та автоматизованого захисту мережевих ресурсів від кібератак. 3.2. Призначення розробки: автоматизація процесу ідентифікації та класифікація потенційних загроз, зниженні помилкових тривог, адаптації до нових загроз та підтримці рішень з кібербезпеки через інтеграцію з іншими системами. 3.3. Вхідні дані: вхідні дані для комп'ютерної моделі виявлення атак у комп'ютерних мережах включають мережевий трафік, системні та застосункові логи, дані про конфігурації, відомі вразливості та сигнатури атак для тренування моделей машинного навчання в розпізнаванні загроз. 3.4. Вихідні дані розробки: результати прогнозу комп’ютерної моделі виявлення атак, які включають оцінки виявлення, сповіщення про потенційні загрози.

4.	Технічні вимоги до програмного виробу	4.1. Функціональні вимоги: - аналіз даних мережевого трафіку; - моделювання та класифікацію атак; - генерацію сповіщень. 4.2. Нефункціональні вимоги: - висока доступність системи, щоб забезпечити безперебійну роботу; - висока продуктивність системи з можливістю ефективної обробки великих обсягів даних; - простота в управлінні та налаштуванні для забезпечення легкості використання та підтримки. 4.3. Вимоги до інтеграції: - підтримка стандартних протоколів та форматів даних; - наявність детальної документацію та технічної підтримки; - забезпечення безпеки інтеграції. 4.4. Вимоги до безпеки: - механізми виявлення вторгнень; - механізм оновлення програмного забезпечення з появою нових видів атак - детальне журналювання для аудиту.

5.	Вимоги до програмної документації	Документацією до виробу «Комп'ютерна модель виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання» вважати: 2) Програму і методику випробувань розробленої програми (представити як додаток В до пояснювальної записки до кваліфікаційної роботи). 3) Опис розробленої комп’ютерної моделі (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).	
6.	Вимоги до техніко-економічних показників	Документацією до виробу «Комп'ютерна модель виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання» вважати: 1) Справжнє Технічне завдання на розробку прототипу мережі (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи). 2) Опис розробленої комп’ютерної моделі (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи). 3) Джерела базової інформації.	
7.		Дата	Назва етапу

	Стадії і етапи розробки	21.12.2023 - 25.01.2024	Підбір наукової літератури
		19.12.2023 - 2.01.2024	Огляд сучасних методів класифікації об’єктів
		2.01.2024 - 2.02.2024	Аналіз інструментальних засобів для вирішення задач машинного навчання для виявлення атак в комп’ютерних мережах.
		2.01.2024 - 2.02.2024	Розробка комп’ютерної моделі виявлення атак в комп'ютерних мережах.
		1.03.2024 - 15.03.2024	Тестування та апробація розробленої системи.
		3.03.2024 - 31.05.2024	Оформлення пояснювальної записки та підготовка презентації
		31.03.2024 - 31.05.2024	Представлення дипломного проекту керівнику дипломної роботи та рецензенту
		15.05.2024 - 31.05.2024	Оформлення звіту за результатами переддипломної практики.
		15.05.2024 - 31.05.2024	Представлення кваліфікаційної роботи керівнику та рецензенту.

8.	Порядок контролю і приймання програмного продукту (моделі)	1. Перевірку ходу розробки комп’ютерної моделі виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику.

Виконавець

Студент групи КІ-41

Ланін Є. С.

_____

Замовник

к.т.н., доц.

Бакуменко Н.С.

_____

Додаток В

Програма і методика випробувань програмного виробу

«Комп'ютерна модель виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання»

1. Об’єкт випробувань

1.1 Назва розробленого прототипу: «Комп'ютерна модель виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання».

1.2 Галузь застосування: Інформаційні технології , комп’ютерні системи, кібербезпека

1.3 Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональні можливості системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення

3.1 Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2 Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп’ютерного класу кафедри в період роботи атестаційної комісії.

3.3 Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавці та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

4.1. Функціональні вимоги:

- аналіз даних мережевого трафіку;
- моделювання та класифікацію атак;
- генерацію сповіщень.

4.2. Нефункціональні вимоги:

- висока доступність системи, щоб забезпечити безперебійну роботу;
- висока продуктивність системи з можливістю ефективної обробки великих обсягів даних;
- простота в управлінні та налаштуванні для забезпечення легкості використання та підтримки.

4.3. Вимоги до інтеграції:

- підтримка стандартних протоколів та форматів даних;
- наявність детальної документацію та технічної підтримки;
- забезпечення безпеки інтеграції.

4.4. Вимоги до безпеки:

- механізми виявлення вторгнень;
- механізм оновлення програмного забезпечення з появою нових видів атак;
- детальне журналювання для аудиту.

5. Вимоги до програмної документації

Програмною документацією до прототипу «Комп'ютерна модель виявлення атак в комп'ютерних мережах за допомогою методів машинного навчання» вважати:

1) Справжнє Технічне завдання на розробку прототипу мережі (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).

2) Опис реалізованого прототипу мережі (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).

3) Джерела базової інформації.

6. Засоби і порядок випробувань

6.1. Засоби випробувань

Засоби випробувань представлено на ПК на яких встановлено наступні програмні засоби: R, RStudio.

6.2. Порядок проведення випробувань

- **Перший етап:**
  Перевірка комплектності та якості програмної документації відповідно до ГОСТ 34.602-89.
- **Другий етап:**
  - Запуск системи та виконання тестів на різних наборах даних.
  - Оцінка точності моделі прогнозування атак на мережу.

7. Проведення випробувань

7.1 Завантаження та аналіз даних: Під час випробувань система має вміти правильно завантажувати дані з мережевим трафіком та видавати таблицю з вхідними даними.

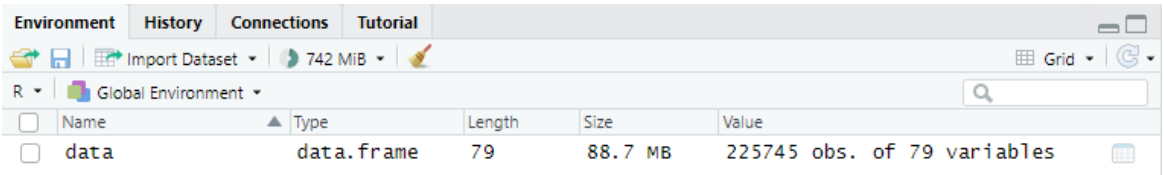
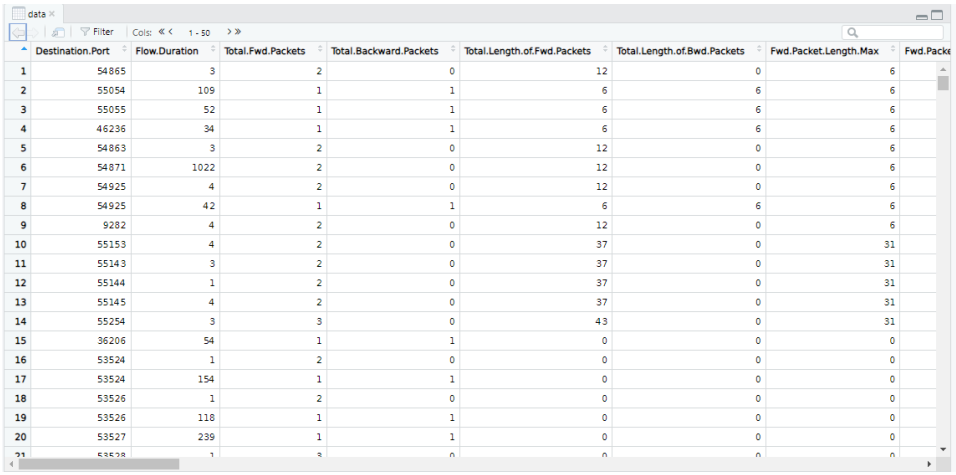


Рисунок В.1 – Результат правильного завантаження даних



data

Filter | Cols: « < 1 - 50 > »

	Destination.Port	Flow.Duration	Total.Fwd.Packets	Total.Backward.Packets	Total.Length.of.Fwd.Packets	Total.Length.of.Bwd.Packets	Fwd.Packet.Length.Max	Fwd.Packe
1	54865	3	2	0	12	0	6	
2	55054	109	1	1	6	6	6	
3	55055	52	1	1	6	6	6	
4	46236	34	1	1	6	6	6	
5	54863	3	2	0	12	0	6	
6	54871	1022	2	0	12	0	6	
7	54925	4	2	0	12	0	6	
8	54925	42	1	1	6	6	6	
9	9282	4	2	0	12	0	6	
10	55153	4	2	0	37	0	31	
11	55143	3	2	0	37	0	31	
12	55144	1	2	0	37	0	31	
13	55145	4	2	0	37	0	31	
14	55254	3	3	0	43	0	31	
15	36206	54	1	1	0	0	0	
16	53524	1	2	0	0	0	0	
17	53524	154	1	1	0	0	0	
18	53526	1	2	0	0	0	0	
19	53526	118	1	1	0	0	0	
20	53527	239	1	1	0	0	0	
21	53528	1	3	0	0	0	0	

Рисунок В.2 – Виведення таблиці завантаження даних

7.2 Результати випробувань:

Робиться поступове прогнозування атак на мережу за допомогою трьох моделей та виводиться результати роботи моделей. Результати прогнозування перевіряються на відповідність заявленим критеріям точності та функціональності.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 24403    34
         1    26 31972

               Accuracy : 0.9989
                 95% CI : (0.9986, 0.9992)
    No Information Rate : 0.5671
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9978

 Mcnemar's Test P-Value : 0.3662

            Sensitivity : 0.9989
            Specificity : 0.9989
         Pos Pred Value : 0.9986
         Neg Pred Value : 0.9992
             Prevalence : 0.4329
         Detection Rate : 0.4324
   Detection Prevalence : 0.4330
      Balanced Accuracy : 0.9989

       'Positive' Class : 0
```

Рисунок В.3 – Виведення матриця невідповідностей для моделі задля оцінки прогнозування моделі

Виконавець: студент групи КІ-41, Ланін Є.С. 

Додаток Г

Лістинг програмного коду

```
# Завантаження пакету, якщо потрібно
if (!require("readr")) install.packages("readr")
library(readr)

if (!require("tidyverse")) install.packages("tidyverse")
library(tidyverse)

if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)

if (!require("corrplot")) install.packages("corrplot")
library(corrplot)

if (!require("caret")) install.packages("caret")
library(caret)

if (!require("ROCR")) install.packages("ROCR")
library(ROCR)

if (!require("randomForest")) install.packages("randomForest")
library(randomForest)

if (!require("dplyr")) install.packages("dplyr")
library(dplyr)

# Завантаження даних
data <- read.csv("D:/Работы/КИ12/Ланин/Диплом/1Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv", sep=",")

# Описова статистика
summary(data)

# Аналіз даних для зазначених змінних
variables <- c("Flow.Duration", "Total.Fwd.Packets", "Total.Backward.Packets",
"Total.Length.of.Fwd.Packets", "Total.Length.of.Bwd.Packets",
"Fwd.Packet.Length.Max", "Bwd.Packet.Length.Max",
"Fwd.IAT.Mean", "Bwd.IAT.Mean", "Label")

# Перетворення змінної Label на числовий формат
df$Label <- as.numeric(as.factor(df$Label))
```

```
# Підрахунок кількості пропущених значень
missing_values <- sapply(df[variables], function(x) sum(is.na(x)))
print(missing_values)

# Статистичний опис змінних
summary(df[variables])

# Підготовка даних для кругової діаграми
label_data <- data %>%
group_by(Label) %>%
summarise(count = n()) %>%
mutate(percentage = count / sum(count) * 100,
label = ifelse(Label == 0, "Безпечний трафік", "DDoS трафік"))

# Створення кругової діаграми для Label
ggplot(label_data, aes(x = "", y = percentage, fill = label)) +
geom_bar(stat = "identity", width = 1) +
coord_polar("y") +
labs(title = "Розподіл трафіку за змінною Label", x = "", y = "") +
scale_fill_manual(values = c("Безпечний трафік" = "blue", "DDoS трафік" =
"red")) +
theme_void() +
theme(legend.title = element_blank())

# Візуалізація Flow.Duration
ggplot(data, aes(x = `Flow.Duration`, fill = as.factor(Label))) +
geom_histogram(position = "dodge", bins = 30) +
labs(title = "Розподіл тривалості потоку за змінною Label", x = "Тривалість
потоку (мікросекунди)", y = "Кількість") +
scale_fill_manual(values = c("0" = "blue", "1" = "red"), name = "Label", labels
= c("Безпечний трафік", "DDoS трафік"))

# Візуалізація Total.Fwd.Packets
ggplot(data, aes(x = `Total.Fwd.Packets`, fill = as.factor(Label))) +
geom_histogram(position = "dodge", bins = 50) + # Збільшення розміру бінів
scale_x_log10() + # Використання логарифмічної шкали для осі x
labs(title = "Розподіл загальної кількості отриманих пакетів за змінною
Label",
x = "Кількість отриманих пакетів (логарифмічна шкала)",
y = "Частота") +
scale_fill_manual(values = c("0" = "blue", "1" = "red"),
name = "Label",
labels = c("Безпечний трафік", "DDoS трафік")) +
```

```
    theme_minimal() +
    theme(legend.title = element_blank())

    # Візуалізація Total.Backward.Packets
    ggplot(data, aes(x = `Total.Backward.Packets`, fill = as.factor(Label))) +
    geom_histogram(position = "dodge", bins = 50) +
    scale_x_log10() +
    labs(title = "Розподіл кількості пакетів, надісланих у відповідь за змінною
Label",
    x = "Кількість надісланих у відповідь пакетів (логарифмічна шкала)",
    y = "Кількість") +
    scale_fill_manual(values = c("0" = "blue", "1" = "red"),
    name = "Label",
    labels = c("Безпечний трафік", "DDoS трафік")) +
    theme_minimal() +
    theme(legend.title = element_blank())


    # Візуалізація Total.Length.of.Fwd.Packets
    ggplot(data, aes(x = `Total.Length.of.Fwd.Packets`, fill = as.factor(Label))) +
    geom_density(alpha = 0.5) +
    scale_x_log10() +
    labs(title = "Розподіл загальної довжини отриманих пакетів за змінною
Label",
    x = "Загальна довжина пакетів вперед (логарифмічна шкала)",
    y = "Щільність") +
    scale_fill_manual(values = c("0" = "blue", "1" = "red"),
    name = "Label",
    labels = c("Безпечний трафік", "DDoS трафік")) +
    theme_minimal() +
    theme(legend.title = element_blank())

    # Візуалізація Total.Length.of.Bwd.Packets
    ggplot(data, aes(x = Label, y = `Total.Length.of.Bwd.Packets`, color =
as.factor(Label))) +
    geom_jitter(alpha = 0.5) +
    scale_y_log10() +
    labs(title = "Розподіл загальної довжини пакетів, надісланих у відповідь за
змінною Label",
    x = "Label",
    y = "Загальна довжина пакетів назад (логарифмічна шкала)") +
    scale_color_manual(values = c("0" = "blue", "1" = "red"),
    name = "Label",
    labels = c("Безпечний трафік", "DDoS трафік")) +
```

```
theme_minimal() +
theme(legend.title = element_blank())


# Візуалізація Fwd.Packet.Length.Max
ggplot(data, aes(x = Label, y = `Fwd.Packet.Length.Max`, color = as.factor(Label))) +
geom_jitter(alpha = 0.5) +
scale_y_log10() +
labs(title = "Розподіл максимальної довжини ориманного пакету за змінною Label",
x = "Максимальна довжина пакету вперед (логарифмічна шкала)",
y = "Кількість") +
scale_color_manual(values = c("0" = "blue", "1" = "red"),
name = "Label",
labels = c("Безпечний трафік", "DDoS трафік")) +
theme_minimal() +
theme(legend.title = element_blank())

# Візуалізація Bwd.Packet.Length.Max
ggplot(data, aes(x = `Bwd.Packet.Length.Max`, fill = as.factor(Label))) +
geom_histogram(position = "dodge", bins = 30) +
labs(title = "Розподіл максимальної довжини пакету, надісланого у відповідь за змінною Label", x = "Максимальна довжина пакету назад", y = "Кількість") +
scale_fill_manual(values = c("0" = "blue", "1" = "red"), name = "Label", labels = c("Безпечний трафік", "DDoS трафік"))

# Візуалізація Fwd IAT Mean
ggplot(data, aes(x = `Fwd.IAT.Mean`, fill = as.factor(Label))) +
geom_density(alpha = 0.5) +
scale_x_log10() +
labs(title = "Розподіл середнього інтервалу між отриманими пакетами за змінною Label",
x = "Середній інтервал між пакетами вперед (логарифмічна шкала)",
y = "Щільність") +
scale_fill_manual(values = c("0" = "blue", "1" = "red"),
name = "Label",
labels = c("Безпечний трафік", "DDoS трафік")) +
theme_minimal() +
theme(legend.title = element_blank())


# Візуалізація Bwd.IAT.Mean
```

```
ggplot(data, aes(x = `Bwd.IAT.Mean`, fill = as.factor(Label))) +
geom_density(alpha = 0.5) +
scale_x_log10() +
labs(title = "Розподіл середнього інтервалу між пакетами, надісланими у відповідь за змінною Label",
x = "Середній інтервал між пакетами назад (логарифмічна шкала)",
y = "Щільність") +
scale_fill_manual(values = c("0" = "blue", "1" = "red"),
name = "Label",
labels = c("Безпечний трафік", "DDoS трафік")) +
theme_minimal() +
theme(legend.title = element_blank())


# Кореляція між змінними
cor_matrix <- cor(df[variables], use = "complete.obs")

# Графік кореляційної матриці
corrplot(cor_matrix, method = "circle")


#_____________________________________________________________________
_____________________________________________


set.seed(123)
split <- createDataPartition(data$Label, p = 0.75, list = FALSE)
trainData <- data[split,]
testData <- data[-split,]

# Навчання моделі логістичної регресії
model <- glm(Label ~ Bwd.Packet.Length.Max + Fwd.Packet.Length.Max
+ Total.Length.of.Fwd.Packets, data = trainData, family = "binomial")

# Передбачення на тестовій вибірці
predictions <- predict(model, testData, type = "response")
prediction_class <- ifelse(predictions > 0.5, 1, 0)

# Оцінка моделі
confusionMatrix <- confusionMatrix(as.factor(prediction_class),
as.factor(testData$Label))
print(confusionMatrix)
```

```
#_____
_____

# Видалення стовпців з усіма пропущеними значеннями
data <- data %>% select_if(~ !all(is.na(.)))

# Перетворення міток у фактор
data$Label <- as.factor(data$Label)

# Вибір тільки числових змінних
numeric_data <- data %>% select(where(is.numeric))

# Видалення числових змінних з нульовою стандартною девіацією
numeric_data <- numeric_data %>% select_if(~ sd(., na.rm = TRUE) != 0)

# Об'єднання числових змінних з мітками
data <- bind_cols(numeric_data, data %>% select(Label))

# Розділення даних на навчальну та тестову вибірки
set.seed(123) # Для відтворюваності результатів
trainIndex <- createDataPartition(data$Label, p = 0.75,
list = FALSE,
times = 1)
dataTrain <- data[trainIndex,]
dataTest  <- data[-trainIndex,]

# Перевірка кореляцій для вибору релевантних змінних
correlation_matrix <- cor(select(dataTrain, -Label), use = "complete.obs")
highly_correlated <- findCorrelation(correlation_matrix, cutoff = 0.9)
dataTrain <- dataTrain[ , -highly_correlated]
dataTest <- dataTest[ , -highly_correlated]

# Побудова моделі логістичної регресії
model <- train(Label ~ ., data = dataTrain, method = "glm", family = "binomial",
trControl = trainControl(method = "cv", number = 5))

# Оцінка моделі на тестовій вибірці
predictions <- predict(model, newdata = dataTest)
confMatrix <- confusionMatrix(predictions, dataTest$Label)

# Виведення результатів
print(confMatrix)
```

```
#_____
_____

set.seed(123)
split <- createDataPartition(data$Label, p = 0.75, list = FALSE)
trainData <- data[split,]
testData <- data[-split,]


# Навчання моделі випадкового лісу
rf_model <- randomForest(Label ~ Bwd.Packet.Length.Max + Fwd.Packet.Length.Max
+ Total.Length.of.Fwd.Packets,
data = trainData, ntree = 500, mtry = 3, importance = TRUE)

# Передбачення на тестовій вибірці
rf_predictions <- predict(rf_model, testData)

# Оцінка моделі випадкового лісу
rf_confusionMatrix <- confusionMatrix(as.factor(rf_predictions), as.factor(testData$Label))
print(rf_confusionMatrix)
```