**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**
**імені ІГОРЯ СІКОРСЬКОГО»**

**Навчально-науковий інститут телекомунікаційних систем**

**Кафедра електронних комунікацій та інтернету речей**

«На правах рукопису»
УДК 004.056.53

До захисту допущено:
В.о. завідувача кафедри
_____ Вячеслав НОСКОВ
«___»_____20__ р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-професійною програмою «Системи електронних комунікацій та Інтернету речей»**

**зі спеціальності 172 «Електронні комунікації та радіотехніка»**

**на тему: «Система виявлення вторгнень у програмно-визначеному мережевому середовищі на основі нейронної мережі»**

Виконав:
студент VI курсу, групи ЦС-31мп
Могилевич Вадим Дмитрович _____

Науковий керівник:
Професор кафедри, доктор технічних наук, професор
Уривський Леонід Олександрович _____

Рецензент:
Доцент кафедри, кандидат технічних наук, доцент
Кононова Ірина Віталіївна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2024 року

**Національний технічний університет України**
**«Київський політехнічний інститут імені Ігоря Сікорського»**
**Навчально-науковий інститут телекомунікаційних систем**
**Кафедра електронних комунікацій та інтернету речей**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Електронні комунікації та радіотехніка»

Освітньо-професійна програма «Системи електронних комунікацій та Інтернету речей»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Вячеслав НОСКОВ
«___»_____20__ р.

**ЗАВДАННЯ**
**на магістерську дисертацію студенту**
**Могилевичу Вадиму Дмитровичу**

1. Тема дисертації «Система виявлення вторгнень у програмно-визначеному мережевому середовищі на основі нейронної мережі», науковий керівник дисертації д.т.н, професор Уривський Леонід Олександрович, затверджені наказом по університету від «07» листопада 2024 р. №4989-С

2. Термін подання студентом дисертації _____

3. Об’єктом дослідження є процес функціонування SDN.

4. Перелік завдань, які потрібно розробити:

- Аналіз існуючих методів виявлення DDoS-атак в SDN та їх обмеження.
- Розробка архітектури системи виявлення вторгнень на основі глибокого навчання.
- Вибір та обґрунтування набору ознак для класифікації мережевого трафіку.
- Навчання та оцінка різних моделей глибокого навчання на еталонних наборах даних.
- Порівняння ефективності запропонованої системи з іншими існуючими рішеннями.

5. Орієнтовний перелік графічного (ілюстративного) матеріалу

- Інфраструктура SDN з її відокремленими площинами управління та передачі даних, а також різними мережевими додатками.
- Структура згорткової нейронної мережі.
- Етапи впровадження NIDS.

6. Орієнтовний перелік публікацій

- Аналіз атак в програмно-керованих мережах.
- Оцінка впливу атак в програмно-керованих мережах.
- Комплексне використання програмно-визначеної мережі та глибокого навчання для виявлення атак.
- Аналіз вразливостей протоколів комутації у програмно-визначених мережах.

7. Дата видачі завдання «10» листопада 2023 р.

Календарний план

| № з/п | Назва етапів виконання магістерської дисертації | Термін виконання етапів магістерської дисертації | Примітка |
|---|---|---|---|
| 1 | Вибір проблеми, теми, мети дослідження | До 15.11.23 | Виконано |
| 2 | Визначення завдань дослідження, структури магістерської дисертації | До 30.11.23 | Виконано |
| 3 | Бібліографічний пошук наукової інформації, підготовка літератури | До 01.02.24 | Виконано |
| 4 | Підготовка матеріалів 1 розділу дисертації | До 01.03.24 | Виконано |
| 5 | Вибір напряму і основних методів дослідження | До 15.03.24 | Виконано |
| 6 | Розробка програми експериментального дослідження | До 01.05.24 | Виконано |
| 7 | Підготовка матеріалів 2 розділу дисертації | До 01.07.24 | Виконано |
| 8 | Підготовка рекомендацій щодо практичного використання результатів дослідження | До 15.07.24 | Виконано |
| 9 | Підготовка матеріалів 3 розділу дисертації | До 01.09.24 | Виконано |

| | | | |
|---|---|---|---|
| 10 | Підготовка матеріалів 4 розділу дисертації | До 01.10.24 | Виконано |
| 11 | Оформлення результатів та фінальні правки | До 30.11.24 | Виконано |

Студент Вадим МОГИЛЕВИЧ

Науковий керівник Леонід УРИВСЬКИЙ

# РЕФЕРАТ

Текстова частина магістерської дисертації: 106 с., 26 рис., 27 табл., 70 джерел.

Актуальність роботи. У сучасних умовах стрімкого розвитку інформаційних технологій та збільшення обсягів передавання даних зростає необхідність у підвищенні рівня безпеки комп'ютерних мереж. Програмно-визначені мережі набули значного поширення завдяки своїй гнучкості, масштабованості та можливостям централізованого управління мережею. Однак, SDN також стикаються з новими загрозами безпеці, що обумовлює потребу у створенні надійних систем для виявлення вторгнень, здатних захистити мережеві інфраструктури від атак і несанкціонованого доступу.

Традиційні IDS не завжди ефективно працюють у середовищах SDN через відмінності в архітектурі, динамічний характер мереж та великий обсяг трафіку. Використання нейронних мереж у IDS дозволяє підвищити точність виявлення вторгнень шляхом аналізу аномалій та шаблонів трафіку в реальному часі.

Метою даної роботи є удосконалення та оцінка ефективності системи виявлення DDoS-атак в програмно-визначених мережах на основі глибокого навчання.

Об’єктом дослідження є процес функціонування мережевої системи виявлення вторгнень.

Предметом дослідження є методи виявлення вторгнень на основі машинного навчання.

У цій роботі досліджуються особливості архітектури програмно-визначених мереж (SDN) та вплив різних видів атак на їх функціонування. Також проводиться обчислення показників точності, повноти та F-міри для моделей машинного навчання, на основі яких будуються графіки для наочності результатів.

Ключові слова: система виявлення вторгнень, програмно визначена мережа, DDoS атаки, нейронні мережі, машинне навчання.

# ABSTRACT

Text section of the thesis: 106 pages, 26 figures, 27 tables, 70 sources.

Relevance of the study. In the modern environment of rapid information technology development and increasing data transmission volumes, the need to enhance computer network security is growing. Software-Defined Networks (SDN) have gained significant popularity due to their flexibility, scalability, and centralized network management capabilities. However, SDNs also face new security threats, necessitating the creation of reliable intrusion detection systems capable of protecting network infrastructures from attacks and unauthorized access.

Traditional Intrusion Detection Systems (IDS) do not always work effectively in SDN environments due to architectural differences, the dynamic nature of networks, and high traffic volumes. The use of neural networks in IDS allows for improved intrusion detection accuracy by analyzing traffic anomalies and patterns in real time.

The aim of this study is to improve and evaluate the effectiveness of a DDoS attack detection system in SDN based on deep learning.

The object of research is the functioning process of a network IDS.

The subject of research is intrusion detection methods based on machine learning (ML).

This study examines the architecture of SDN and the impact of various types of attacks on their functionality. It also involves calculating accuracy, recall, and F-measure metrics for ML models, with visualizations of results for clarity.



Keywords: intrusion detection system, software-defined network, machine learning, DDoS attacks, neural networks.

# ЗМІСТ

# ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

API – application programming interface;

DDoS – distributed denial of service;

DL – deep learning;

IDS – intrusion detection system;

IoT – internet of things;

ML – machine learning;

NIDS – network-based intrusion detection system;

OF – OpenFlow;

SAE – stacked autoencoder;

SDN – software-defined network;

SLA – service level agreement;

SMB – small and medium-sized business;

SMR – softmax regression;

STL – self-taught learning;

ЗНМ – згорткові нейронні мережі;

МН – машинне навчання;

РНМ – рекурентні нейронні мережі;

СППР – система підтримки прийняття рішень.

# ВСТУП

У сучасному цифровому світі, де кіберзагрози стають дедалі витонченішими і масштабнішими, забезпечення безпеки інфо-комунікаційних систем є одним з найважливіших завдань.

Програмно-визначені мережі (SDN) пропонують нові можливості для підвищення гнучкості та ефективності мережевих інфраструктур, але водночас вони стають новою мішенню для кібератак.

Машинне навчання демонструє значний потенціал у виявленні аномалій у мережевому трафіку, що робить його перспективним інструментом для боротьби з DDoS-атаками. Його здатність навчатися на великих обсягах даних, виявляти складні шаблони та адаптуватися до нових типів атак робить його ідеальним кандидатом для створення ефективних систем виявлення вторгнень.

Метою даної роботи є удосконалення та оцінка ефективності системи виявлення DDoS-атак в програмно-визначених мережах на основі глибокого навчання. Зокрема, ми зосередимося на виявленні багатовекторних атак, які комбінують різні типи атак для підвищення ефективності.

Наукова новизна роботи полягає у використанні глибоких нейронних мереж для виявлення DDoS-атак в SDN, а також у розробці нового набору ознак, які дозволяють ефективно розрізняти нормальний та аномальний трафік.

Практична значущість роботи полягає в тому, що розроблена система може бути використана для підвищення рівня безпеки програмно-визначених мереж і забезпечення безперебійної роботи критично важливих сервісів.

В роботі будуть розглянуті такі питання:

- Аналіз існуючих методів виявлення DDoS-атак в SDN та їх обмеження.
- Розробка архітектури системи виявлення вторгнень на основі глибокого навчання.
- Вибір та обґрунтування набору ознак для класифікації мережевого трафіку.

- Навчання та оцінка різних моделей глибокого навчання на еталонних наборах даних.
- Порівняння ефективності запропонованої системи з іншими існуючими рішеннями.

Результати дослідження можуть бути використані для створення більш ефективних систем захисту інформаційних систем від DDoS-атак та сприяти розвитку технологій кібербезпеки.

# 1 АНАЛІЗ ПРОГРАМНО-ВИЗНАЧЕНОЇ МЕРЕЖІ ТА ГЛИБОКОГО НАВЧАННЯ

З бурхливим розвитком інформаційних технологій та появою новітніх трендів, таких як хмарні обчислення та Big Data, зростають й вимоги до електронних комунікаційних мереж. Їх реалізація стає дедалі складнішою, адже:

обсяг даних, що передаються мережами, постійно зростає, що потребує більшої пропускної здатності та потужніших мережевих пристроїв;

сучасні мережі забезпечують передавання трафіку різного типу, включаючи веб-трафік, потокове відео, голосові дзвінки та IoT-дані. Кожен тип трафіку має свої характеристики та вимоги до обслуговування, що ускладнює його ефективне керування;

користувачі вимагають швидкого доступу до контенту та додатків, що потребує від мережі можливості передавати дані на високих швидкостях;

для ефективного керування складними мережами з різноманітним трафіком потрібні потужні та багатофункціональні інструменти для моніторингу та управління.

Це зумовлює появу нових, функціонально складних мереж з ускладненою інфраструктурою. Старі методи моніторингу та управління вже не відповідають сучасним вимогам, тому виникає гостра потреба в інноваційних підходах.

З зростанням розмірів мережі управління нею стає складним. Мережа з недоліками в управлінні відкриває для зловмисників шляхи для використання вразливостей безпеки для вторгнень. Крім того, недорогі інтернет-підписки та загальнодоступні інструменти для атак дозволяють зловмисникам запускати невиявлені атаки або атаки «нульового дня в мережі. Використання систем на основі машинного навчання дозволяють виявляти такі види атак. Однак ручна інженерія, пов'язана з підходами машинного навчання для правильного вибору ознак з мережевого трафіку обмежує точність виявлення атак.

Сумісне використання програмно-визначених мереж та нейроних мереж у

вигляді глубого навчання дозволять вчасно виявляти атаки за зменшувати їх вплив на мережеві сервіси. SDN централізує управління мережею і контролює її з логічно єдиної точки. Підхід на основі DL значно покращує вибір ознак для класифікації або прогнозування в неконтрольованому режимі [1].

В розділі проведено аналіз програмно-визначених мереж, а також технології глибокого навчання.

## 1.1 Архітектура програмно-визначеної мережі

Основна концепція SDN полягає у розділенні площин управління та передачі даних мережевих пристроїв (комутаторів). Функції управління переміщуються з комутаторів до логічно централізованого контролера.

Контролер має інформацію про всю визначену мережу та дозволяє адміністратору регулювати трафік централізовано.

Комутатори слугують елементами переадресації та можуть бути налаштовані для виконання різних завдань за допомогою мережевих додатків, реалізованих на контролері. Комутатори та інші пристрої забезпечують пересилку трафіку відповідно до інструкцій контролера.

SDN в порівнянні з традиційною мережею надає широкі можливості для мережевого менеджменту та проведення досліджень [2]. У програмно-визначених мережах контролер відіграє ключову роль, надаючи абстракцію визначеної мережі для додатків. Це дозволяє централізовано вирішувати такі завдання, як маршрутизація та комутація, замість того, щоб ці функції виконувалися на кожному мережевому пристрої окремо. SDN надає інтерфейси для реалізації політик ефективного управління мережею, а також дозволяє швидко розробляти прототип і аналізувати його вплив на робочу мережу, не впливаючи на реальний трафік, розділяючи трафік за допомогою додатків SDN.

Таким чином, часовий проміжок для переходу від прототипу до реальної реалізації значно скорочується. Завдяки цьому SDN широко застосовується при

дослідженні трафіку, мобільності, моніторингу мережі та мереж центрів обробки даних [3].

Площина управління і площина даних відокремлені від комутаторів в SDN, отже, комутатори стають елементами пересилання пакетів.

На рис. 1.1 показана інфраструктура SDN (особливість − відокремлення площин управління, даних та додатків).

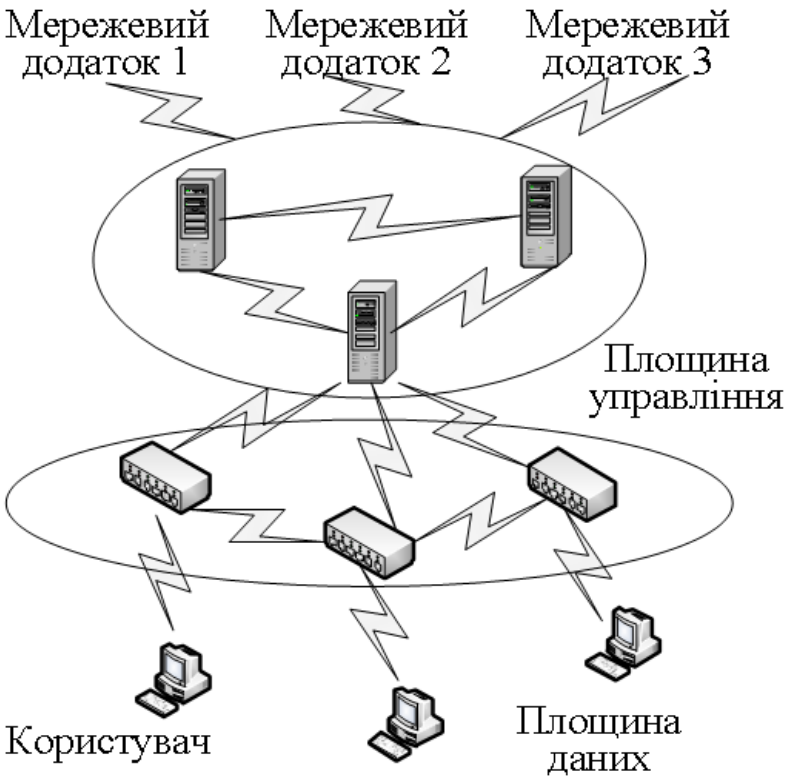


Рисунок 1.1 − Інфраструктура SDN з її відокремленими площинами управління та передачі даних, а також різними мережевими додатками

Комутатори, кінцеві хости та зв'язок між ними формують площину даних. Контролер з об'єднаною площиною управління може бути як окремим сервером, так і групою логічно централізованих, але розподілених серверів. Контролер взаємодіє з комутаторами через API інтерфейси [4].

Контролер і комутатори обмінюються керуючими і конфігураційними повідомленнями, інкапсульованими в протоколі OpenFlow (OF), використовуючи зашифрований або звичайний канал зв'язку TCP [5]. Ці повідомлення використовуються для налаштування з'єднання комутаторів з контролером, збору статистики, а також для управління потоками трафіку в мережі. OpenFlow

реалізує апаратну абстракцію, надаючи контролеру можливість взаємодії з пристроями різних виробників та апаратними типами (маршрутизаторами, комутаторами, балансувальниками навантаження тощо) через стандартний інтерфейс. Протокол OpenFlow функціонує як інтерфейс між контролером та комутатором, налаштовуючи таблицю потоку (flow tables) [6].

Комутатори мають декілька таблиць потоків, в яких відображаються правила потоків. Такі таблиці визначають дії які повинні відбуватись з підмножиною пакетів, що відносяться до потоку: пересилка (forward), відкидання (drop) або модифікацію заголовків (redirect).

Пересилка (переадресація) пакетів потоку на визначений порт (або набір портів) забезпечує комутацію пакетів через мережу. У більшості комутаторів пересилка здійснюється на лінійних швидкостях.

Відкидання пакетів потоку може використовуватися з міркувань безпеки, блокуючи несанкціонований трафік, зупиняючи DOS-атаки або зменшуючи надмірний широкомовний трафік з кінцевих хостів.

Ікапсуляція пакету та переадресація на контролер SDN дозволяє контролеру приймати рішення та пересилати пакет назад до комутатора. Зазвичай цей метод використовується для першого пакету нового потоку, даючи контролеру можливість вирішити, чи додавати потік до таблиці потоку. В іншому випадку цей метод може бути використаний для пересилки всіх пакетів на контролер, якщо програмний додаток вимагає такої функціональності.

Потік в SDN визначається як мережевий потік, який містить пакети з однаковими значеннями певних полів заголовка. Контролер встановлює правила для потоків на основі політик, продиктованих мережевими додатками. У табл. 1.1 наведено приклад таблиці потоків з кількома встановленими правилами [7].

Таблиця 1.1 − Таблиця потоків у комутаторі з встановленими правилами

| Поля співпадіння | Дії | Лічільник | | Пріоритет |
|---|---|---|---|---|
| | | Пакети | Байти | |
| tcp,nw dst=10.10.1.5,tp dst=110 | 1 | 152 | 1248 | 1000 |
| icmp,nw dst=10.10.1.10,icmp type=0 | drop | 1000 | 64000 | 2000 |
| tcp,nw dst=10.10.1.10,tp dst=80 | 2 | 355 | 9424 | 1000 |

Розглянемо інсталяцію потоку в SDN. Правила потоку встановлюються в таблицях потоків комутаторів проактивно або реактивно [7]. У проактивному режимі контролер встановлює правила всередині таблиць, коли комутатор приєднується до мережі. У реактивному режимі, навпаки, контролер встановлює правила у відповідь на вхідні потоки.

Встановлення правил в реактивному режимі відбувається наступним чином: пакет, що відповідає потоку, потрапляє на комутатор, шукає правило, яке відповідає заголовкам вхідного пакета, у своїх таблицях потоків. Якщо в таблицях для потоку пакета існує правило (знайдено співпадаючий запис), то комутатор виконує дії, що відповідають цьому правилу. Якщо правило недоступне (в таблиці потоку не знайдено жодного співпадання, тобто відбувається подія table-miss), то створюється повідомлення PACKET_IN яке інкапсулює заголовки пакетів вхідного потоку та надсилає їх контролеру. Контролер аналізує заголовки вхідного потоку з PACKET_IN і надсилає повідомлення PACKET_OUT або FLOW_MOD на комутатори на основі політик мережевих додатків.

Правила потоку встановлюються, коли контролер надсилає повідомлення FLOW_MOD, що інкапсулює різні поля правила, яке буде встановлено для потоку. Комутатори виконують дії над наступними пакетами потоку, використовуючи встановлене правило, замість того, щоб перенаправляти їх на контролер. Встановлення правил потоку в реактивному режимі в SDN показано на рис. 1.2. [8].

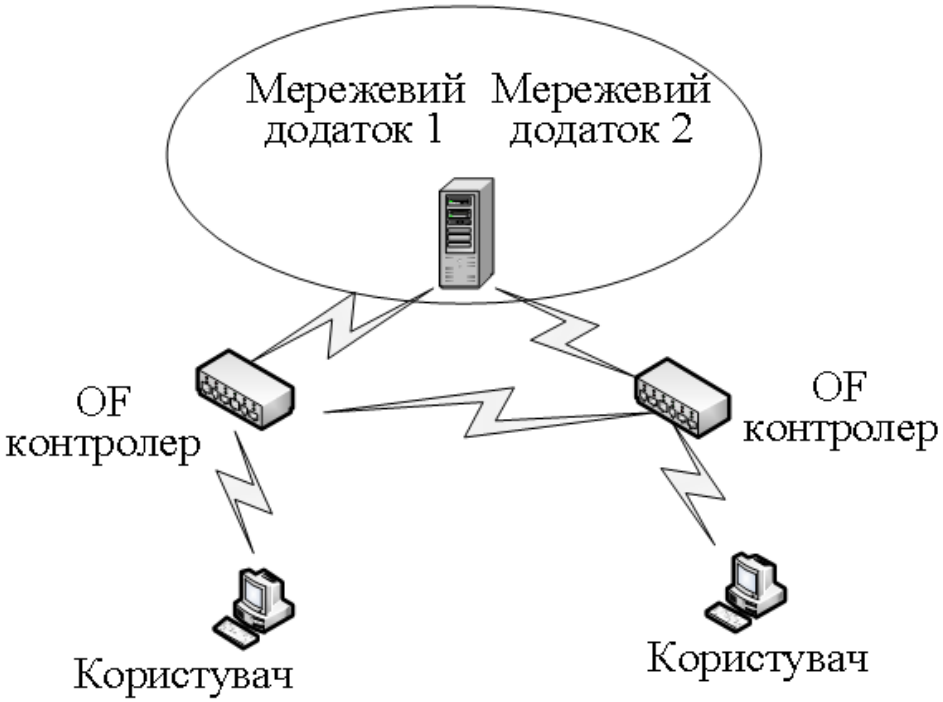


Рисунок 1.2 − Встановлення правил потоку реактивного режиму в SDN

Правила потоку мають жорсткий тайм-аут (тайм-аут простою) та видаляються з таблиць потоків для управління розміром пам'яті таблиць. Правила не встановлюються у відповідь на повідомлення PACKET_OUT. Контролер дає вказівку комутаторам пересилати пакет з одного або декількох портів без встановлення правил для пакету.

Отже, аналіз показав, що однією з основних переваг SDN є централізоване управління, проте це також створює критичну вразливість. Якщо контролер SDN буде скомпрометовано або він зазнає відмови, вся мережа може вийти з ладу. Оскільки контролер SDN має повний контроль над мережею, він є мішенню для кібератак. Атаки типу "людина посередині", підробка правил потоку, DDoS-атаки на контролер можуть легко паралізувати всю мережу.

## 1.2 Аналіз глибоких нейронних мереж

Машинне навчання (МН) успішно застосовується для вирішення завдань, які важко сформулювати математично. Підходи машинного навчання, що

застосовуються для вирішення цих завдань, використовують створені вручну ознаки, отримані з необроблених даних, і навчаються на цих ознаках. Ці створені вручну ознаки надаються як вхідні дані для методів класифікації або регресії. Однак, розробка ознак вимагає багато зусиль, потребує знання предметної області та впливає на продуктивність системи [9]. Глибоке навчання (Deep learning, DL), це новий підхід машинного навчання, що дозволяє вирішувати завдання, що пов'язані з інженерією ознак. Існують різні підходи DL, які використовуються для вирішення проблем в області комп'ютерного зору, обробки зображень і розпізнавання мови, такі як SAE, RBM, згорткові нейронні мережі та рекурентні нейронні мережі. Ці підходи перевершують різні традиційні підходи машинного навчання для вирішення багатьох проблем [10].

Для розпізнавання сигналів та образів найбільш популярними та ефективними є згорткові нейронні мережі (ЗНМ). Їх можна застосувати для будь-якого сигналу, чи то даних з датчиків, аудіосигналів, образів тощо. Цей вид нейромереж є багато-шаровим перцептроном, що складається з багатьох рівнів вузлів, прихованих і вихідного шарів, та має односпрямований інформаційний потік. Функцією активації для вузлів прихованого шару зазвичай обирається монотонна нелінійна S-подібна функція, в той час як для вузлів вихідного шару достатньо використовувати лінійну функцію. Цей тип нейронної мережі може апроксимувати будь-яке неперервне відображення, якщо кількість прихованих шарів є достатньо великою. У задачах розпізнавання образів нейронні мережі з нелінійною S-подібною активаційною функцією та кількома шарами можуть досягати високої точності при впізнаванні об'єктів. Ці характеристики багатошарової нейронної мережі прямого поширення є теоретичною основою для застосування багатошарових перцептронів у моделюванні та діагностиці похибок розпізнавання образів. Похибки можуть бути визначені двома способами: шляхом програмування моделі виправлення або вибору класифікатора шаблонів.

Якість розпізнавання образів нейронними мережами залежить від ефективності навчання на основі вибірки даних із великою кількістю

навчальних пар (вхід-вихід). За результатами навчання визначається функція помилки (функція втрат). Процес навчання направлений на мінімізацію цієї помилки, що дозволяє штучному інтелекту коригувати допустимі ваги зв'язків між нейронами.

Архітектура згорткової нейронної мережі, як показано на рис. 1.3, має важливу особливість: нейрони шару не мають індивідуальних вагових коефіцієнтів, а використовують спільні ваги у вигляді матриць невеликого розміру, званих ядрами згортки. Це дозволяє зменшити кількість параметрів порівняно з повнозв'язними мережами, що сприяє підвищенню продуктивності та економії пам'яті [11].

Рекурентні нейронні мережі (РНМ) – це клас нейронних мереж, де зв'язки між елементами утворюють направлену послідовність. Вони використовуються здебільшого для генерації послідовних даних, таких як текст, природна мова та часові ряди. РНМ називаються рекурентними, оскільки виконують однакову задачу для кожного елемента послідовності, причому результат залежить від попередніх обчислень. Теоретично, рекурентні мережі можуть обробляти послідовності будь-якої довжини, хоча на практиці використовуються зазвичай лише вихідні дані останніх кількох обчислень. Рекурентна нейронна мережа розгортається в кілька шарів, причому розгортка залежить від кількості слів у послідовності. Наприклад, при введенні речення з *n* слів використовується *n* шарів.

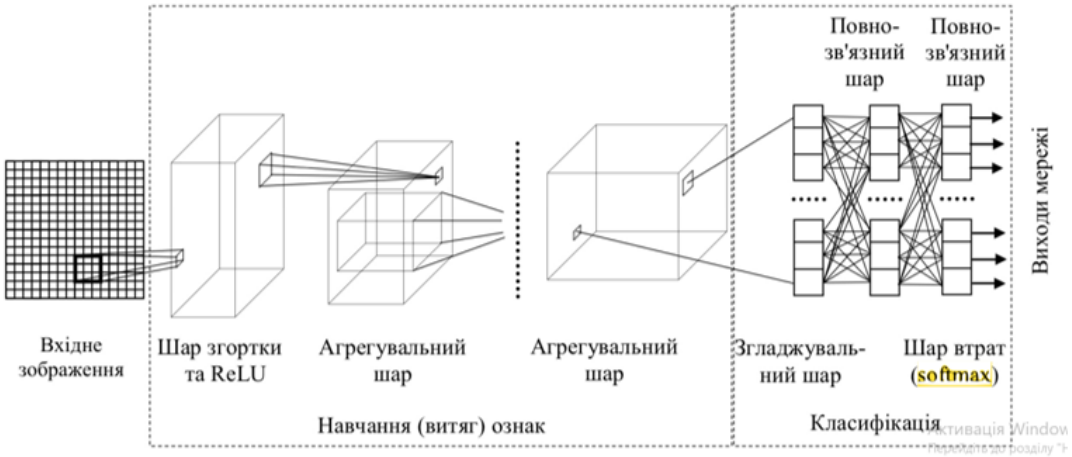


Рисунок 1.3 − Структура згорткової нейронної мережі

До переваг РНМ можна віднести:

здатність до обробки послідовних даних. РНМ можуть враховувати контекст попередніх елементів у послідовності, що робить їх ідеальними для задач, де важлива залежність між даними;

моделювання довгострокових залежностей. РНМ можуть моделювати залежності між елементами, які знаходяться далеко один від одного в послідовності, що робить їх кращими для задач, де важлива довгострокова пам'ять;

гнучкість. РНМ можуть бути використані для широкого спектру задач, включаючи класифікацію, прогнозування та генерацію;

можливість навчання на невеликих наборах даних. РНМ можуть навчатися на невеликих наборах даних, що робить їх корисними для задач, де доступ до даних обмежений.

До недоліків РНМ відносяться:

складність. РНМ складніші для навчання та розуміння, ніж інші типи нейронних мереж;

чутливість до шуму. РНМ можуть бути чутливі до шуму в даних, що може негативно впливати на їхню продуктивність;

високі обчислювальні витрати. Навчання та використання РНМ може бути обчислювально дорогим [12].

Стековий автокодер − це модель нейронної мережі, яка кодує вхідні дані в абстрактне представлення, а потім декодує це представлення для відновлення тих самих вхідних даних на виході [13].

Автокодери використовуються для зменшення розмірів даних, коли нелінійна функція описує зв’язок між залежними та незалежними функціями. Автокодери використовуються для автоматичного виділення ознак із даних. Розріджений автокодер складається з трьох шарів, що мають по $M$ вузлів на вхідному та вихідному шарах, а також $N$ вузлів на прихованому шарі. М вхідних вузлів представляють запис з $M$ атрибутами, тобто, $X = \{x_1, x_2, ..., x_m\}$. На етапі навчання отримані результати $\hat{X}$ на вихідному шарі порівнюється з

вхідним $X$ щоб з'ясувати, чи достатньо декодував прихований шар. Розріджена мережа автокодерів, показана на рис. 1.4 а, знаходить оптимальні ваги матриць, $U \in R^{N*M}$ та $U^{'} \in R^{N*M}$, і базовий вектор $b_1 \in R^{N*1}$ та $b_1^{'} \in R^{N*1}$ при цьому намагаючись знайти наближення функції тотожності, тобто $X \approx \hat{X}$ під час навчання.

$$J = \frac{1}{2r}\left(\sum_{i=1}^{r}\| X_i - \hat{X}_i \|^2 + \frac{\lambda}{2}\sum_{n,m} U^2 + \sum_{n,m} U^{r2} + \sum_{n} b_1^2 + \sum_{n} b_1^{r2}\right) + \beta\sum_{j=1}^{N} KL\left(\rho \| \hat{\rho}_j\right) \qquad (1.1)$$

Рівняння (1.1) представляє функцію вартості, яка використовується для навчання оптимальної ваги в розрідженому автокодері. Функція вартості мінімізується за допомогою градієнтного спуску. Алгоритм зворотного поширення використовується для обчислення частинних похідних, необхідних для градієнтного спуску.

Функція вартості подібна до тієї, що використовується в нейронній мережі, за винятком третього члена. Перший член рівняння являє собою середнє значення суми квадратів різниць всіх вхідних значень і відповідних вихідних значень для всіх $r$ записів у наборі даних. Другий член − це член регуляризації з $\lambda$ як параметром вагового розпаду для подолання перенавчання моделі.

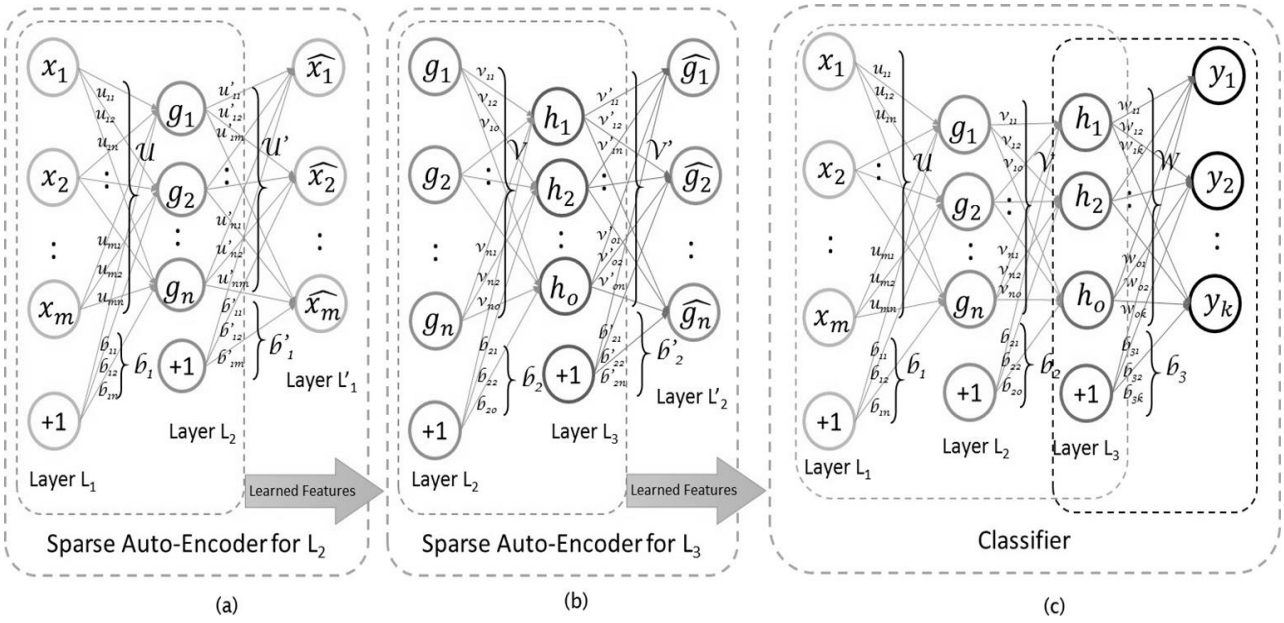


Рисунок 1.4 − Модель глибокого навчання на основі стекового автокодера (SAE)

Останній член є штрафним членом розрідженості, який використовується як обмеження для вузлів прихованого шару, щоб підтримувати низькі середні значення активації. Він виражається за допомогою розбіжності Кульбака-Лейблера (KL), як показано в (1.2)

$$KL\left(\rho \| \hat{\rho}_j\right) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \tag{1.2}$$

де $\rho \in \{0, 1\}$ − параметр обмеження розрідженості.

$KL\left(\rho \| \hat{\rho}_j\right)$ стає мінімальним, коли $\rho = \hat{\rho}_j$, де $\hat{\rho}_j$ представляє середнє значення активації прихованого вузла $j$ за всіма навчальними входами. Для активації вузлів на різних рівнях використовується багато функцій, таких як сигмоїда, гіперболічний тангенс та випрямлена лінійна одиниця (*ReLU*). Використовуємо сигмоїдну функцію, $g_{U,b_1}(z) = \frac{1}{1+e^{-z}}$ для активації $g_{U,b_1}$, що показано в (1.3)

$$g_{U,b_1}(X) = g\left(UX + \beta_1\right) = \frac{1}{1+e^{-(UX+\beta_1)}} \tag{1.3}$$

Мережа стає SAE, коли кілька розріджених автокодерів складаються один з одним [14]. У SAE виходи розрідженого автокодера подаються на входи наступного розрідженого автокодера. Оптимальні вагові матриці та вектори зсуву для кожного розрідженого автокодера досягаються за допомогою навчання. Наприклад, перший прихований шар для першого розрідженого автокодера $G$ навчається на вхідних даних $X$ для отримання $U$, $U_r$, $b_1$, $b_{1r}$. Після навчання шар $G = \{g_1, g_2, ..., g_n\}$ кодує вхідні дані $X$ за допомогою $U$ та $b_1$. Потім закодовані значення використовуються як вхідні дані для навчання другого розрідженого автокодера для отримання оптимальних значень параметрів $V$, $V^{'}$ $b_2$, $b_2^{'}$, як показано на рис. 1.4, b. Цей процес продовжується до тих пір, поки не буде навчений останній розріджений автокодер і мережа не стане Deep Belief Network (DBN). Вихід останнього шару з останнього розрідженого автокодера, остаточне представлення вхідних даних, подається

на класифікатор. Нарешті, для покращення продуктивності моделі, всі шари SAE розглядаються як єдина модель і виконується тонка настройка, як показано на рис. 1.4, с.

## 1.3 Аналіз науково-технічної літератури

Проведемо аналіз науково-технічної літератури за трьома напрямками: безпека та виклики в SDN; мережеві системи виявлення вторгнень (NIDS), що розроблені з використанням машинного навчання та підходів на основі глибокого навчання (DL), що застосовують еталонний набір даних про вторгнення (NSL-KDD), а також різні системи виявлення DDoS, які реалізовані в середовищі SDN.

*Питання безпеки та виклики в SDN.*

У [15] розглянуто історія виникнення, архітектура SDN та її характеристики, а також інтерфейси API. Розглянуто питання безпеки та загрози, а також запропоновано шляхи їхнього вирішення.

У [16] представлено аналіз безпеки SDN, який забезпечує захист графічного інтерфейсу користувача на основі автентифікації, інтеграції SSL/TLS та послуг з ведення журналів/аудиту безпеки. Для шифрування даних та покращення безпеки середовища SDN запропоновано використовувати авторизацію FortNOX та шифри AES і DES.

В роботі [17] визначені основні атаки на безпеку в площинах SDN, включаючи площину додатків, площину управління та площину даних. Крім того, це дослідження також визначає підходи, що використовуються експертами та дослідниками для розробки рішень безпеки для площин SDN. В дослідженні представилено таксономію атак і запропонувано модель спільної безпеки після всебічної ідентифікації атак на площині SDN.

У дослідженні [18] проведено аналіз можливостей SDN для боротьби з атаками. Ця робота представляє таксономію, що підкреслює фундаментальні риси та внесок SDN як захисного механізму (SaaDM).

У [19] описано проблеми безпеки SDN, і запропоновано рішення, які

необхідно прийняти мережевому адміністратору для підвищення безпеки SDN.

У [20] надано огляд безпеки SDN, що включає різні вразливості та атаки на SDN. Мета роботи полягає в стимулюванні дослідників вирішувати проблеми безпеки, а також розглядаються рекомендації щодо безпеки SDN.

У [21] представлено поглиблений огляд основних проблем безпеки та відповідних рішень в рамках архітектури програмно-визначених мереж (SDN). Дослідження класифікує ці проблеми безпеки відповідно до методології моделювання загроз STRIDE.

*Системи виявлення вторгнень з використанням машинного навчання та глибокого навчання.*

У статті [22] представлено аналіз методів ML та DL, які порівнюються з точки зору їхньої точності та потенціалу виявлення різних типів вторгнень.

У роботі [23] пропонується емпіричне дослідження алгоритмів ML та DL для виявлення відомих та невідомих атак у мережах загального та спеціального призначення. Досліджено, як алгоритми ML та DL можуть навчатися на обмеженій кількості даних, зберігаючи при цьому високу точність.

У дослідженні [24] запропоновано несиметричний глибокий автокодер для задач виявлення мережевих вторгнень і проведено його аналіз. Перевірено надійність та ефективність запропонованого NIDS за допомогою еталонного набору даних, а саме KDD CUP'99. Запропонована система, яка може бути використана в дослідженнях мережевої безпеки, а також в системах виявлення та класифікації на основі DL-методу.

Робота [25] представляє аналіз технологій, протоколів, архітектури та загроз, пов'язаних з системами Інтернету речей, а також надає огляд моделей виявлення вторгнень, що виникають через скомпрометовані пристрої Інтернету речей. Ця робота також охоплює аналіз різних методів машинного навчання та глибокого навчання.

У статті [26] огляд стратегій та засобів захисту від атак на основі машинного навчання. Висвітлено різні типи атак, які можуть вплинути на IDS, та представлено стратегії захисту для зменшення або усунення впливу цих атак.

Метою дослідження [27] є створення NIDS, що поєднує алгоритми глибокого навчання та різноманітні типи даних для покращення ефективності виявлення хакерських атак та посилення мережевої безпеки. Це дослідження пропонує унікальний спосіб ефективного поєднання машинного навчання/глибокого навчання з новими наборами даних у NIDS. Новизна цього підходу полягає в його здатності виявляти складні взаємозв'язки між даними і моделями глибокого навчання, що дозволяє оптимізувати виявлення атак і підвищити гнучкість у боротьбі з новими загрозами. Використовуючи потенціал різноманітних наборів даних, цей підхід є важливим кроком вперед у підвищенні продуктивності та ефективності систем NIDS в реальних умовах.

У [28] проведено порівняння систем виявлення вторгнень, що використовують або систему зіставлення шаблонів, або машинне навчання для виявлення аномалій. У той час як підходи зіставлення шаблонів, як правило, страждають від високого рівня помилкових позитивних результатів (FPR), системи на основі машинного навчання, такі як SVM і KNN, передбачають потенційні атаки, розпізнаючи окремі особливості. Однак ці моделі часто працюють з обмеженим набором функцій, що призводить до нижчої точності та вищого FPR. У дослідженні представлена модель глибокого навчання, яка використовує переваги згорткової нейронної мережі (CNN) у поєднанні з двонаправленою мережею довгострокової короткочасної пам'яті LSTM (Bi-LSTM) для вивчення просторових і часових характеристик даних.

Дослідження [29] спрямоване на підвищення продуктивності систем виявлення вторгнень (IDS) шляхом використання потужності різних алгоритмів машинного навчання, а саме Random Forest, нейронних мереж і одновимірних згорткових нейронних мереж. В роботі ретельно оцінюється точність кожного алгоритму, і проводиться порівняльний аналіз для визначення оптимального підходу для побудови ефективної моделі IDS.

*Виявлення DDoS в середовищі SDN.*

Дослідження [30] присвячене вивченню взаємодії між SDN та IoT. У ньому досліджуються різні типи DDoS-атак і висвітлюються різні методи

захисту, виявлення та пом'якшення наслідків, що застосовуються для боротьби з DDoS-загрозами в мережах IoT на основі SDN (SDN-IoT).

У статті [31] запропоновано метод ентропії злиття, який виявляє атаки шляхом вимірювання випадковості мережевих подій. Перевагами цього методу є висока швидкість виявлення атак та очевидне зменшення значення ентропії. Ефективно використовується взаємодоповнюваність інформаційної ентропії та ентропії лог-енергії. Експериментальні результати показують, що значення ентропії сценаріїв атаки на 91,25% нижче, ніж нормальних сценаріїв, що має більшу перевагу і значущість у порівнянні з іншими методами виявлення атак.

У [32] пропонується метод навчання на основі дерева рішень для виявлення DDoS-атак в системах SCADA на основі SDN шляхом точного розрізнення нормального трафіку і трафіку DDoS-атак. Для навчання та тестування моделей навчання були отримані дані про нормальний трафік та трафік DDoS-атак у певній змодельованій експериментальній топології мережі. Для оптимізації продуктивності моделей на основі дерева рішень використано методи, що базуються на відборі ознак та налаштуванні гіперпараметрів. Експериментальні результати показують, що вибір ознак, комбінація різних моделей дерев рішень та налаштування гіперпараметрів можуть призвести до більш точної моделі машинного навчання з кращими показниками виявлення DDoS-атак на SCADA-системи на основі SDN.

Метою статті [33] є всебічний огляд приблизно 70 відомих механізмів, що використовуються для виявлення та пом'якшення наслідків розподілених атак типу «відмова в обслуговуванні» (DDoS) в мережах SDN. Ці механізми систематизовано в чотири основні групи, а саме: методи, засновані на теорії інформації, методи, засновані на машинному навчанні, підходи, засновані на штучних нейронних мережах (ШНМ), та інші методи.

У [34] представлено ефективно інтегровану SDN-платформу, яка вирішує недоліки попередніх рішень для захисту від DDoS-атак. Платформа забезпечує раннє та точне виявлення шаблонів DDoS-трафіку в хмарних середовищах на основі SDN. У цьому фреймворку застосовуються методи рекурсивного

усунення ознак (RFE), просторової кластеризації на основі щільності (DBSCAN), методи часових рядів, такі як авторегресивне інтегроване ковзне середнє (ARIMA), експоненту Ляпунова, експоненційний згладжуючий фільтр, динамічний поріг та класифікатор, заснований на правилах. Модель RDAER була оцінена на наборі даних CICDDoS 2019, досягнувши точності 99,92% і швидкого часу виявлення 20 секунд, що перевершує існуючі методи.

У роботі [35] проведено аналіз вплив атак типу DoS і DDoS на рівні даних і прикладному рівні. Показано, що конфлікт правил на основі потоку на прикладному рівні викликає загрозу безпеці.

## 1.4 Постановка наукової задачі

Отже, можна зробити висновок про існування на даний час наступних протиріч:

на практиці – між високими вимогами до інформаційної безпеки SDN і реальною безпекою обладнання мережі;

в теорії – між обмеженими можливостями відомих наукових результатів і необхідністю підвищення інформаційної безпеки SDN.

Тому актуальним є наукове завдання, яке направлене на вирішення вказаних вище протиріч і полягає в удосконалені мережевої системи виявлення вторгнень програмно-визначеної мережі на основі використання методів глибокого навчання.

Метою дослідження є удосконалення та оцінка ефективності системи виявлення DDoS-атак в програмно-визначених мережах на основі глибокого навчання.

Об’єктом дослідження є процес функціонування мережевої системи виявлення вторгнень.

Предметом дослідження є методи виявлення вторгнень на основі машинного навчання.

Відповідно до поставленої мети дослідження, сформульовану наукову

задачу доцільно розбити на ряд складових взаємопов’язаних частин, що визначають напрямки досліджень та черговість їх виконання:

1) аналіз програмно-визначеної мережі та глибокого навчання;
2) аналіз еталонного набору даних NSL-KDD;
3) оцінювання мережевої системи виявлення вторгнень на основі використання набору даних NSL-KDD;
4) оцінка впливу атак в програмно-визначених мережах на основі мережевої системи виявлення вторгнень;
5) розробка стартап проекту.

## Висновки до розділу 1

Розглянуто концепцію програмно-визначених мереж, основною особливістю яких є відокремлення площини керування від площини передачі даних. Це дозволяє значно підвищити гнучкість управління мережею, забезпечуючи централізоване керування трафіком за допомогою контролера.

Важливою перевагою SDN є здатність контролера зосереджувати інформацію про всі мережеві пристрої, що дозволяє вирішувати такі задачі, як маршрутизація і комутація, на централізованому рівні, що спрощує процеси адміністрування. Одним із ключових аспектів SDN є використання протоколу OpenFlow, який забезпечує стандартизоване управління потоками трафіку через різні мережеві пристрої, такі як комутатори та маршрутизатори, незалежно від їхнього виробника. Це дозволяє інтегрувати різнорідні апаратні платформи в єдину систему та налаштовувати правила роботи з потоками даних відповідно до заданих політик. Завдяки можливості оперативної зміни конфігурацій мережевих пристроїв і динамічному перенаправленню трафіку через спеціалізовані системи аналізу, SDN забезпечує високу ефективність в управлінні трафіком та моніторингу мереж. Отже, SDN є перспективною технологією для побудови високонадійних і масштабованих мереж, яка надає нові можливості для досліджень, оптимізації і управління трафіком,

забезпечуючи ефективне функціонування як у звичайних умовах, так і в умовах підвищеного навантаження.

Аналіз глибоких нейронних мереж демонструє їх значний потенціал у вирішенні складних завдань, що стосуються розпізнавання образів, обробки сигналів та генерації послідовних даних. Однією з ключових переваг глибоких нейронних мереж є здатність до автоматичного виділення ознак із необроблених даних без необхідності створення вручну ознак, що дозволяє скоротити трудовитрати та підвищити продуктивність систем. Ця особливість відкриває нові горизонти для застосування DL у різних галузях, таких як комп'ютерний зір, обробка мови та аналіз великих даних.

Загалом, глибокі нейронні мережі пропонують інноваційні рішення для складних задач розпізнавання і обробки даних. Застосування цих підходів значно розширює можливості сучасних інформаційних систем, дозволяючи досягати високої точності та продуктивності у задачах обробки великих масивів інформації.

Огляд наукової літератури демонструє стрімкий розвиток методів забезпечення безпеки SDN, особливо в контексті застосування інноваційних підходів машинного та глибокого навчання для виявлення та нейтралізації кіберзагроз, зокрема, DDoS-атак. Зокрема, було виявлено, що сучасні програмно-визначені мережі стикаються з низкою безпекових викликів, які вимагають розробки нових рішень для підвищення захисту. Ряд досліджень пропонує впровадження шифрування, багаторівневої аутентифікації та авторизації для зменшення ризиків. Протокол OpenFlow дозволяє ефективно керувати потоками трафіку, проте він також має певні вразливості, що вимагають подальших досліджень.

Системи виявлення вторгнень на основі алгоритмів машинного навчання та глибокого навчання значно підвищують ефективність захисту мережі. Ці системи можуть не лише виявляти вторгнення в режимі реального часу, але й успішно розпізнавати нові, раніше невідомі загрози. Особливо варто відзначити системи, що використовують еталонний набір даних NSL-KDD, який дозволяє

моделювати різні типи атак та оцінювати ефективність методів виявлення.

Для боротьби з DDoS-атаками в SDN-мережах застосовуються різні підходи, серед яких виділяються методи на основі інформаційної ентропії та машинного навчання. Дослідження показують, що використання гібридних підходів, таких як поєднання методів нейронних мереж і кластеризації, дозволяє досягти високої точності виявлення атак та значно скорочує час реакції на загрози.

Загалом, проведений аналіз свідчить про перспективність використання сучасних технологій штучного інтелекту в SDN для забезпечення безпеки мереж і боротьби з новими кіберзагрозами, що є ключовим фактором розвитку надійних та безпечних мереж 5G та IoT.

# 2 МЕРЕЖЕВА СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕНЬ

## 2.1 Мережева система виявлення вторгнень на основі використання набору даних NSL-KDD

Загальнодоступні еталонні набори даних дозволяють розробляти системи або моделі та порівнювати продуктивність моделей з іншими, подібних до еталонних наборів даних. Одним з найпопулярніших наборів даних є NSL-KDD, що широко використовується фахівцями з мережевої безпеки для розробки та оцінки NIDS. Він походить від відомого набору даних KDD Cup 99 [36] але був розроблений для подолання його суттєвих обмежень. У підрозділі пропонується система виявлення вторгнень для набору даних NSL-KDD з використанням підходу глибокого навчання, тобто самонавчання (STL) [37].

### 2.1.1 Аналіз набору даних NSL-KDD

Набір даних KDD Cup 99 був підготовлений [38] з використанням перехопленого мережевого трафіку в рамках програми оцінки IDS DARPA 1998 року [39]. Мережевий трафік було перехоплено в локальній мережі, яка імітувала реальну мережу бази ВПС США для звичайного трафіку. У тій самій локальній мережі були змодельовані різні атаки, щоб зібрати аномальний трафік, починаючи від відомих і закінчуючи ще не виявленими атаками. Набір даних KDD Cup 99 протягом багатьох років використовувався як еталонний набір даних для розробки та оцінки мережевої системи виявлення вторгнень. Недоліком цього набору даних є те, що він має велику кількість надлишкових записів у навчальних і тестових даних. Було виявлено, що навчальні та тестові дані мають майже 78% та 75% надлишкових записів відповідно. Розроблені на цьому наборі даних СППР стають упередженими до класифікації записів атак, що часто зустрічаються, і

дають погані результати класифікації менш частих, але шкідливих записів через цю надлишковість. Було виявлено, що більшість NIDS на основі машинного навчання змогли успішно класифікувати навчальні та тестові дані з найменшою точністю - 98% та 86% відповідно. У цьому випадку порівняння та оцінка різних NIDS стають складними, оскільки всі вони дають відмінний результат на цьому наборі даних. NSL-KDD з'явився для того, щоб подолати обмеження набору даних KDD Cup 99. Цей набір даних є похідним від набору даних KDD Cup 99. Для покращення набору даних KDD Cup 99 у NSL-KDD було застосовано наступні підходи. По-перше, з набору даних KDD Cup 99 було видалено всі надлишкові записи з тренувальних та тестових даних. По-друге, всі записи, що залишилися в наборі даних KDD Cup 99, були розділені на 21 набір на основі кількості алгоритмів навчання, які можуть правильно класифікувати записи.

У табл. 2.1 показано неперервні ознаки в наборі даних з їхніми типами. Номінальні та бінарні ознаки показані в табл. 2.2 разом з їх типами та кількістю похідних ознак після кодування 1 до n. У наборі даних виявлено чотири типи атак: DDoS, User-to-Root (U2R), Root-to-Local (R2L) та зондування. DDoS-атаки використовуються для перевантаження системних ресурсів хостів небажаними запитами, щоб вони не могли обробляти легітимні запити.

В U2R-атаці користувач, який не є root-користувачем, використовує вразливості системи, щоб отримати доступ від імені root-користувача і завдати шкоди системі.

При R2L-атаці зловмисники використовують вразливості системи, щоб отримати доступ до хосту віддалено. Зондування використовується для виявлення мережевої інформації, такої як сервіси та топологія, що допомагає запустити атаку.

У табл. 2.3 показано різні атаки в кожній категорії. Навчальні дані містять 23 класи трафіку, включаючи 22 атаки і один нормальний клас. Тестові дані

містять 38 класів трафіку, включаючи 21 атаку, подібну до тієї, що була виявлена в навчальних даних, 16 нових атак і один нормальний клас. Всі ці атаки поділяються на чотири категорії, як описано вище. Розподіл записів у навчальних і тестових даних для нормального трафіку і трафіку з різними атаками показано у табл. 2.4.

Таблиця 2.1 − Неперервні ознаки та їх типи в наборах даних KDD Cup 99 та NSL-KDD [40]

| Характеристика | Тип | Характеристика | Тип |
|---|---|---|---|
| duration | basic | count | traffic |
| src_bytes | basic | srv_count | traffic |
| dst_bytes | basic | serror_rate | traffic |
| wrong_fragment | basic | srv_serror_rate | traffic |
| urgent | basic | rerror_rate | traffic |
| hot | content | srv_rerror_rate | traffic |
| num_failed_logins | content | same_srv_rate | traffic |
| num_compromised | content | diff_srv_rate | traffic |
| root_shell | content | srv_diff_host_rate | traffic |
| su_attempted | content | dst_host_count | traffic |
| num_root | content | dst_host_srv_count | traffic |
| num_file_creations | content | dst_host_same_srv_rate | traffic |
| num_shells | content | dst_host_diff_srv_rate | traffic |
| num_access_files | content | dst_host_same_src_port_rate | traffic |
| num_outbound_cmds | content | dst_host_srv_diff_host_rate | traffic |
| dst_host_serror_rate | traffic | dst_host_srv_serror_rate | traffic |
| dst_host_rerror_rate | traffic | dst_host_srv_rerror_rate | traffic |

Запис зберігався в наборі, номер якого дорівнював кількості класифікаторів, які точно класифікували цей запис. Записи були відібрані з кожного набору у частці, обернено пропорційній частці записів у цьому наборі від загальної кількості записів у всіх наборах. Така багатокрокова обробка набору даних KDD Cup зробила кількість записів у наборі даних NSL-KDD прийнятною для навчання різних методів машинного навчання [36].

Таблиця 2.2 − Номінальні та бінарні ознаки в наборах даних KDD Cup 99 та NSL-KDD [40] разом з їхніми типами та кодуванням 1-to-n

| Характеристика | Тип | Кодування 1-to-n |
|---|---|---|
| protocol_type | basic | 3 |
| service | basic | 70 |
| flag | basic | 11 |
| land | basic | 1 |
| logged_in | content | 1 |
| hline is host login | content | 1 |
| is guest login | content | 1 |

Таблиця 2.3 − Види атак та різні атаки кожного типу в наборі даних NSL-KDD [36]

| Вид атаки | Атаки |
|---|---|
| Denial-of-Service | back, land, neptune, pod, smurf, teardrop, apache2, mailbomb, processtable, udpstorm |
| User-to-root | buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, xterm |
| Root-to-local | ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, httptunnel, named, sendmail, worm, xlock, xsnoop, snmpgetattack, snmpguess |
| Probe | ipsweep, nmap, portsweep, satan, mscan, saint |

Таблиця 2.4 − Розподіл записів у наборі даних NSL-KDD [36]

| Вид трафіку | | Навчання | Тестування |
|---|---|---|---|
| Нормальний | | 67343 | 9711 |
| Атака | DoS | 45927 | 7458 |
| | U2R | 52 | 67 |
| | R2L | 995 | 2887 |
| | Probe | 11656 | 2421 |

Кожен запис у наборі даних NSL-KDD має 41 ознаку, включаючи три номінальні, чотири бінарні та 34 неперервні ознаки, а також мітку для нормального або особливого типу атаки. Ці ознаки поділяються на три різні типи:

основні характеристики отримуються безпосередньо з TCP/IP-з'єднання;

характеристики трафіку накопичуються за часовий інтервал, наприклад, дві секунди, для одного і того ж хоста або сервісу;

характеристики вмісту витягуються з даних прикладного рівня з'єднання.

### 2.1.2 Оцінювання мережевої системи виявлення вторгнень на основі використання набору даних NSL-KDD

Існує два підходи до оцінювання NIDS, розроблених для набору даних NSL-KDD [41]. У найбільш поширеному підході навчальні дані використовуються як для навчання, так і для оцінювання. Оцінювання виконується або з використанням *n*-кратної перехресної перевірки, або навчальні дані розбиваються на навчальні, перехресні та тестові набори даних. NIDS, розроблені за допомогою цього підходу, мають високу точність і низький рівень хибних спрацьовувань.

Другий підхід використовує навчальні та тестові набори даних окремо для цілей навчання та тестування. Оскільки навчальні та тестові дані походять з різних розподілів, точність, досягнута в цьому підході, не така висока, як у першому підході.

Тому в роботі доцільно застосовувати другий підхід для реалістичної оцінки NIDS. Але, для повноти отриманих результатів також представлено результати для першого підходу.

На рис. 2.1 наведено етапи реалізації NIDS. Набір даних NSL-KDD містить різні типи ознак з різними значеннями (2.1.1). Важливо попередньо обробити набір даних, перш ніж використовувати його для вхідних даних у самонавчанні, як показано на рис. 2.2.

Для перетворення номінальних ознак у дискретні використовується кодування 1 до n. Ми знайшли в наборі даних ознаку, значення якої залишається рівним 0 для всіх доступних записів у навчальних і тестових даних. Ця ознака була видалена з набору даних, оскільки її постійне значення у всіх записах робить її незначущою. Кількість ознак стає 121 після виконання вищезгаданих кроків.


Рисунок 2.1 − Етапи впровадження NIDS

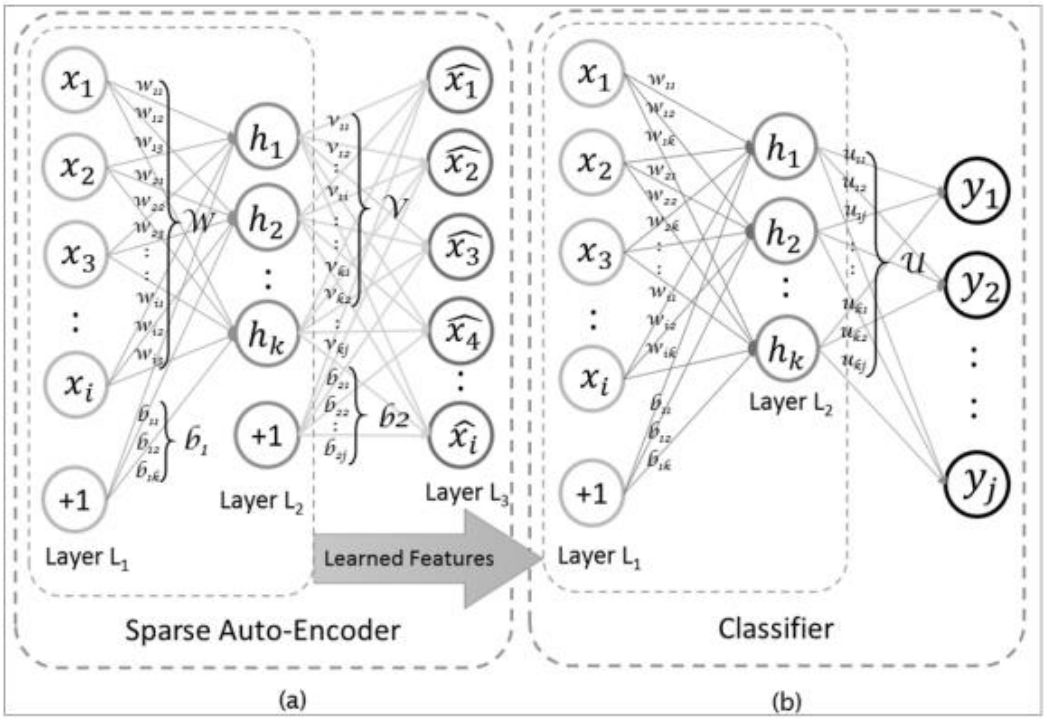


Рисунок 2.2 − Реалізація самонавчання

Значення у вихідному шарі на етапі навчання ознак, показаному на рис. 2.2 а, обчислюються за допомогою функції Sigmoid, яка генерує значення в діапазоні від 0 до 1. Значення вихідного шару на цьому етапі є наближенням до значень вхідного шару. Тому ми нормалізуємо значення ознак у вхідному шарі в діапазоні від 0 до 1, використовуючи max-min нормалізацію. Ми використовуємо перетворені навчальні дані NSL-KDD без міток для некерованого навчання за допомогою розрідженого автокодера на першому етапі самонавчання. На другому етапі ми перетворюємо навчальні дані на нові ознаки, використовуючи параметри, отримані на першому етапі. Ці нові ознаки передаються до м'якої максимальної регресії (SMR) для класифікації. У запропонованій реалізації використовується те ж саме джерело даних, тобто навчальні дані NSL-KDD, як немічені та мічені дані для навчання ознак та навчання класифікатора відповідно.

Для оцінки ефективності класифікаторів використовуються наступні метрики:

1. Точність (P). Визначається як відсоткове співвідношення точно передбачених записів до всіх передбачених записів і розраховується як відсоткове співвідношення кількості істинних спрацьовувань (TP) до суми істинних спрацьовувань (TP) і хибних спрацьовувань (FP) для класифікованих записів.

$$P = \frac{TP}{(TP + FP)} \times 100\%$$

2. Повнота (R). Визначається як відсоткове співвідношення точно передбачених записів до всіх наявних записів для певного класу в наборі даних і розраховується як відсоткове співвідношення кількості істинних спрацьовувань (TP) до суми істинних спрацьовувань (TP) і хибнонегативних спрацьовувань (FN) для класифікованих записів.

$$R = \frac{TP}{(TP + FN)} \times 100\%$$

3. F-міра (F). Дає цілісну оцінку моделі за точністю та повнотою. Розраховується як середнє значення точності та повноти.

$$F = \frac{2 \times P \times R}{(P + R)}$$

### 2.1.3 Оцінка ефективності мережевої системи виявлення вторгнень

NIDS було досліджено для трьох різних типів класифікації на основі категоризації атак. У першому типі кожен запис класифікується як один з двох класів (2-клас) - нормальний або аномальний, при цьому всі атаки вважаються аномальними. У другому типі кожен запис відноситься до одного з п'яти класів (5-класів), включаючи нормальний і чотири різні типи атак, шляхом ідентифікації атаки з її типом. В останньому типі кожен запис класифікується як один з 23 класів, включаючи нормальний або одну з 22 атак, доступних в навчальному наборі даних. Ми виміряли значення точності, повноти та F-міри для атак у випадку 2-класової та 5-класової класифікації в оцінці тестового набору даних. Для 5-класної класифікації ми обчислили зважені значення цих метрик.

Оцінка точності класифікації самонавчального навчання (STL) проходила на навчальних даних, використовуючи 10-кратну перехресну перевірку для 2-класів, 5-класів та 23 класів. Порівнювалась ефективність STL з м'якою максимальною регресією (SMR), яка застосовувалася до набору даних без будь-якого навчання на основі ознак. Виявлено, що STL показав кращі результати для 2-класової класифікації порівняно з SMR (рис. 2.3). Однак, результати STL та SMR дуже схожі для 5-класової та 23-класової класифікації. STL досягнув понад 98% точності класифікації для всіх типів класифікації. Проведена оцінка значення точності, повноти та F-міри для 2-класової класифікації на навчальному наборі даних. Оцінюючи NIDS за допомогою 10-кратної перехресної перевірки, деякі типи записів про атаки можуть бути пропущені на етапі навчання або оцінки 5-класової та 23-класової класифікації. Тому

розглянемо ці метрики для 2-класової класифікації. Виявлено, що STL досягнув кращих результатів за всіма цими метриками порівняно з SMR. STL досяг значення 98,84% для F-міри, тоді як SMR досяг 96,79%, як показано на рис. 2.3.

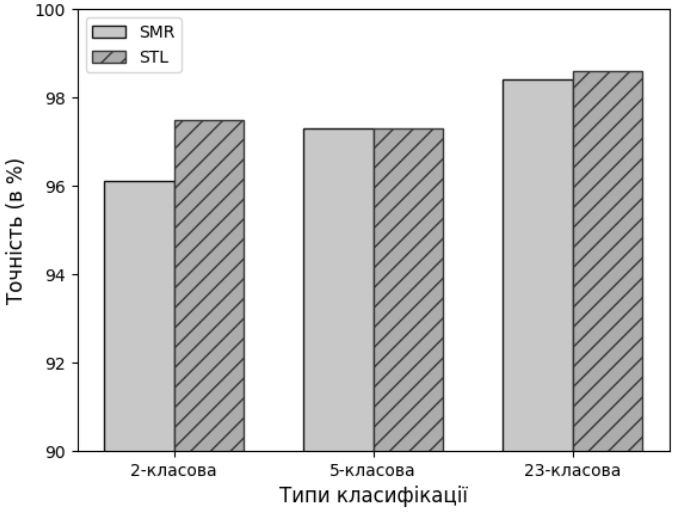


Рисунок 2.3 − Досягнута точність класифікації для 2-класу, 5-класу та 23-класу на навчальних даних з використанням STL та SMR

Продуктивність STL оцінюється для 2-класу та 5-класу за допомогою тестового набору даних. Визначено, що STL працює краще порівняно з SMR, як показано на рис. 2.4. STL досягнув значення точності 88,39%, тоді як SMR досягнув 78,06% для 2-класової класифікації. Точність, отримана за допомогою STL для 2-класної класифікації, перевищує багато попередніх результатів.

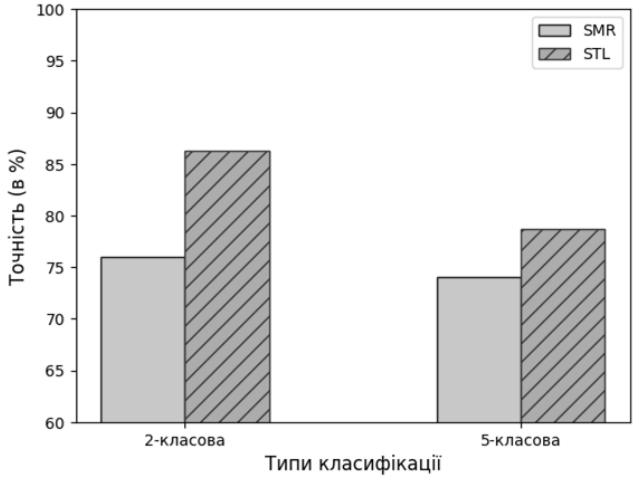


Рисунок 2.4 − Досягнута точність класифікації для 2-го та 5-го класів на тестових даних з використанням STL та SMR

Найкращий показник точності, про який повідомлялося в [36], становив 82% для NB-Tree. STL досягнув значення точності 79,10%, тоді як SMR досягнув 75,23% для 5-класової класифікації. На рис. 2.5 і рис. 2.6 показано значення точності, повноти і F-міри для 2-класової і 5-класової класифікації.

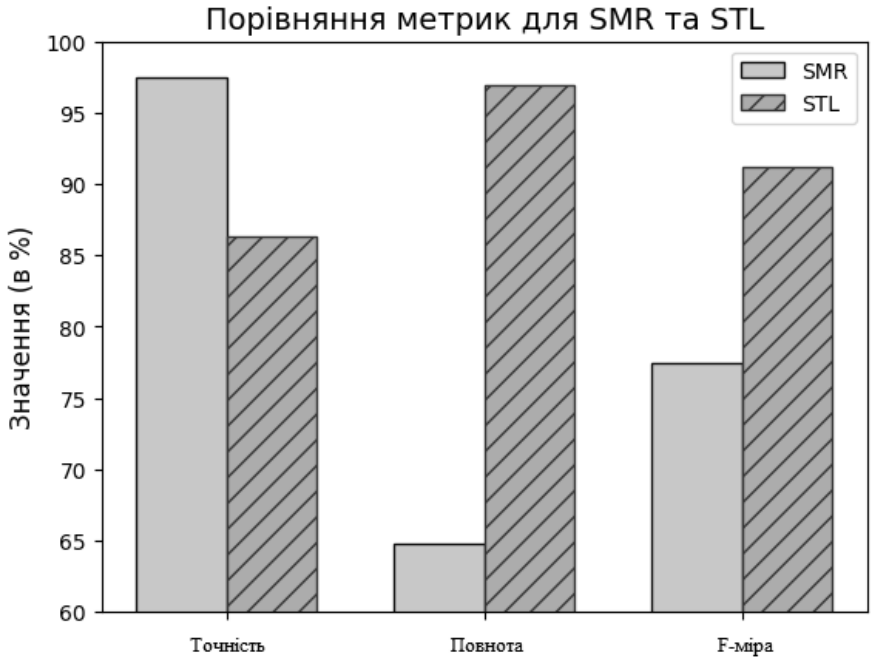


Рисунок 2.5 − Значення точності, повноти та F- міри, досягнуті для 2-го класу за даними тестування з використанням STL та SMR

Точність, досягнута в STL, є нижчою порівняно з SMR для класифікації 2-класів. Значення точності для STL та SMR становлять 85,44% та 96,56% відповідно. Однак, STL досягнув кращих значень для повноти порівняно з SMR. Показники повноти для STL і SMR становлять 95,95% і 63,73%, відповідно. STL перевищив SMR за значенням F-міри завдяки високому значенню повноти. STL отримав значення 90,4% для F-міри, тоді як SMR - 76,8%. Ми спостерігали подібні закономірності для 5-класної класифікації, показані на рис. 2.6. Значення F-міри для STL та SMR становлять 75,76% та 72,14% відповідно.

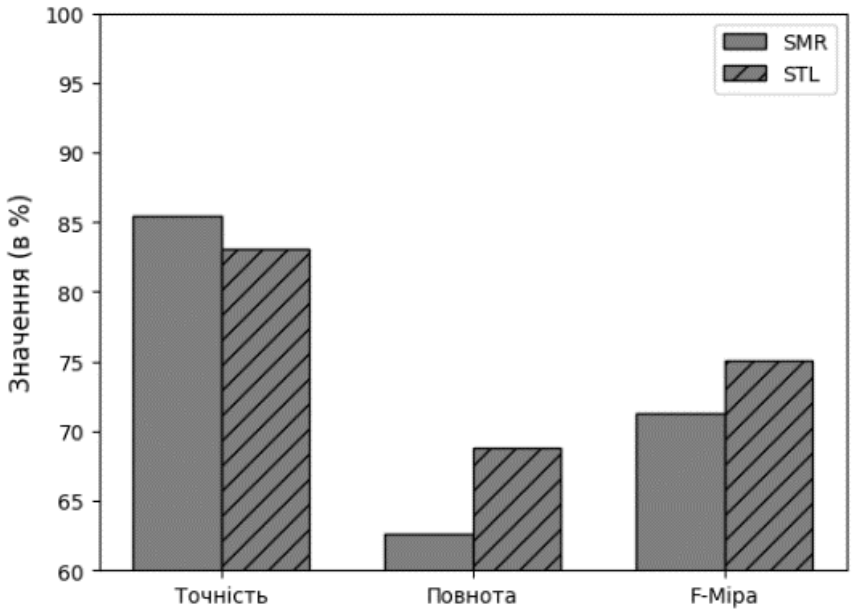


Рисунок 2.6 − Значення точності, повноти та F- міри, досягнуті для 5-го класу за даними тестування з використанням STL та SMR

Отже, було проаналізовано основні аспекти та переваги використання набору даних NSL-KDD для мережевої системи виявлення вторгнень. NSL-KDD є основним інструментом для оцінки продуктивності NIDS завдяки збалансованості, універсальності та реалістичності представлених загроз. Набір даних містить різноманітні типи атак, що дозволяє моделювати реальні сценарії мережевих вторгнень та випробовувати системи на здатність виявляти різноманітні аномалії.

## 2.2 Мережева система виявлення вторгнень на основі глибокого навчання в SDN

Атаки в SDN можуть відбуватися як на площині даних, так і на площині управління. Атаки на площині даних дуже схожі на атаки, які відбуваються в традиційній мережі і націлені на сегмент мережі або кілька хостів. На відміну від цього, весь мережевий трафік може бути порушений, якщо атака відбувається на площині управління SDN. Один з підходів, який використовується в атаках на площині управління, полягає у виявленні потоків

трафіку, для яких контролер обробляє всі пакети потоків без встановлення будь-яких правил в таблицях потоків комутаторів. Після цього генеруються пакети для цих потоків в мережі. Системні ресурси контролера та комутаторів інтенсивно витрачаються на обробку цих пакетів та їх буферизацію відповідно.

У цьому параграфі представимо реалізацію NIDS на основі DL в якості мережевого додатку в SDN для виявлення багатовекторних DDoS-атак в обох площинах.

### 2.2.1 DDoS-атаки в мережі

Існують різні види DDoS-атак, які зловмисники можуть запускати в мережі, включаючи протокольний та прикладний рівні [42]. Під час атаки об'ємного впливу відбувається перевантаження системи або мережевого каналу шляхом генерування великого об'єму трафіку або запитів. Її мета — викликати збій через перевищення пропускної здатності або ресурсів. Такий трафік споживає значну частину пропускної здатності мережі і блокує зв'язок між атакованим клієнтом і легітимними хостами із зовнішніх мереж. Зловмисники або боти використовують UDP і ICMP ping-flood для запуску прямих або віддзеркалених DDoS-атак, як показано на рис. 2.7. У прямих DDoS-атаках зловмисники надсилають трафік безпосередньо клієнтам з підроблених IP-адрес. В той час як у віддзеркалених DDoS-атаках зловмисники надсилають трафік у вигляді запитів на велику кількість відкритих загальнодоступних серверів, підміняючи ідентифікатори хостів-жертв. Публічні сервери перевантажують хости жертви великим обсягом трафіку, що генерується у відповідь на запити. Вплив атаки посилюється у випадку посилення DNS і NTP, коли повідомлення-відповіді мають у 100 разів більше корисного навантаження, ніж запити. При DDoS-атаках на основі протоколів виснажуються системні ресурси додатково до споживання пропускної здатності каналу зв'язку. Наприклад, зловмисники переповнюють чергу з'єднань TCP-сервера напіввідкритими незавершеними з'єднаннями і роблять його нездатним

приймати нові з'єднання від легітимних клієнтів за допомогою TCP SYN flood-атаки. Атаки на основі додатків націлені на протоколи прикладного рівня, такі як HTTP і HTTPS. Ці атаки запускаються на низькій швидкості і виглядають як звичайний трафік з точки зору мережі та транспортного рівня. Сценарії атак також змінилися, і тепер атаки запускаються в комбінації двох або більше видів DDoS-атак, що робить завдання виявлення складним і трудомістким.

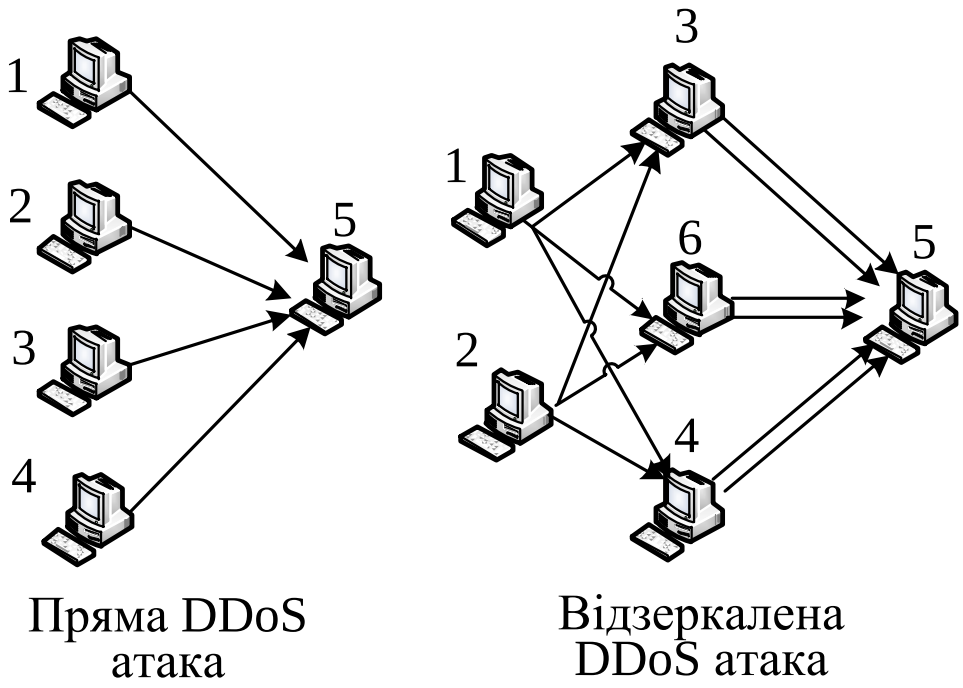


Рисунок 2.7 − Різні техніки для запуску DDoS-атаки

### 2.2.2 Впровадження NIDS

У досліджуваній системі виявлення вторгнень DDoS-атаки ідентифікуються на мережевому та транспортному рівнях за припущенням, що зловмисники підробляють свої IP-адреси під час запуску атак, щоб уникнути відслідковування. Система NIDS (рис. 2.8) складається з:

інсталятор збору та обробки даних про трафік (модуль TCFI);

аналізатор ознак (модуль FE);

класифікатор трафіку (модуль TC).

Для забезпечення оперативного виявлення атак та мінімізації помилкових спрацьовувань обчислення потоку виконується на основі пакетів для виявлення атак без використання додаткових протоколів агрегації, таких як netflow [43] і sFlow [44].

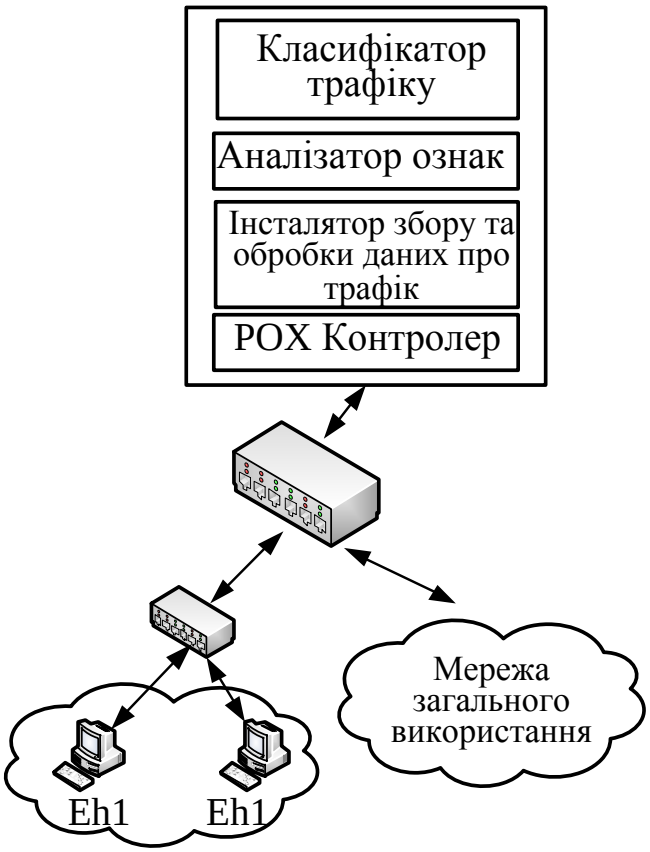


Рисунок 2.8 − NIDS з компонентами, впровадженими в SDN для виявлення DDoS-атак

Модуль TCFI виконується паралельно з модулями FE і TC, які викликаються функцією таймера. Отже, використовуються дві різні реалізації модуля TCFI. В одній реалізації контролер дає вказівку комутаторам пересилати йому всі пакети, які вони отримують з мережі. З іншого боку, друга реалізація вказує комутаторам пересилати обмежену кількість потоків і пакетів. Розглянемо першу реалізацію. Модуль TCFI перевіряє тип і код повідомлення OF, витягнутого з отриманого пакета PACKET IN на контролері. Тип і код повідомлення описують причину надходження пакета. Пакет може надійти на контролер від комутатора через подію пропуску в його таблицях потоків або переадресацію за правилом потоку на порт, пов'язаний з контролером і

призначеними фізичними портами. TCFI витягує з пакета різні заголовки мережевого і транспортного рівнів, щоб ідентифікувати потік, до якого він належить. Потік визначається як потік пакетів з однаковими значеннями типу протоколу, IP-адреси джерела та призначення, а також номерами портів джерела та призначення в трафіку TCP або UDP. Потік ICMP схожий на потік TCP або UDP, за винятком того, що він має тип і код ICMP-повідомлення замість номерів портів. Модуль TCFI витягує ще кілька заголовків пакетів, які використовуються модулем FE для вилучення ознак з потоків трафіку. Всі ці витягнуті заголовки зберігаються у списку пакетів для кожного вхідного пакета на контролері.

У табл. 2.5 показано заголовки пакетів, що визначені інсталятором збору та обробки даних для трафіку TCP, UDP та ICMP

Таблиця 2.5 − Заголовки, що визначені з пакетів TCP, UDP та ICMP

| TCP | | UDP | ICMP |
|---|---|---|---|
| Src IP | Window | Src IP | Src IP |
| Dst IP | SYN | Dst IP | Dst Port |
| Src Port | ACK | Src Port | ICMP Type |
| Dst Port | URG | Dst Port | ICMP Code |
| Protocol | FIN | Protocol | Protocol |
| Data Size | RST | Data Size | Data Size |
| TTL | PUSH | TTL | TTL |

Інсталятор збору та обробки даних про трафік виконує вищезгадане завдання, коли пакет надходить до контролера внаслідок переадресації за правилом потоку, вже встановленим у таблицях потоків. Наступні додаткові завдання виконуються, коли пакет надходить через подію пропуску таблиці потоків у комутаторі. По-перше, TCFI шукає симетричний потік, що відповідає потоку прибулого пакета, у списку потоків. Цей список потоків заповнюється TCFI всіма унікальними потоками, що досягають мережі протягом певного часового вікна.

Потоки на основі TCP або UDP визначаються як симетричні потоки, якщо вони належать до одного протоколу, а IP-адреса джерела та номер порту в

одному потоці мають ті ж значення, що й IP-адреса призначення та номер порту в іншому потоці, і навпаки. Потоки типу запиту і відповіді стають симетричними, якщо вони відбуваються в певному інтервалі для трафіку на основі ICMP. Якщо для вхідного потоку знайдено симетричний потік, то модуль TCFI видаляє симетричний потік зі списку потоків і встановлює два правила потоку, одне для вхідного потоку, а інше для симетричного потоку всередині таблиць потоків разом з пересиланням поточного пакету за допомогою повідомлення PACKET OUT. Наступні пакети цих потоків перенаправляються на контролер і призначені фізичні порти відповідно до встановлених правил. Встановлення правил потоку лише для симетричних потоків ґрунтується на припущенні, що зловмисники зазвичай підміняють свої IP-адреси, щоб уникнути відстеження та отримання трафіку у відповідь від жертв. Тому атакуючий трафік здебільшого асиметричний. Правила потоку встановлюються тільки для симетричних потоків, щоб уникнути перенасичення таблиці потоків через ці асиметричні потоки. Якщо для вхідного пакета не знайдено симетричного потоку, TCFI перевіряє список потоків, щоб знайти, чи існує потік для вхідного пакета в списку потоків. Якщо такий потік знайдено, TCFI пересилає пакет з комутаторів за допомогою PACKET OUT, не встановлюючи для нього жодних правил.

В іншому випадку потік для пакета додається до списку потоків, а потім пакет пересилається, як і в попередньому випадку.

Алгоритм реалізації TCFI наведено у [45]. Алгоритм TCFI відрізняється від алгоритму детектора максимальної ентропії [46]:

оцінюються характеристики різних видів потоків, що дозволяє виявляти більш тонкі аномалії в поведінці мережі;

контролер отримує пакет через подію пропуску в таблиці потоків, або через переадресацію за правилом потоку, встановленим у комутаторі TCFI;

заголовки пакетів зберігаються у списку пакетів для кожного пакета, що надходить до контролера.

Аналізатор ознак (модуль FE) викликається NIDS за допомогою функції таймера. Модуль FE зчитує заголовки пакетів зі списку пакетів, який був заповнений TCFI. Ознаки витягуються з заголовків для певного часового вікна. FE скидає список пакетів і список потоків, щоб зберегти заголовки і потоки для наступного часового вікна. Перелік 68 ознак, вилучених модулем FE для TCP, UDP та ICMP, показано у табл. 2.6 табл. 2.7 та табл. 2.8. Для трафіку TCP, UDP та ICMP модуль FE виділив 34, 20 та 14 ознак відповідно. Ці ознаки були зведені до оптимального набору ознак за допомогою підходу DL на основі SAE для класифікації. Аналізатор ознак виокремлює ці ознаки для всіх хостів у мережі, які мають вхідний трафік протягом спостережуваного часового вікна. Проведене дослідження враховує всі пакети, що переглядаються на контролері, ознаки виділяються шляхом групування їх у потоках TCP, UDP або ICMP. Аналізатор ознак обчислює медіану кількості пакетів і байт у потоці для ознак № 9-12, 43-46 і 63-67. Також обчислює ентропію H(F) для характеристик елементів № 8, 14, 16, 18, 20, 42, 48, 50, 54, 62 і 68. Ентропія визначається наступним чином:

$$H(F) = -\sum_{i=1}^{n} \frac{f_i}{\sum_{j=1}^{n} f_j} \times \log_2 \frac{f_i}{\sum_{j=1}^{n} f_j}$$

де множина $F = \{f_1, f_2, ..., f_n\}$ представляє частоту кожного окремого заголовка для потоків TCP, UDP та ICMP. NIDS викликає модуль класифікатор трафіку після того, як аналізатор ознак завершує роботу. Модуль класифікатора трафіку реалізований за допомогою SAE і класифікує трафік для хоста в один з восьми класів, включаючи звичайний і сім типів класів DDoS-атак. Ці DDoS-атаки базуються на протоколах TCP, UDP або ICMP, що запускаються зловмисниками окремо або в комбінації двох чи трьох з них.

Таблиця 2.6 − Характеристики, що визначені аналізатором ознак для потоків TCP

| | Опис ознаки |
|---|---|
| 1 | # of incoming TCP flows |
| 2 | Ratio of TCP flows over total incoming flows |
| 3 | # of outgoing TCP flows |
| 4 | Ratio of TCP flows over total outgoing flows |
| 5 | Ratio of symmetric incoming TCP flows |
| 6 | Ratio of asymmetric incoming TCP flows |
| 7 | # of distinct src IP for incoming TCP flows |
| 8 | Entropy of src IP for incoming TCP flows |
| 9 | Bytes per incoming TCP flow |
| 10 | Bytes per outgoing TCP flow |
| 11 | # of packets per incoming TCP flow |
| 12 | # of packets per outgoing TCP flow |
| 13 | # of distinct window size for incoming TCP flows |
| 14 | Entropy of window size for incoming TCP flows |
| 15 | # of distinct TTL values for incoming TCP flows |
| 16 | Entropy of TTL values for incoming TCP flows |
| 17 | # of distinct src ports for incoming TCP flows |
| 18 | Entropy of src port for incoming TCP flows |
| 19 | # of distinct dst ports for incoming TCP flows |
| 20 | Entropy of dst ports for incoming TCP flows |
| 21 | Ratio of dst ports ≤ 1024 for incoming TCP flows |
| 22 | Ratio of dst port > 1024 for incoming TCP flows |
| 23 | Ratio of TCP incoming flows with SYN flag set |
| 24 | Ratio of TCP outgoing flows with SYN flag set |
| 25 | Ratio of TCP incoming flows with ACK flag set |
| 26 | Ratio of TCP outgoing flows with ACK flag set |
| 27 | Ratio of TCP incoming flows with URG flag set |
| 28 | Ratio of TCP outgoing flows with URG flag set |
| 29 | Ratio of TCP incoming flows with FIN flag set |
| 30 | Ratio of TCP outgoing flows with FIN flag set |
| 31 | Ratio of TCP incoming flows with RST flag set |
| 32 | Ratio of TCP outgoing flows with RST flag set |
| 33 | Ratio of TCP incoming flows with PUSH flag set |
| 34 | Ratio of TCP outgoing flows with PUSH flag set |

Таблиця 2.7 − Характеристики, що визначені аналізатором ознак для потоків UDP

| | Опис ознаки |
|---|---|
| 35 | # of incoming UDP flows |
| 36 | Ratio of UDP flows over total incoming flows |
| 37 | # of outgoing UDP flows |
| 38 | Ratio of UDP flows over total outgoing flows |
| 39 | Ratio of symmetric incoming UDP flows |
| 40 | Ratio of asymmetric incoming UDP flows |
| 41 | # of distinct src IP for incoming UDP flows |
| 42 | Entropy of src IP for incoming UDP flows |
| 43 | Bytes per incoming UDP flow |
| 44 | Bytes per outgoing UDP flow |
| 45 | # of packets per incoming UDP flow |
| 46 | # of packets per outgoing UDP flow |
| 47 | # of distinct src ports for incoming UDP flows |
| 48 | Entropy of src ports for incoming UDP flows |
| 49 | # of distinct dst ports for incoming UDP flows |
| 50 | Entropy of dst ports for incoming UDP flows |
| 51 | Ratio of dst ports ≤ 1024 for incoming UDP flows |
| 52 | Ratio of dst port > 1024 for incoming UDP flows |
| 53 | # of distinct TTL values for incoming UDP flows |
| 54 | Entropy of TTL values for incoming UDP flows |

Таблиця 2.8 − Характеристики, що визначені аналізатором ознак для потоків ICMP

| | Опис ознаки |
|---|---|
| 55 | # of incoming ICMP flows |
| 56 | Ratio of ICMP flows over total incoming flows |
| 57 | # of outgoing ICMP flows |
| 58 | Ratio of ICMP flows over total outgoing flows |
| 59 | Ratio of symmetric incoming ICMP flows |
| 60 | # of asymmetric incoming ICMP flows |
| 61 | # of distinct src IP for incoming ICMP flows |
| 62 | Entropy of src IP for incoming ICMP flows |
| 63 | Bytes per incoming ICMP flow |
| 64 | Bytes per outgoing ICMP flow |
| 65 | # of packets per incoming ICMP flow |
| 66 | # of packets per outgoing ICMP flow |
| 67 | # of distinct TTL values for incoming ICMP flows |
| 68 | Entropy of TTL values for incoming ICMP flows |

## 2.3 Аналіз характеристик NIDS

Схема бездротової мережі для проведення дослідження характеристик NIDS представлена на рис. 2.9.

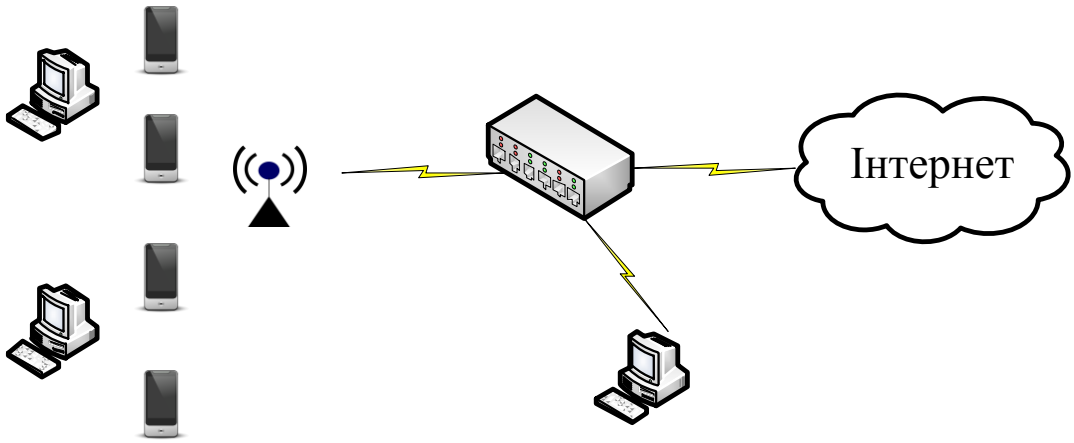


Рисунок 2.9 − Схема бездротової мережі

Моделі використання мережі різнилися для кожного користувача, що спричинило варіативність у розподілі трафіку. Дослідження проводилось протягом трьох днів. Трафік мережі збирався в системі Linux за допомогою інструментів tcpdump [47] та дзеркалювання портів. Перші два дні трафік зберігався і використовувався як звичайний, тоді як трафік третього дня було змішано з трафіком атаки, зібраним окремо та позначеним як шкідливий. Було створенні умови, коли в мережі (вузлах) одночасно передається звичайний та шкідливий трафік, що значно ускладнює виявлення атак для систем виявлення вторгнень. Змішування трафіку виконувалося за допомогою інструменту bit-twist [48], який дозволяє модифікувати заголовки пакетів у файлах трасування.

Об’єднані файли трасування з нормальним та атакувальним трафіком були відтворені разом за допомогою tcpreplay [49].

Зібраний нормальний трафік складався з даних веб-сервісів, потокового мовлення, месенджерів та ігор. Інструмент атаки hping3 [50] використовувався для запуску DDoS-атак з різним розміром та частотою пакетів. За один раз запускався лише один тип атаки для зручності маркування та вилучення

функцій. Хост-система використовувала tcpreplay для відтворення траси атаки та нормального трафіку по черзі. Характеристики, що були визначені аналізатором ознак для кожного часового вікна, були збережені у файлах наборів даних у форматі .CSV для навчання класифікатора трафіку. Інтервал для запуску аналізатора ознак був встановлений на 60 секунд [51]. Файли з наборами даних були розділені на навчальні та тестові. Розподіл записів у наборах даних показано в табл. 2.9.

Таблиця 2.9 − Кількість записів у навчальному та тестовому наборах даних для нормального та різного трафіку DDoS-атак

| Тип трафіку | | Кількість записів | |
|---|---|---|---|
| | | Навчання | Тестування |
| Нормальний (N) | | 49179 | 21076 |
| Атака | TCP (T) | 5471 | 2344 |
| | UDP (U) | 5273 | 2260 |
| | ICMP (I) | 1602 | 686 |
| | TCP&UDP (TU) | 4692 | 2011 |
| | TCP&ICMP (TI) | 4739 | 2031 |
| | UDP&ICMP (UI) | 4437 | 1902 |
| | All (A) | 5615 | 2407 |

Характеристики трафіку в наборах даних були реальними значеннями з різними діапазонами в наборі даних. Для їх нормалізації в діапазоні [0, 1] перед передачею на навчання використовувалася нормалізація максиміна:

$$X_{norm} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \forall x_i \in X$$

Продуктивність системи було оцінено на наборах даних, наведених у табл. 2.9.

Для обчислення точності, повноти та F-міри було використано матрицю помилок (M). Це квадратна матриця N × N, де N - кількість класів у наборі даних. Стовпчик матриці представляє передбачення певного класу для всіх наявних класів, включаючи сам клас у наборі даних. Аналогічно, кожен рядок представляє передбачення всіх наявних класів для певного класу, включаючи цей клас. Діагональні елементи матриці є істинно-позитивними (ІП)

значеннями для кожного класу. Сума елементів вздовж рядка, за винятком діагонального елемента, представляє прогноз як хибно-позитивний (ХП) для класу, що відповідає рядку. Сума елементів у стовпчику, за винятком діагонального елемента, представляє прогноз як хибно-негативний (ХН) для класу, що відповідає стовпчику. Різні параметри ефективності можуть бути визначені наступним чином:

Точність (A) це відсоток точно класифікованих записів ( $N_i$ ) від загальної кількості записів у наборі даних ( $N_t$ )

$$A = \frac{N_i}{N_t} \times 100 .$$

Точність (P) це відсоток правильно передбачених записів від усіх передбачених записів для класу. Точність для класу j можна визначити наступним чином, використовуючи матрицю помилок (M)

$$P_j = \frac{TP_j}{TP_j + FP_j} \times 100 = \frac{M_{j,j}}{M_{j,j} + \sum_{\substack{i=1 \\ i \neq j}}^{N} M_{j,i}} \times 100 .$$

Повнота (R) це відсоток правильно передбачених записів серед усіх доступних записів для певного класу в наборі даних. Повноту для класу j можна визначити наступним чином, використовуючи матрицю плутанини (M)

$$R_j = \frac{TP_j}{TP_j + FN_j} \times 100 = \frac{M_{j,j}}{M_{j,j} + \sum_{\substack{i=1 \\ i \neq j}}^{N} M_{i,j}} \times 100 .$$

F-міра (F). Дає цілісну оцінку моделі і розраховується як середнє гармонійне значення точності та повноти у відсотках. Визначається наступним чином для класу j

$$F_j = \frac{2 \times P_j \times R_j}{P_j + R_j} \times 100$$

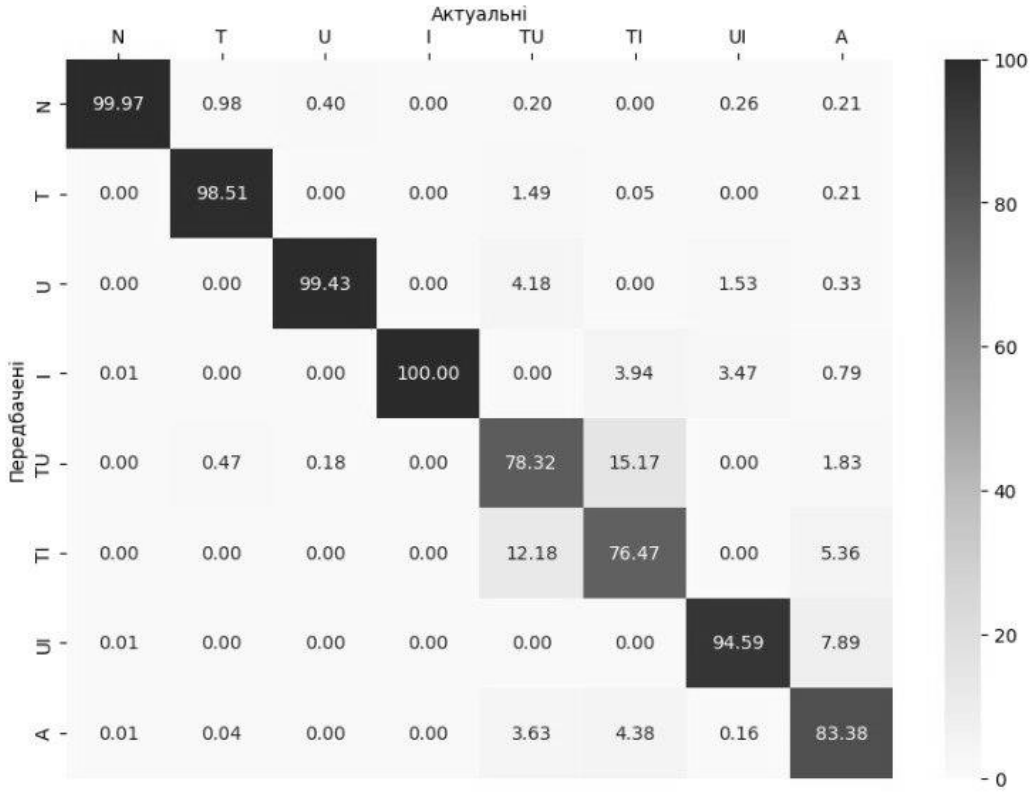


Рисунок 2.10 − Матриця помилок для 8-класової класифікації в моделі SAE

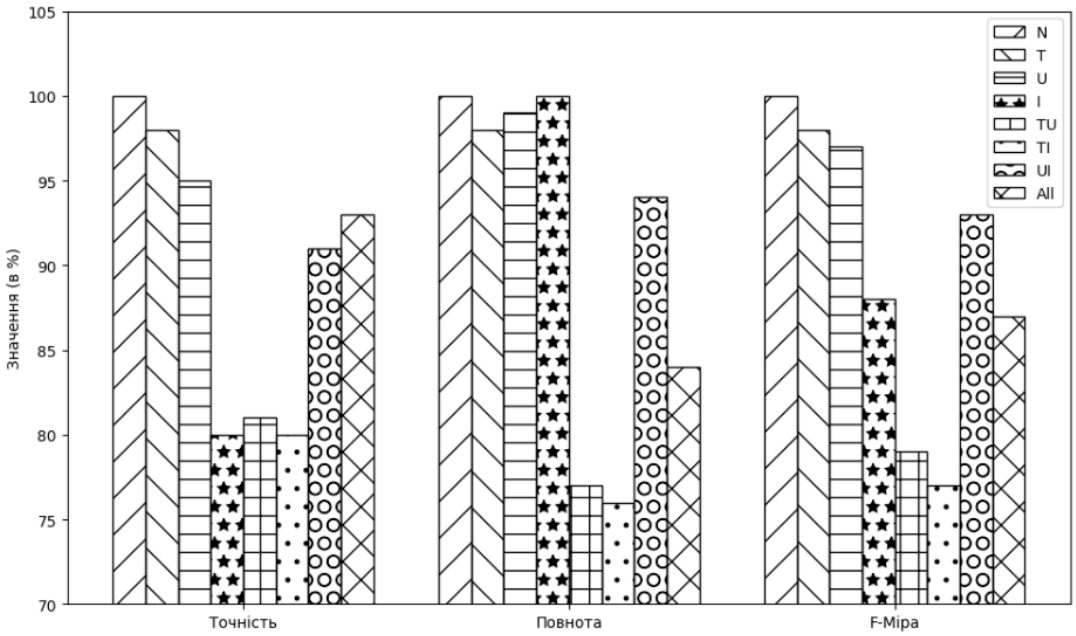


Рисунок 2.11 − Значення точності, повноти та F-міри для 8-го класу

Модель класифікації на основі SAE для модуля ТС була реалізована з використанням навчального набору даних, а оцінка була проведена з використанням тестового набору даних. Було розроблено 8-класову модель класифікації трафіку для поділу трафіку на звичайний та сім різних типів

DDoS-атак. Для порівняння з SAE була розроблена модель виявлення атак з використанням регресії softmax. Після виконання параметрів пошуку визначено, що модель SAE досягла кращої точності порівняно з регресією softmax (табл. 2.10). На рис. 2.11 показано обчислені значення точності, повноти та F-міри для моделей SAE, які були отримані з матриці помилок, показаної на рис. 2.10. Як видно з рис. 2.11, модель має значення F-міри більше 90% у випадку звичайного трафіку, TCP, UDP та UDP разом з атакуючим трафіком ICMP. Модель має відносно низькі значення F-міри для комбінованих атак TCP з ICMP та UDP через їх класифікацію до інших типів атак (рис. 2.10). Однак, їх класифікація до нормального трафіку становить менше 0,2%.

Таблиця 2.10 − Порівняння точності класифікації між моделями на основі softmax та SAE

| Метод | Точність (%) |
|---|---|
| Softmax | 95,30 |
| SAE | 96,65 |

Таблиця 2.11 − Точність і частота помилкових спрацьовувань для моделей SAE 8-го і 2-го класів

| Метод | Точність (%) | Хибно-позитивні значення (%) |
|---|---|---|
| Softmax | 95,76 | 0,6 |

Запропонована 2-класова модель класифікації, яка розглядає різні DDoS-атаки як один клас атак. Ефективність класифікації за 2-ма класами показана на рис. 2.13. Було досягнуто точності 99,8%, а також значення F-міри 99,93% і 99,66% відповідно. Ці значення були отримані з матриці помілок, показаної на рис. 2.12. Точність та частота помилкових спрацьовувань показані в табл. 2.11 для 2-класних та 8-класних моделей SAE.

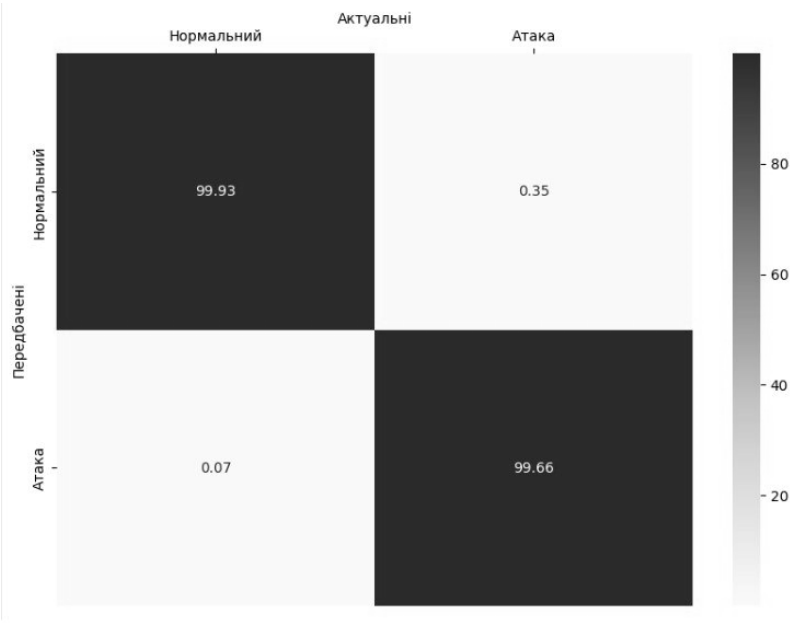


Рисунок 2.12 − Матриця помилок для 2-класної класифікації

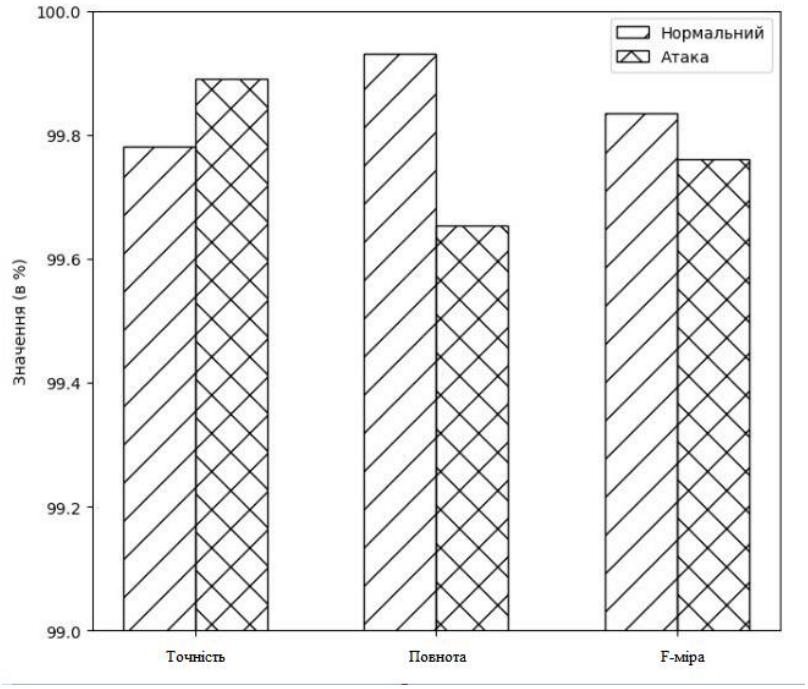


Рисунок 2.13 − Значення точності, повноти та F-міри для 2-класної класифікації

Висновки до розділу 2

Проведений аналіз демонструє важливість використання набору даних NSL-KDD для розробки та оцінки мережевих систем виявлення вторгнень. NSL-KDD є покращеною версією KDD Cup 99 і вирішує проблему

надлишкових записів, які впливають на точність класифікації. Відсутність надлишкових даних у NSL-KDD дозволяє створювати більш точні та об'єктивні моделі виявлення вторгнень. Завдяки детальному поділу даних на нормальні та аномальні трафіки (включаючи DoS, U2R, R2L та зондування), NSL-KDD є важливим інструментом для досліджень в галузі безпеки мереж. Запропоновані категорії атак і структуровані ознаки дають можливість системам глубокого навчання навчатися на повному наборі даних і ефективно виявляти нові та складні загрози. Крім того, покращене кодування та класифікація даних роблять цей набір придатним для моделювання реальних умов мережевих загроз. Особливо слід відзначити, що NSL-KDD дозволяє проводити оцінку систем виявлення вторгнень з використанням алгоритмів глибокого навчання, що підвищує їхню здатність виявляти нові типи атак і вдосконалювати захист мережевих інфраструктур. Це робить NSL-KDD одним з основних інструментів для оцінки NIDS, які використовують сучасні підходи на основі глибокого навчання.

Мережева система виявлення вторгнень, побудована на основі глибокого навчання, демонструє високу ефективність у виявленні багатовекторних DDoS-атак в середовищі SDN. Аналіз показав, що атаки можуть відбуватися як на площині управління, так і на площині даних, що підкреслює необхідність комплексного підходу до їх виявлення та запобігання.

Особливо ефективною є інтеграція NIDS на основі глибокого навчання в інфраструктуру SDN, оскільки вона забезпечує централізований контроль над мережею і дозволяє швидко і точно аналізувати великий обсяг мережевого трафіку. Використання методів глибокого навчання, таких як розріджений автокодер для навчання ознакам, підвищує точність виявлення DDoS-атак навіть у випадках, коли атаки є складними, комбінованими або спрямовані на різні протоколи (TCP, UDP, ICMP).

Проведене дослідження характеристик NIDS на основі глибокого навчання у бездротовому середовищі показало високу ефективність підходу для

виявлення DDoS-атак. Модель, заснована на SAE, показала значну перевагу в точності класифікації трафіку порівняно з традиційними методами.

Модель SAE досягла точності понад 96% для 8-класової класифікації, що перевищує показники моделі на основі softmax (95,30%). Найвищі значення F-міри (більше 90%) спостерігалися для звичайного трафіку та атак типу TCP, UDP і UDP+ICMP. Незважаючи на незначне зниження продуктивності для комбінованих атак типу TCP+ICMP і UDP, загальний рівень помилкових позитивних спрацьовувань був мінімальним.

Запропонована 2-класова модель SAE показала точність до 99,8%, що демонструє її високу надійність для класифікації DDoS-атак. Ці результати підтверджують, що використання NIDS на основі глибокого навчання в бездротових мережах є ефективним інструментом для виявлення загроз і може бути застосоване в реальних мережевих середовищах для захисту від багатовекторних атак.

Важливим результатом дослідження є те, що система NIDS дозволяє ефективно виявляти атаки на основі аналізу симетричних та асиметричних потоків, що дає змогу мінімізувати вплив хибно позитивних спрацьовувань.

# 3 ОЦІНКА ВПЛИВУ АТАК В ПРОГРАМНО-ВИЗНАЧЕНИХ МЕРЕЖАХ НА ОСНОВІ МЕРЕЖЕВОЇ СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ

Зростаюча складність кіберзагроз та постійні зміни технологій вимагають подальшого розвитку і вдосконалення інформаційної безпеки програмно-визначеної мережі. Тому актуальним є аналіз впливу атак та вдосконалення методів і засобів захисту програмно-визначеної мережі.

Розділ присвячено аналізу впливу атак на площину даних програмно-визначеної мережі.

Визначено та проведено оцінку загроз програмно-визначеної мережі, доведено, що безпека цієї мережі значною мірою залежить від захисту площини управління мережі.

Дослідження охоплює широкий спектр сценаріїв атак, ґрунтуючись на розташуванні клієнт-серверів (веб-хостів) та зловмисників. Це дозволило детально проаналізувати різні типи атак та їх потенційні наслідки.

## 3.1 Аналіз загроз в програмно-визначених мережах

Проведений аналіз в п. 1.1 показав, що програмно-визначені мережі стають все більш поширеними в сучасних IT-інфраструктурах. Це пов’язано з фундаментальною ідеєю SDN − розділення площини управління та площини даних, що є перспективною для зручного управління мережею [52].

Архітектура SDN пропонує новий підхід до побудови мережевої інфраструктури, але при цьому може мати потенційні вразливості з точки зору інформаційної безпеки. Необхідність розділення доступу до мережевих застосунків під час взаємодії з контролером, а також питання автентифікації та авторизації при роботі з застосунками на контролері, є лише кількома аспектами безпеки, які слід враховувати при проектуванні та експлуатації SDN. Контролер управління, як ключовий елемент інфраструктури SDN, є найбільш

вразливим компонентом, оскільки атака на нього може призвести до критичних наслідків для всієї мережі [53].

Важливим аспектом управління програмно-визначеними мережами є розуміння потенційних ризиків та оперативна реакція на атаки, що включає вдосконалення захисних механізмів і застосування сучасних технологій кібербезпеки.

Ідентифікація та аналіз атак у програмно-визначених мережах є актуальним і важливим напрямком у сфері кібербезпеки, що вимагає подальших досліджень для розробки нових методів захисту та забезпечення стабільності мереж [Додаток А].

Виклики безпеки в мережах, керованих SDN, є більш загрозливими порівняно з традиційними мережами. Зазвичай, у традиційній комп’ютерній мережі декілька серверів, які є частиною цієї мережі, стають об'єктами атак. На відміну від цього, якщо зловмисники скомпрометують площину управління SDN, під загрозою опиниться вся визначена мережа. Проведений аналіз робіт [54, 55, 56] дозволив виділити такі вразливості та загрози:

*1. Загрози управління SDN*

Управління SDN зосереджене на вирішенні різноманітних завдань, що стосуються ефективного та гнучкого керування мережевими ресурсами. Однією з основних переваг SDN є централізоване управління мережевими ресурсами. Замість розподіленого управління на окремих мережевих пристроях, SDN дозволяє централізовано керувати всією мережею через центральний контролер.

Скомпрометувати SDN через управління складно, оскільки необхідні автентифікація та авторизація, але якщо вони будуть скомпрометовані, вплив на мережу буде серйозним. Помилкове адміністрування мережі (наприклад, неправильно налаштований контролер) може створити ризик відключення мережі [57].

Мережеві додатки, які працюють поверх контролера, можуть походити зі сторонніх джерел. Ці програми разом із контролером успадковують привілеї

для управління мережевою поведінкою і можуть бути шкідливими або мати помилки безпеки. Додаток з помилками безпеки може бути використаний для мережевих або хост-орієнтованих атак і може призвести до розкриття інформації або довільного виконання коду з привілеями адміністратора [58]. Шкідливий SDN-додаток може виконувати різні системні команди і в найгіршому випадку може завершити роботу контролера командою виходу з системи, а також може використовувати доступні системні ресурси, такі як процесор та пам'ять, обмежуючи доступ до них іншим програмам.

Зловмисник використовуючи привілеї адміністратора може маніпулювати системними змінними та впливати на роботу мережі в цілому. Наприклад, зміна системного часу може відключити комутатори від контролера, якщо для автентифікації використовується цифровий сертифікат [59, 60]. Комутатори, що знаходяться під впливом зловмисника можуть перешкодити SDN виконувати заплановані завдання, визначені мережевою політикою.

*2. Загрози площини управління*

Площина управління включає в себе політики мережевих додатків, а також обмін трафіком між комутаторами і контролером для адміністрування визначеної мережі. Політика одного мережевого додатку може суперечити іншим. Мережа може функціонувати неочікувано через відсутність пріоритетів у політиках. Наприклад, задана дія змінює правила маршрутизації всередині таблиць потоків (використовуються для переадресації трафіку на основі певних правил, що визначаються програмним забезпеченням SDN-контролера. Ці правила можуть включати в себе MAC-адреси, IP-адреси, порти, протоколи та інші атрибути пакетів). Мережева програма може використати цю дію для модифікації заголовків пакетів, щоб обійти політики брандмауера, що застосовуються іншими програмами. У найгіршому випадку шкідлива мережева програма з високим пріоритетом може видалити правила з таблиць потоків.

Згідно зі специфікацією OpenFlow (OF), канал зв’язку між контролером і комутаторами може бути реалізований за допомогою шифрування TLS/SSL або

звичайного TCP. Аналіз показав, що багато виробників комутаторів і контролерів використовують протокол TCP, щоб уникнути складності, пов'язаної з зашифрованим каналом [61]. Це допустимо при функціонуванні в безпечній інфраструктурі. Однак, якщо трафік сигналізації здійснюється через незахищену мережу, атаки типу «людина посередині» або підслуховування можуть бути успішно реалізовані. Це може статися в програмно-визначених мобільних мережах, мережах Wi-Fi або якщо сигнальний трафік проходить через мережу, що знаходиться під керуванням зловмисника. Зловмисники можуть виявляти сигнальний трафік з каналу, щоб визначити топологію мережі, а також модифікувати його, щоб змусити мережу працювати непередбачувано. Крім того, вони можуть перевантажувати таблиці потоків, встановлюючи велику кількість правил, як тільки отримують доступ до каналу [62].

Крім того, комутатор під керуванням зловмисника, з підробленою ідентичністю справжнього комутатора може відключити останній від мережі, що в подальшому призведе до відключення всіх кінцевих хостів, пов'язаних зі справжнім комутатором, і може призвести до порушення мережевого трафіку [63]. Комутатор під керуванням зловмисника може перевантажити контролер занадто великою кількістю фальшивих запитів PACKET IN і обмежити його доступність для обробки справжніх запитів потоку. Також, комутатор і контролер обмінюються ехо-запитами та повідомленнями-відповідями, щоб перевірити наявність та справність з'єднання між ними. Комутатор під управлінням зловмисника також може використовувати ці керуючі повідомлення для перевантаження контролера.

*3. Загрози на рівні даних*

Площина даних складається з комутаторів та інших мережевих пристроїв і головним чином відповідає за обробку даних, їх пересилання, відкидання, а також збір статистики. Функціонування площини даних відбувається на основі правил потоків, що надаються контролером мережі. Якщо для пакета відбувається подія flow tablemiss, пакет пересилається на контролер для встановлення правила в комутаторі або дій, які необхідно виконати для пакета.

Тому час відгуку для першого пакета в потоці, як правило, довший, ніж час відгуку наступних пакетів того ж потоку. Ця особливість SDN допомагає зловмисникам відслідковувати SDN з площини даних [64]. Зловмисник може виявити потоки, для яких контролер не встановлює правила в таблицях потоків; замість цього він надсилає повідомлення PACKET OUT для їх обробки у відповідь на повідомлення PACKET IN, що надсилаються комутаторами, які відповідають потоку. Зловмисники можуть надсилати такі потоки, при обробці яких відбувається перевантаження контролера. Пакети в цих потоках займають буфер пам'яті комутатора до тих пір, поки не отримають відповіді від контролера, що призводить до погіршення продуктивності мережі. Кількість пакетів, які комутатор може зберігати в своєму буфері під час події PACKET IN обмежена і визначається під час з'єднання комутатора з контролером. Коли комутатор має достатньо пам'яті для буферизації пакетів, він надсилає повідомлення PACKET IN з невеликою частиною заголовка пакета разом з ідентифікатором буфера. У відповідь на повідомлення PACKET IN контролер надсилає повідомлення FLOW MOD або PACKET OUT, використовуючи ідентифікатор буфера в отриманому повідомленні PACKET IN. Як тільки комутатор отримує повідомлення від контролера, він видаляє пакет з буфера, ідентифікатор буфера якого збігається з ідентифікатором, зазначеним у повідомленні.

Таким чином, канал управління несе менший обсяг трафіку для повідомлення PACKET IN і його повідомлень-відповідей. Однак, коли комутатори вичерпують свій буфер пам'яті через велику кількість фальшивих запитів PACKET IN від зловмисників, вони починають надсилати повний пакет в повідомленні PACKET IN, а також отримувати весь пакет з повідомленням-відповіддю від контролера. Таким чином, канал перевантажується великим обсягом трафіку порівняно з попереднім випадком [65].

В дослідженні [66] проведено реалізацію та аналіз тих атак, для яких зловмисники не вимагають жодної автентифікації для доступу до системи SDN. У цьому контексті обрано атаку на площину даних (рис. 3.1).

При атаки на площину даних зловмисники надсилають фальшиві запити, щоб завантажити контролер і комутатори для їх обробки, що призводить до затримок і втрат у налаштуванні правил потоку для легального трафіку.

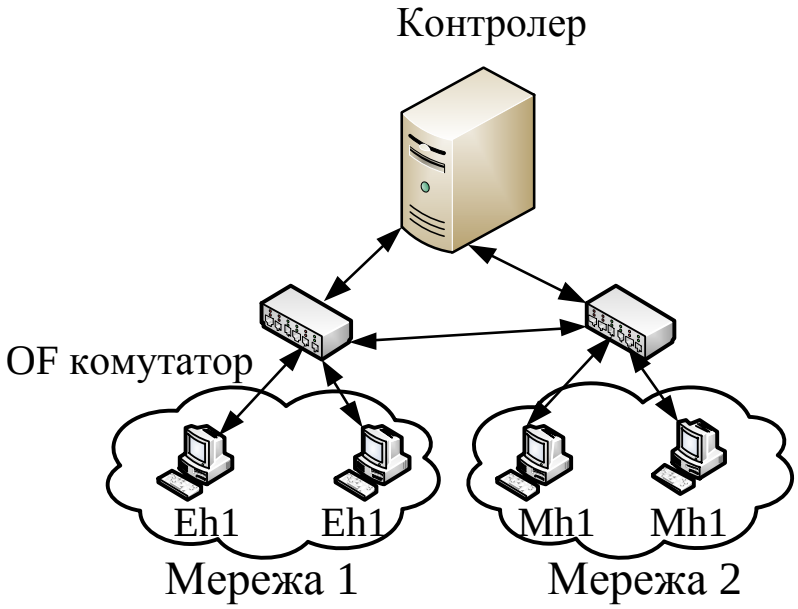


Рисунок 3.1 – варіант структури мережі для дослідження атаки на площину даних: Eh1,Eh2 – веб-хости користувачів, Mh1, Mh2 – зловмисники

Проведена оцінка впливу цієї атаки на затримку встановлення з'єднання та втрату клієнтських запитів до веб-серверів. Зі зростанням електронної комерції, мережеві оператори повинні відповідати Service Level Agreement (SLA) для своїх веб-клієнтів. Доступність веб-сервісів і час відгуку є двома важливими показниками в будь-якому SLA. Розрахунок цих показників проводився на основі вимірювання втрат клієнтських запитів і затримки встановлення з'єднання.

## 3.2 Оцінка впливу атак в програмно-визначених мережах

На основі контролера POX (контролер з відкритим вихідним кодом на основі Python, який забезпечує платформу для швидкої розробки SDN-додатків) та програмного емулятора SDN – Mininet (емулює мережеві хости, комутатори та зв'язки між ними) проведено дослідження атаки на площину даних.

Комутатори на основі OF реалізовані за допомогою Open vSwitch. Комутатори підключаються до контролера, який працює на тому ж хості або на віддаленому хості. Проведена модифікація наявного компоненту в POX для Ethernet-комутатора. Модифікований компонент встановлює правила потоку тільки для веб-трафіку, використовуючи повідомлення FLOW MOD. Для будь-якого іншого трафіку він генерує подію PACKET OUT для перенаправлення пакетів з певних портів комутаторів без встановлення правил. Тайм-аут простою для правила потоку встановлений на 15 секунд. Встановлене правило буде видалено з таблиць потоків, якщо комутатор не отримає жодного пакета, який відповідає правилу протягом цього періоду очікування. Веб-клієнт надсилає запит на веб-сервер через певний проміжок часу, що вважається легітимним трафіком. Вплив атаки оцінюються у двох випадках.

У першому випадку інтервал між кожним запитом встановлюється меншим, ніж тайм-аут встановленого правила. У другому випадку інтервал встановлюється більшим, ніж тайм-аут встановленого правила.

Для автоматизації клієнтських запитів була використана утиліта curl (утиліта з відкритим вихідним кодом командного рядка, який підтримує різні протоколи, що використовуються для передачі даних мережею). Використовуючи цю утиліту можна встановити максимальний часовий ліміт, тобто тайм-аут з'єднання, до якого він намагатиметься з'єднатися з сервером. З'єднання вважається втраченим після закінчення тайм-ауту (у дослідженні тайм-аут – 60 сек.).

Віртуальні хости були з'єднані з двома комутаторами, які в свою чергу були підключені до контролера. Швидкість і затримка з'єднання між контролером і комутатором, комутатором і комутатором, та хостом-комутатором наведено в табл.3.1.

Таблиця 3.1 – Лінійна швидкість та затримка різних видів з’єднань

| Тип з’єднання | Швидкість | Затримка |
|---|---|---|
| Контролер-комутатор | 100 Мбіт/с | 1 мс |
| Комутатор-комутатор | 100 Мбіт/с | 1 мс |
| Хост-комутатор | 100 Мбіт/с | 1 мс |

Для достовірної оцінки впливу атак з'єднано 10 веб-клієнт-серверних пар з комутаторами. Клієнти надсилають 20 запитів до відповідних серверів через фіксований інтервал часу за допомогою curl. Ping розглядався як ворожа атака. Хости зловмисника розглядалися як частина керованої мережі, які розташовані розподілено, щоб імітувати ботнети. Хост-зловмисник надсилає ping-пакети на свій одноранговий хост у мережі. Під час експериментів варіювалась частота надходження ping-пакетів і проводилось дослідження їхнього впливу на веб-сервіси. Розглянуто такі сценарії атаки залежно від місця розташування клієнт-серверів (веб-хостів) та зловмисників:

веб-хости і зловмисники знаходяться в одній мережі;

веб-хости знаходяться в одній мережі, а зловмисники – в іншій мережі;

веб-хости знаходяться в одній мережі, а зловмисники розподілені в різних мережах;

веб-хости та зловмисники розподілені в різних мережах.

Аналіз результатів експерименту показав, що у випадку знаходження веб-хостів клієнтів та зловмисників в одній мережі затримка встановлення з'єднання досить висока, коли клієнтські запити надсилаються після закінчення періоду тайм-ауту правил потоку, порівняно з тим, коли вони надсилаються до закінчення тайм-ауту (рис. 3.2).

У першому випадку затримка встановлення з'єднання зросла до 22 секунд, коли частота атак досягла 14 Kps (тисяч на секунду), тоді як у другому випадку - до 3 сек. Частка втрачених запитів зростає зі зростанням частоти атак в обох випадках, як показано на рис. 3.3. Однак, вона є вищою в першому випадку, ніж у другому, і досягає 98%. Причина високого рівня втрат, навіть коли запити генеруються до закінчення терміну дії правил потоку, пов'язана з великою кількістю пакетів ping-флуду від зловмисників, які призводять до вичерпання буфера пам'яті в комутаторі для черги вхідних пакетів для легітимного трафіку.

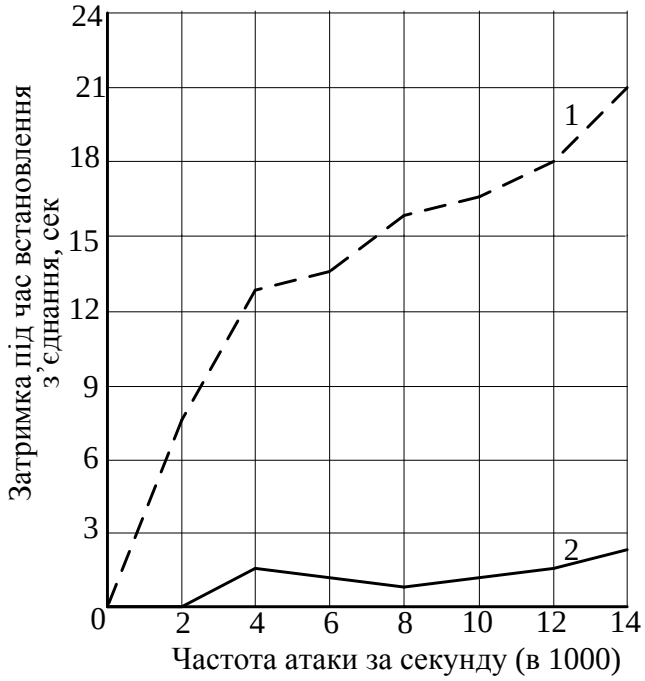


Рисунок 3.2 – Середня затримка встановлення підключення при знаходженні всіх веб-хостів та зловмисників в одній мережі: 1 − кількість надісланих запитів до закінчення дії правил потоку, 2 − кількість надісланих запитів після закінчення дії правил потоку

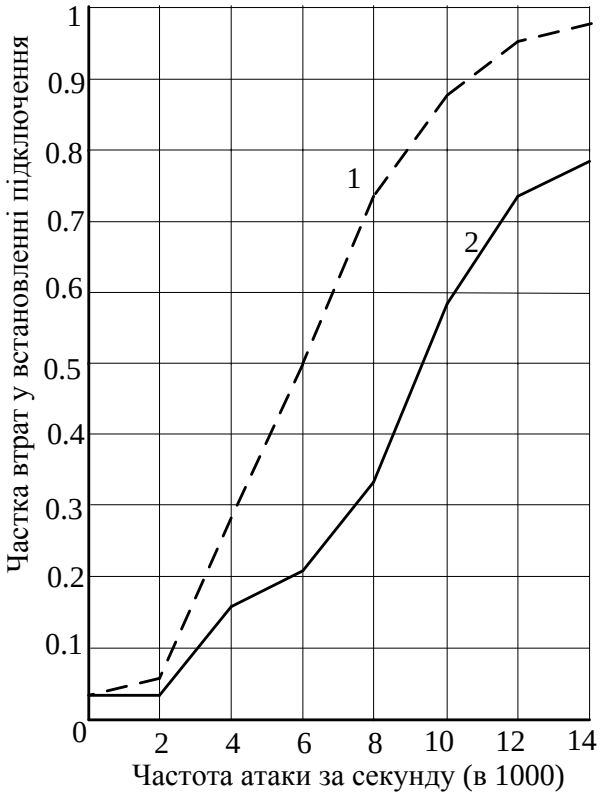


Рисунок 3.3 – Середня частка втрат при знаходженні всіх веб-хостів та зловмисників в одній мережі: 1 − кількість надісланих запитів до закінчення дії правил потоку, 2 − кількість надісланих запитів після закінчення дії правил потоку

Якщо веб-хости знаходяться в одній мережі, а зловмисники – в іншій мережі (у цій експериментальній топології веб-вузли користувачів підключено до одного комутатора, а хости зловмисника до іншого комутатора, відповідно), то вплив атаки є дуже незначним (рис. 3.4, рис. 3.5). Затримка зросла до 0,11 с і 0,06 с для запитів, що були згенеровані після та до закінчення терміну дії правил потоку, відповідно.

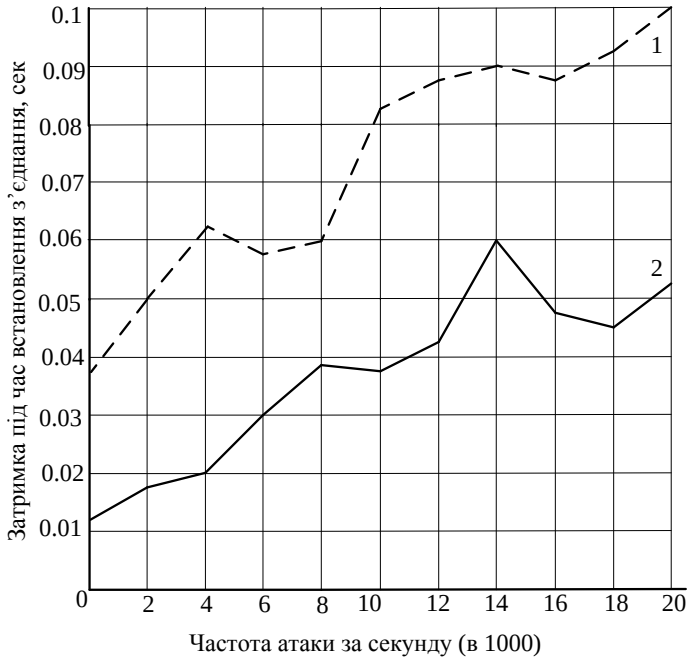


Рисунок 3.4 – Середня затримка встановлення підключення при знаходженні веб-хостів в одній мережі, а зловмисників в іншій мережі: 1 − кількість надісланих запитів до закінчення дії правил потоку, 2 − кількість надісланих запитів після закінчення дії правил потоку

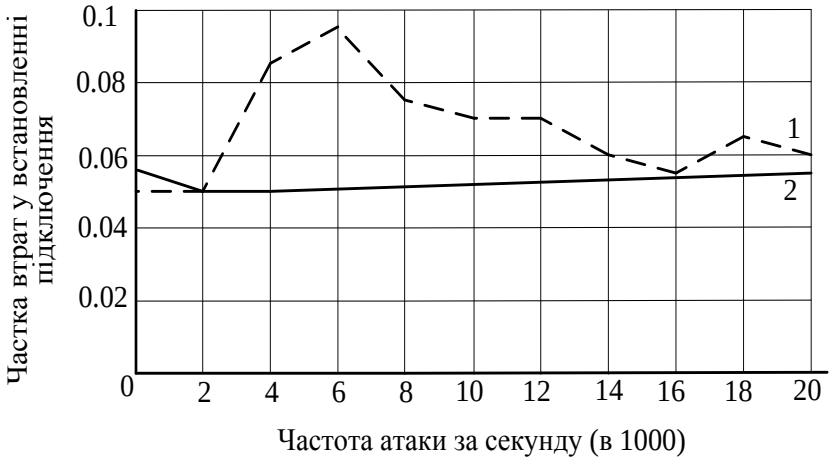


Рисунок 3.5 – Середня частка втрат при знаходженні веб-хостів в одній мережі, а зловмисників – в мержі веб-хостів та в іншій мережі: 1 − кількість надісланих запитів до закінчення дії правил потоку, 2 − кількість надісланих запитів після закінчення дії правил потоку

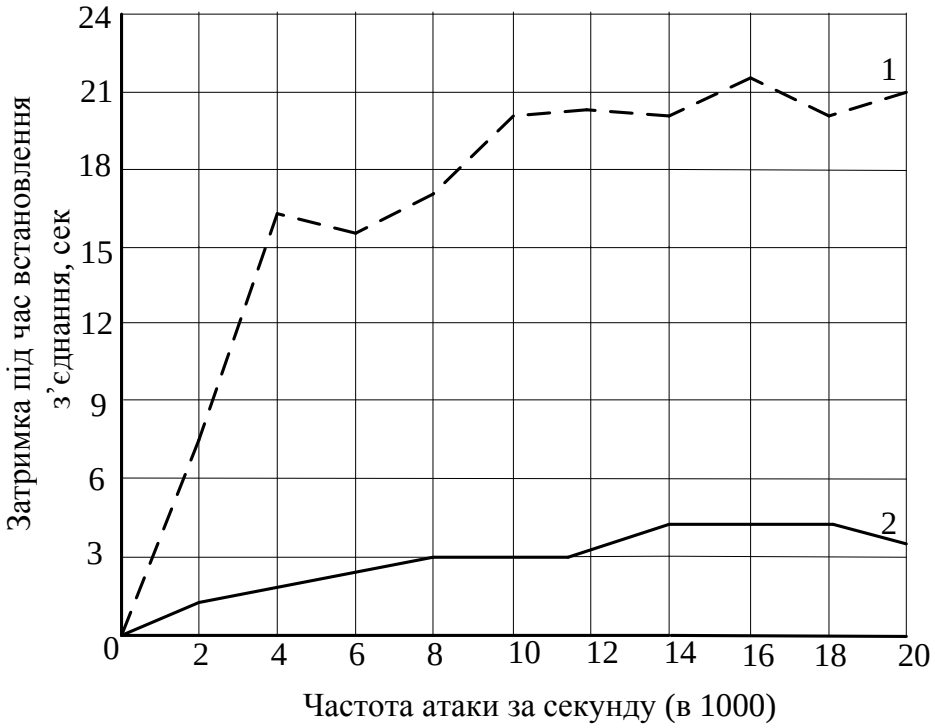


Рисунок 3.6 – Середня затримка встановлення підключення коли веб-хости знаходяться в одній мережі, а зловмисники – в мережі веб-хостів та в іншій мережі: 1 − кількість надісланих запитів до закінчення дії правил потоку, 2 − кількість надісланих запитів після закінчення дії правил потоку

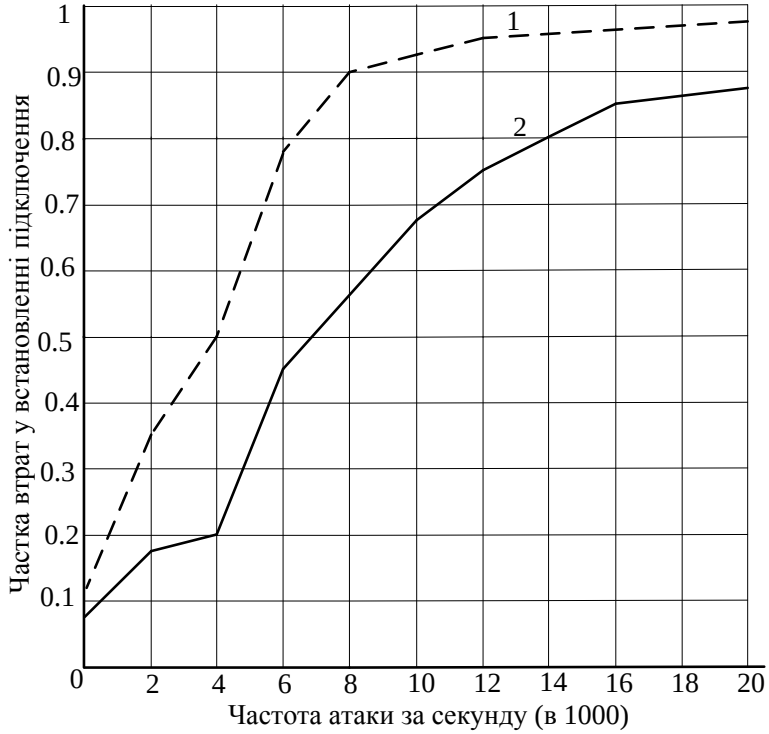


Рисунок 3.7 – Середня частка втрат коли веб-хости знаходяться в одній мережі, а зловмисники – в мержі веб-хостів та в іншій мережі: 1 − кількість надісланих запитів до закінчення дії правил потоку, 2 − кількість надісланих запитів після закінчення дії правил потоку

У випадку коли веб-хости і зловмисники розподілені в різних мережах (клієнти пов'язані з одним комутатором, а сервери були пов'язані з іншим комутатором. Зловмисники були пов'язані з обома комутаторами) наслідки атаки є найсерйознішими з усіх. Ця атака споживає ресурси контролера, переповнює буфер пам'яті та таблиці потоків комутаторів, а також переповнює всі канали зв'язку. Затримка встановлення з'єднання збільшилася до 52 секунд, навіть коли запити відправлені до закінчення терміну дії правил потоку, що дає уявлення про падіння реального трафіку на комутаторах через велику кількість пакетів зловмисника (рис. 3.8).

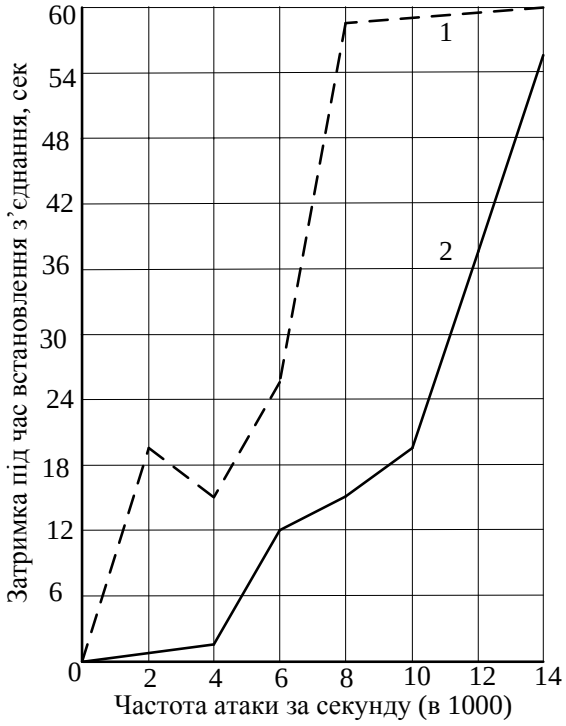


Рисунок 3.8 – Середня затримка встановлення підключення коли веб-хости та зловмисники розподілені по різним мережам: 1 − кількість надісланих запитів до закінчення дії правил потоку, 2 − кількість надісланих запитів після закінчення дії правил потоку

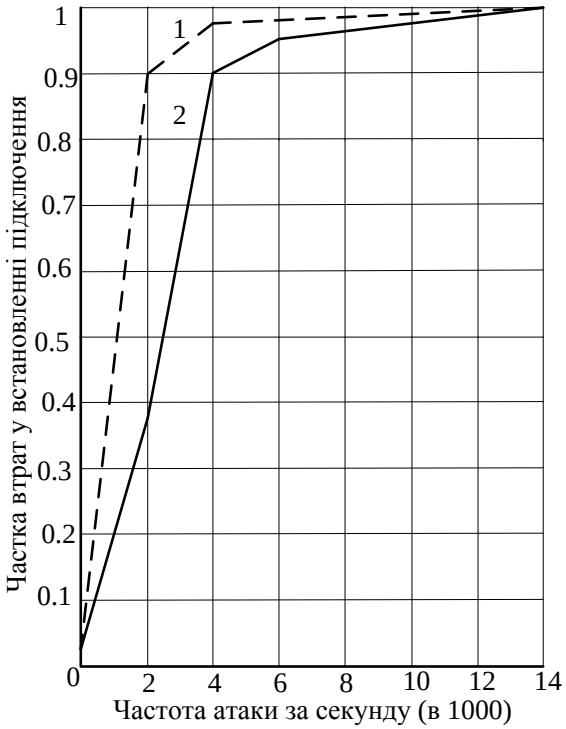


Рисунок 3.9 – Середня частка втрат коли веб-хости знаходяться в одній мережі, а зловмисники – в мержі веб-хостів та в іншій мережі: 1 − кількість надісланих запитів до закінчення дії правил потоку, 2 − кількість надісланих запитів після закінчення дії правил потоку

Частка втрат з'єднань досягала майже 100%, коли швидкість атаки досягала 6 Кбіт/с у випадку, коли запити надсилаються після закінчення терміну дії правил потоку (рис. 3.9) [66].

## Висновки до розділу 3

Архітектура SDN суттєво змінює структуру мережі. Ключовою особливістю SDN є розділення площини управління та передачі даних, що дає можливість централізовано контролювати мережеві ресурси та застосовувати мережеві політики. Однак, цей підхід має і свої недоліки, зокрема з точки зору інформаційної безпеки, оскільки контролер SDN є вразливою точкою, атака на яку може мати критичні наслідки для всієї мережі.

Аналіз показав, що до основних загроз в SDN можна віднести:

1. Загрози управління SDN. Централізоване управління мережевими ресурсами створює значні переваги для адміністрування мережі, однак потребує високого рівня захисту. Зловмисники можуть використати вразливі мережеві додатки для втручання в роботу мережі, викликавши збій у налаштуваннях контролера. Мережеві додатки зі сторонніх джерел можуть мати вразливості або шкідливі компоненти, що відкриває доступ зловмисникам до системних ресурсів, зокрема процесора та пам'яті, обмежуючи їх доступ для легальних програм.

2. Загрози площини управління. Політики мережевих додатків можуть конфліктувати, що призводить до непередбачуваних наслідків у роботі мережі. Неправильне налаштування таблиць потоків може дозволити шкідливим програмам обходити правила маршрутизації. Відсутність шифрування каналу зв'язку між контролером і комутаторами створює ризики атак типу «людина посередині» або підслуховування. Зловмисники можуть модифікувати сигнальний трафік і маніпулювати мережевою топологією.

3. Загрози площини даних. Комутатори можуть бути перевантажені фальшивими запитами, що призводить до зростання затримок і втрат у

налаштуванні правил для легального трафіку. Це створює серйозні перешкоди для обслуговування клієнтів, особливо у веб-сервісах з високими вимогами до SLA. Зловмисники можуть впливати на буферизацію пакетів у комутаторах, що призводить до перевантаження каналу зв'язку між контролером та комутаторами, збільшуючи обсяг керуючого трафіку.

Проведена оцінка впливу атак зловмисників на продуктивність мережевих сервісів, що працюють через SDN.

Оцінка впливу різних атак дає уявлення про аналіз їх ризиків і дозволяє розрахувати загальну систему оцінки вразливостей, що відповідає цим атакам. Визначено, що продуктивність веб-сервісів, враховуючи час відгуку та доступність, значно погіршується за наявності атак.

Негативний вплив атак на час відгуку та доступність створює загрози для забезпечення операторами нормованого SLA для своїх клієнтів.

# 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

В цьому розділі запропоновано реалізація стартап проекту на основі мережевої системи виявлення вторгнень на основі глибокого навчання, проведено маркетинговий аналіз та обрано основні вектори розвитку проекту, а також оцінено його маркетингову цінність.

## 4.1 Опис ідеї проекту

### 4.1.1 Основна ідея проекту

Ідея проекту полягає у створенні інноваційної системи безпеки мережі, яка використовує переваги нейронних мереж для виявлення атак і загроз, що можуть бути пропущені традиційними системами.

У табл. 4.1 зображено зміст ідеї, цільова аудиторія, переваги пропозиції та цілі на найближчий рік.

Метою стартап-проекту є побудова високопродуктивної, адаптивної та масштабованої системи кібербезпеки на основі нейроних мереж, яка забезпечить виявлення та запобігання нових і еволюціонуючих кіберзагроз у реальному часі. Система повинна перевершувати традиційні методи виявлення вторгнень за точністю, швидкістю реакції та здатністю адаптуватися до нових викликів.

Для валідації потенційних техніко-економічних переваг програмного забезпечення з популярними гравцями на ринку було проведено аналіз існуючих пропозицій програмного забезпечення та проведено їх порівняльний аналіз із запропонованою ідеєю.

Для порівняння було обрані рішення світових виробників, які є альтернативними запропонованому проекту:

Таблиця 4.1 − Опис ідеї стартап проекту

| Елемент опису | Детальний опис |
|---|---|
| Назва стартапу | CyberGuard (CG) |
| Основна ідея | Створення інноваційної платформи для автоматизації бізнес-процесів за допомогою новітніх цифрових технологій, таких як штучний інтелект, машинне навчання та аналіз даних. |
| Цільова аудиторія | Малий і середній бізнес, який прагне оптимізувати свої процеси, зменшити витрати та підвищити ефективність завдяки впровадженню інноваційних рішень. |
| Унікальна торгова пропозиція (УТП) | Просте у використанні рішення, що поєднує в собі комплексні інструменти для аналізу, автоматизації та управління бізнес-процесами, доступні навіть для компаній без глибоких технічних знань. |
| Основні продукти/послуги | 1. Платформа для автоматизації бізнес-процесів.<br>2. Інструменти для аналітики даних.<br>3. Інтеграції з популярними системами управління (ERP, CRM).<br>4. Консультаційні послуги для налаштування і впровадження. |
| Проблеми, які вирішує | Скорочення часу на рутинні процеси, зниження ризиків помилок, підвищення прозорості бізнесу, покращення прийняття рішень на основі даних. |
| Конкурентні переваги | Інтуїтивний інтерфейс, адаптованість під специфіку клієнта, доступна ціна, підтримка новітніх технологій, таких як AI та ML. |
| Етап розробки | Наразі проєкт перебуває на етапі прототипування та тестування MVP (мінімально життєздатного продукту) з першими клієнтами. |
| Стратегія виходу на ринок | Орієнтація на прямий продаж, партнерства з іншими технологічними компаніями, активний маркетинг у соціальних мережах та цифрових каналах. |
| Фінансова модель | Підписка на платформу з щомісячною оплатою, додаткові платежі за консультації та інтеграційні послуги. |
| Цілі на найближчий рік | Залучення перших 100 клієнтів, розширення функціональності платформи, вихід на міжнародний ринок. |

Cisco. Один з лідерів ринку, пропонує широкий спектр рішень для підприємств будь-якого розміру. Відрізняється високою продуктивністю та глибокою інтеграцією з іншими продуктами Cisco.

Fortinet. Комплексні рішення для безпеки мережі, включаючи NIDS. Відзначається високою ефективністю виявлення загроз і гнучкою ліцензійною політикою.

Palo Alto Networks. Спеціалізується на наступному поколінні фаєрволів, що включають в себе функції NIDS. Відрізняється високою швидкістю обробки трафіку і можливостями попередження загроз.

Snort. Відкритий вихідний код. Активна спільнота користувачів Snort забезпечує підтримку, розвиток і розширення можливостей системи.

Suricata. Відкритий код, проект, що активно розвивається. Відрізняється високою продуктивністю і широкими можливостями.

Проведемо аналіз характеристик конкурентів проекту.

В табл. 4.2 вказано сильні, слабки та нейтральні характеристики кожного із конкурентів.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

| **Постачальник** | **Сильні характеристики** | **Слабкі характеристики** | **Нейтральні характеристики** | **Поріг входу** | **Користувальницький досвід** | **Гнучкість** |
|---|---|---|---|---|---|---|
| **Cisco** | Висока продуктивність, глибока інтеграція з іншими продуктами Cisco | Висока вартість продуктів і рішень | Пропонує широкий спектр рішень для різних розмірів бізнесу | Високий | Залежить від досвіду роботи з продуктами Cisco | Гнучка, але більше орієнтована на продукти Cisco |
| **Fortinet** | Висока ефективність виявлення загроз, гнучка ліцензійна політика | Складність налаштування для новачків | Комплексні рішення для мережевої безпеки | Середній | Інтуїтивний інтерфейс після налаштування | Гнучка, але може вимагати додаткових ресурсів для налаштування |
| **Palo Alto Networks** | Висока швидкість обробки трафіку, можливості попередження загроз | Висока вартість впровадження | Спеціалізується на фаєрволах наступного покоління | Високий | Дуже позитивний, але вимагає навичок | Висока гнучкість, можливості розширення функцій |
| **Snort** | Відкритий вихідний код, активна підтримка спільноти, постійний розвиток | Не завжди забезпечує високу продуктивність на великих навантаженнях | Легко інтегрується з іншими рішеннями | Низький | Добрий досвід для технічно підкованих користувачів | Гнучка завдяки відкритому коду |
| **Suricata** | Висока продуктивність, широкі можливості налаштування | Вимагає значних знань для ефективного використання | Проект з відкритим кодом, активно розвивається | Середній | Технічно орієнтований досвід | Дуже гнучка через можливості конфігурації |

Аналіз конкурентів та вивчення їх позицій на ринку показав, що кожен має свою аудиторію, але повністю не відповідає всім потребам користувачів.

### 4.1.2 Технологічний аудит ідеї проекту

Представлена технологічна здійсненність ідеї стартап проекту (табл. 4.3).

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

| **Фактор** | **Опис** | **Вплив на здійсненність** |
|---|---|---|
| Технологічні ресурси | Наявність сучасних технологій для розробки NIDS, таких як алгоритми машинного навчання і аналітики. | Високий |
| Досвід команди | Команда розробників має досвід у створенні систем безпеки та програмного забезпечення. | Високий |
| Доступ до ринку | Можливість доступу до потенційних клієнтів і партнерів через стратегічні альянси. | Середній |
| Фінансування | Наявність інвестицій для підтримки розвитку проекту і виходу на ринок. | Високий |
| Конкуренція | Аналіз конкурентного середовища та здатність пропонувати унікальні рішення. | Середній |
| Відповідність стандартам | Відповідність галузевим стандартам та нормативам безпеки. | Високий |

Фактори оцінюють ключові аспекти, які впливають на успішність реалізації проекту, такі як технологічні ресурси, досвід команди, доступ до ринку, фінансування, конкуренція та відповідність стандартам. Кожен фактор оцінюється за його впливом на реалізацію ідеї, що допомагає виявити сильні та слабкі сторони проекту [67].

## 4.2 Аналіз ринкових можливостей запуску проекту CyberGuard

Для визначення актуальності та доцільності розробки проекту потрібно провести аналіз ринкових можливостей його запуску, тобто прооаналізувати переваги та загрози, які можуть вплинути на маркетингову стратегію розвитку проекту.

У табл. 4.4 наведено комплексний аналіз потенційного ринку для стартапу, що спеціалізується на розробці систем виявлення вторгнень (NIDS). Вона охоплює ключові аспекти ринку, включаючи цільові сегменти, обсяги, тенденції та конкурентне середовище.

Цільовий ринок складається з малих та середніх підприємств (SMB), великих корпорацій, комунікаційних компаній, фінансових установ та державного сектору.

Таблиця 4.4 – Загальна характеристика ринку стартап-проекту

| Характеристика | Опис |
|---|---|
| **Цільовий ринок** | Малі та середні підприємства (SMB), великі підприємства, фінансові установи, телекомунікаційні компанії, державний сектор. |
| **Обсяг ринку** | Глобальний ринок рішень з кібербезпеки зростає щорічно. Ринок NIDS оцінюється в мільярди оцінюється в мільярди доларів із стабільним ростом на 12-15% щорічно. |
| **Тенденції ринку** | Зростання кількості кіберзагроз і атак, перехід на гібридні та хмарні інфраструктури, зростання попиту на автоматизацію безпеки. |
| **Основні конкуренти** | Cisco, Fortinet, Palo Alto Networks, Snort, Suricata, а також інші постачальники рішень NIDS та кібербезпеки. |
| **Бар'єри входу** | Складнощі із забезпеченням високого рівня продуктивності, високі стандарти кібербезпеки, необхідність сертифікації, патенти та авторські права. |
| **Попит на NIDS** | Високий попит на рішення з кібербезпеки серед підприємств, які впроваджують хмарні та гібридні інфраструктури, ринки, що швидко розвиваються. |
| **Регуляторні вимоги** | Дотримання стандартів кібербезпеки (GDPR, ISO/IEC 27001, NIST), необхідність відповідності регуляторним нормам залежно від країни. |
| **Сегментація клієнтів** | 1. Великі підприємства (складні мережі, високі вимоги до безпеки).<br>2. Малі та середні бізнеси (обмежений бюджет, але високий попит на кібербезпеку).<br>3. Державні установи (критична інфраструктура, конфіденційність). |
| **Цінова політика** | Гнучка цінова політика на основі кількості користувачів, рівня функціональності, ліцензування та обслуговування. |
| **Потреби клієнтів** | Ефективний захист від атак, легке інтегрування у наявну інфраструктуру, доступність підтримки та оновлень, масштабованість. |
| **Конкурентні переваги стартапу** | Інноваційні технології, нові методи виявлення загроз, гнучкість у налаштуванні, нижча вартість порівняно з великими гравцями, можливість швидкого впровадження. |
| **Можливості росту** | Інтеграція з хмарними сервісами та платформами, розробка спеціальних рішень для різних галузей (фінансовий сектор, державні структури, медицина). |
| **Ризики** | Швидкі технологічні зміни, загроза кібератак для самої системи, сильна конкуренція з боку гігантів ринку. |
| **Потреба у фінансуванні** | Необхідні інвестиції для розробки продукту, маркетингу та просування, забезпечення технічної підтримки клієнтів. |
| **Партнерства** | Можливість інтеграції з постачальниками хмарних послуг (AWS, Azure), телекомунікаційними провайдерами, консалтинговими фірмами з кібербезпеки. |

Різні сегменти клієнтів мають специфічні потреби, що включають захист критичної інфраструктури, конфіденційність даних та бюджетні обмеження.

Обсяг ринку демонструє, що глобальний ринок кібербезпеки продовжує стабільно зростати, з особливо високим попитом на рішення з безпеки мереж, включаючи NIDS. Попит зростає через збільшення кількості кібератак і активне впровадження хмарних та гібридних інфраструктур.

Основні конкуренти, такі як Cisco, Fortinet, Palo Alto Networks, а також відкриті рішення на кшталт Snort і Suricata, представляють важливі виклики для входження на ринок, оскільки ці компанії вже мають сильні позиції та розвинені продукти. Проте стартап може виділитися завдяки інноваційним технологіям та гнучким підходам до налаштування.

Бар'єри входу включають високі вимоги до кібербезпеки, необхідність сертифікації продуктів та наявність патентів у конкурентів. Успішний вихід на ринок вимагає дотримання міжнародних стандартів та регуляторних норм (наприклад, GDPR, ISO/IEC 27001), що додає складності в процес розробки.

Цінова політика має бути гнучкою, залежно від масштабу та потреб клієнтів, що дозволить конкурувати з великими гравцями. Важливою перевагою для стартапу є можливість запропонувати конкурентоспроможні ціни при збереженні високого рівня функціональності.

Можливості для росту полягають у здатності інтегрувати рішення NIDS з хмарними сервісами та платформами, а також у можливості розробляти спеціалізовані продукти для різних галузей, таких як фінансовий сектор, державні структури чи медицина.

Ризики для стартапу полягають у швидких змінах технологій та загрозах з боку кібератак. Також сильна конкуренція з боку великих гравців може стати викликом.

Загалом, можна зробити висновок про сприятливі умови для стартапу в сфері NIDS, водночас підкреслюючи важливість інновацій, гнучкості та правильного позиціонування на ринку для успішної конкуренції.

В табл. 4.5 наведено характеристику потенційних клієнтів стартап-проекту за такими показниками:

1. Сегмент клієнтів. Окреслено п'ять основних груп потенційних клієнтів NIDS стартапу, які відрізняються за масштабом, специфікою діяльності та рівнем потреб.

2. Основні потреби. Відображає ключові запити кожного сегмента. Малим підприємствам важлива простота та доступність, великим корпораціям – надійність і безпека. Фінансові установи зосереджені на захисті даних, а державні органи – на стійкості до загроз національного рівня.

3. Вимоги до функціональності. Кожен сегмент має свої технічні запити. Наприклад, телекомунікаційні компанії потребують високої продуктивності та швидкого реагування, медичні заклади – конфіденційності даних.

4. Бюджет. Сегменти різняться за бюджетом, від середнього у медичних установах до великого в корпораціях та державному секторі, де пріоритетом є максимальна безпека.

5. Рівень технічної підготовки. Оцінюється, наскільки клієнти мають підготовлені технічні команди для налаштування та управління NIDS.

6. Рівень безпеки. Описує загрози для кожного сегмента, від мінімальних у малих підприємствах до дуже високих у фінансових і державних структурах.

7. Потенціал для росту. Визначає можливості для розширення ринку в кожному сегменті.

Для актуальної оцінки можливостей виходу на ринок необхідно провести аналіз загроз, які можуть вплинути на успіх стартап проекту (табл. 4.6).

До фінансових загроз можна віднести недофінансування або неправильний розподіл бюджету, що може зупинити розвиток стартапу.

До технічної загрози можно віднести те, що при збільшенні обсягів трафіку або кількості клієнтів система може не витримати навантаження.

До ринкової загрози можна віднести те, що великі конкуренти мають значні переваги, що може ускладнити вихід на ринок.

До правових загроз можна віднести те, що важливо дотримуватися норм законодавства, зокрема, щодо захисту персональних даних.

До операційної загрози можна віднести те, що затримки у розробці можуть знизити конкурентоспроможність продукту.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

| **Сегмент клієнтів** | **Основні потреби** | **Вимоги до функціональності** | **Бюджет** | **Рівень технічної підготовки** | **Рівень безпеки** | **Потенціал для росту** |
|---|---|---|---|---|---|---|
| Малі та середні підприємства (SMB) | Захист від вторгнень, простота інтеграції, бюджетні рішення | Легкість налаштування, базові функції захисту | Невеликий | Середній | Низький – мінімальні загрози | Високий |
| Великі корпорації | Максимальна безпека, безперервність роботи, гнучкість | Висока продуктивність, глибока інтеграція | Великий | Високий | Високий – загрози на рівні корпоративних мереж | Середній |
| Телекомунікаційні компанії | Захист критичної інфраструктури, забезпечення надійності мережі | Висока продуктивність, швидке виявлення загроз, стабільність | Великий | Високий | Високий – загрози на рівні масштабних мереж | Високий |
| Фінансові установи | Захист конфіденційної інформації, відповідність регуляторним нормам | Максимальна безпека, відповідність стандартам | Дуже великий | Високий | Дуже високий – захист критичних даних | Середній |
| Державний сектор | Високий рівень безпеки, захист від національних і міжнародних загроз | Висока надійність, захищеність, відповідність державним нормам | Великий | Високий | Дуже високий – захист від кібератак на державні структури | Середній |
| Медичні установи | Захист конфіденційних медичних даних, відповідність медичним стандартам | Безпека, відповідність регуляціям (наприклад, HIPAA) | Середній | Низький – середній | Середній – конфіденційність та цілісність даних | Високий |

До кіберзагрози можна віднести те, що уразливості в самому продукті можуть призвести до кібератак на клієнтів.

Таблиця 4.6 – Фактори загроз

| **Категорія загроз** | **Опис загрози** | **Ймовірність виникнення** | **Можливі наслідки** | **Заходи для мінімізації** |
|---|---|---|---|---|
| Фінансові загрози | Недостатнє фінансування на розробку продукту | Висока | Зупинка проєкту через брак ресурсів | Пошук інвесторів, залучення грантів |
| Технічні загрози | Низька продуктивність системи при великому обсязі трафіку | Середня | Зниження якості роботи, втрата клієнтів | Оптимізація коду, залучення фахівців з масштабування |
| Ринкові загрози | Конкуренція з великими гравцями (Cisco, Fortinet, Palo Alto) | Висока | Втрата ринкової частки | Розробка унікальних фіч, вихід на нові ринки |
| Правові загрози | Порушення законодавства про захист даних | Низька | Штрафи, зупинка діяльності | Юридичний аудит, впровадження політик відповідності |
| Операційні загрози | Затримки в розробці продукту | Середня | Втрата клієнтів, затримка виходу на ринок | Впровадження методологій Agile, чіткий графік розробки |
| Загрози кібербезпеки | Уразливості в коді NIDS, що призведуть до кібератак | Середня | Втрата довіри клієнтів, витік даних | Регулярні перевірки безпеки, тестування на проникнення |

Проведемо аналіз умов конкуренції в галузі NIDS (табл. 4.7). В якості інструменту аналізу конкуренції в галузі NIDS будемо використовувати модель Портера [68].

Таблиця 4.7 – Аналіз конкуренції в галузі NIDS за моделлю Портера

| **Фактор** | **Опис** | **Вплив на конкурентоспроможність** |
|---|---|---|
| Сила конкурентів у галузі | Наявність багатьох учасників ринку, таких як Cisco, Fortinet, Palo Alto Networks, Snort, що пропонують схожі продукти. | Висока конкуренція призводить до необхідності інновацій і підвищення якості обслуговування. |
| Загроза нових учасників | Доступність технологій та зниження бар'єрів входу сприяють появі нових стартапів. | Висока загроза, що вимагає від існуючих компаній інвестувати в унікальні пропозиції та диференціацію. |
| Загроза замінників | Альтернативні рішення, такі як IDS/IPS системи, можуть задовольнити потреби споживачів. | Середня загроза, оскільки споживачі можуть перейти на інші рішення, якщо вони пропонують кращу цінність. |
| Сила постачальників | Наявність постачальників, які пропонують компоненти для NIDS, таких як програмне забезпечення та апаратура. | Низька до середньої сила постачальників, оскільки ринок надає безліч постачальників і варіантів. |
| Сила покупців | Клієнти мають великий вибір постачальників та можуть легко змінити постачальника. | Сильна сила покупців, яка змушує компанії пропонувати конкурентні ціни та високу якість обслуговування. |

Перевагами цієї моделлі є:

- комплексний підхід. Модель Портера дозволяє аналізувати галузь з п’яти різних перспектив — це конкуренція серед існуючих компаній, загроза нових учасників, вплив замінників, вплив постачальників та вплив клієнтів. Такий комплексний підхід дає можливість глибше розуміти силу конкурентів і перспективи ринку.
- оцінка бар’єрів для входу. Галузь NIDS має високі бар’єри для входу, зокрема через складність технології, потребу в спеціалізованих знаннях та ресурсах. Аналіз за Портером допомагає виявити ці бар’єри і визначити, наскільки складно буде новим гравцям увійти на ринок.
- загроза замінників. NIDS можна замінити іншими технологіями безпеки, такими як мережеві фаєрволи або рішення на базі штучного інтелекту.

Модель Портера допомагає оцінити загрозу від замінників і зрозуміти, як компанія може захистити свою частку ринку.

- вплив постачальників і клієнтів. В галузі NIDS часто використовуються апаратні або програмні компоненти від сторонніх постачальників, які мають великий вплив на конкурентоспроможність. Клієнти, з іншого боку, можуть вимагати нових функцій або нижчих цін. Портерова модель допомагає аналізувати вплив цих груп на бізнес і визначати стратегії для управління ними.
- конкуренція між наявними гравцями. Галузь NIDS включає сильних гравців, таких як Cisco, Fortinet, Palo Alto Networks та інші.

Аналіз за Портером дозволяє оцінити конкурентну динаміку між ними, зрозуміти їхні сильні та слабкі сторони і побачити можливі ніші для інноваційного стартапу.

Загалом, використання моделі Портера дозволяє краще зрозуміти ринкову ситуацію, визначити конкурентні переваги та загрози, а також розробити стратегію для успішного виходу на ринок.

Після проведення аналізу конкуренції в галузі NIDS доцільно провести обґрунтування факторів конкурентоспроможності проекту [69].

Обґрунтування факторів конкурентоспроможності це важливий аспект стратегічного аналізу та планування, який допомагає оцінити сильні та слабкі сторони стартап проекту, зрозуміти конкурентне середовище та визначити шляхи для зміцнення своїх позицій на ринку.

У контексті управління конкурентоспроможністю важливо враховувати різні чинники, які впливають на здатність запропонованої системи CyberGuard конкурувати, такі як якість продукції, ціна, інновації, ефективність маркетингу, рівень технологічного розвитку, управління ресурсами та інше.

На основі проведеного аналізу можна сформувати перелік основних факторів, які свідчать про високу конкурентоспроможність проекту (табл. 4.8).

Таблиця 4.8 – Обґрунтування факторів конкурентоспроможності

| **Фактор конкурентоспроможності** | **Опис фактору** | **Обґрунтування впливу на конкурентоспроможність** |
|---|---|---|
| Інноваційність продукту | Впровадження нових технологій та рішень, які відрізняють стартап від конкурентів. | Інноваційні продукти дозволяють стартапу виділитися на ринку, залучати нових клієнтів і створювати бар'єри для конкурентів. |
| Якість продукту | Високий рівень функціональності, зручності використання та надійності продукту. | Забезпечення високої якості покращує користувацький досвід, збільшує задоволеність клієнтів і сприяє довгостроковим відносинам з ними. |
| Цінова політика | Конкурентні ціни або надання додаткової цінності за ті ж гроші. | Конкурентна ціна дозволяє привернути більше клієнтів, особливо на етапі виходу на ринок, що сприяє швидкому зростанню частки ринку. |
| Технічна підтримка та обслуговування | Надання якісної та швидкої технічної підтримки, навчання користувачів, оновлення продукту. | Якісна підтримка сприяє довірі до продукту, знижує рівень відмов клієнтів та покращує репутацію компанії на ринку. |
| Команда та експертиза | Досвідчена команда з експертними знаннями у відповідній галузі. | Сильна команда забезпечує успішне виконання проєкту, швидке реагування на зміни ринку та впровадження нових функцій. |
| Стратегічні партнерства | Співпраця з іншими компаніями, що доповнюють або підсилюють продукт стартапу. | Партнерства можуть надавати доступ до нових ринків, технологій або ресурсів, що підвищує загальну конкурентоспроможність стартапу. |
| Маркетинг та брендінг | Створення впізнаваного бренду, ефективна маркетингова стратегія, позиціювання на ринку. | Сильний бренд і чітке позиціювання допомагають стартапу захопити увагу цільової аудиторії і сприяють розвитку лояльності клієнтів. |
| Гнучкість та адаптивність | Здатність швидко адаптуватися до змін на ринку, оновлювати продукт відповідно до нових потреб клієнтів. | Гнучкість дозволяє оперативно реагувати на нові виклики і можливості, що допомагає стартапу залишатися конкурентоспроможним у динамічних умовах. |
| Інвестиційна привабливість | Можливість залучення інвестицій завдяки сильному бізнес-плану, успішному трек-рекорду та великому потенціалу зростання. | Фінансова підтримка дозволяє масштабувати діяльність, впроваджувати нові функції і збільшувати долю ринку. |
| Інноваційні канали збуту та продажів | Використання нових каналів продажів, таких як онлайн-платформи, прямий продаж через соціальні мережі, тощо. | Нові канали збуту дозволяють швидше і дешевше досягати цільової аудиторії, збільшуючи обсяги продажів та покращуючи загальну ефективність бізнесу. |

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) і слабких (Weak) сторін, загроз (Threats) і можливостей (Opportunities) на основі визначених ринкових загроз і можливостей, а також сильних і слабких сторін (табл. 4.9).

Перелік ринкових загроз і можливостей складається на основі аналізу факторів загроз і факторів можливостей маркетингового середовища. Ринкові загрози та можливості є наслідками (прогнозованими результатами) впливу факторів і, на відміну від них, ще не реалізовані на ринку, але мають певну ймовірність здійснення [70].

Таблиця 4.9 – SWOT-аналіз стартап-проекту

| | |
|---|---|
| Сильні сторони:<br>- Інноваційний продукт з унікальними функціями<br>- Висока кваліфікація команди<br>- Гнучкість у прийнятті рішень та швидка адаптація до ринкових змін<br>- Ефективна стратегія брендингу та маркетингу | Слабкі сторони:<br>- Відсутність довгострокового досвіду на ринку<br>- Обмежені фінансові ресурси на початкових етапах розвитку<br>- Залежність від технологічних партнерів |
| Можливості:<br>- Зростання попиту на цифрові рішення<br>- Можливість виходу на міжнародні ринки<br>- Розширення лінійки продуктів або послуг<br>- Стратегічні альянси та партнерства | Загрози:<br>- Висока конкуренція на ринку<br>- Швидка зміна технологічних трендів<br>- Регуляторні бар’єри та зміни законодавства<br>- Можливі фінансові кризи та нестабільності |

Це знову підтверджує, що навіть незважаючи на свою специфіку, наш проект потребує значних зусиль для того, щоб увійти у ринок, зафіксуватися та пропонувати свої можливості своїм клієнтам (табл. 4.10).

На основі SWOT-аналізу розробляємо альтернативи ринкового впровадження стартап-проекту.

Таблиця 4.10 – Альтернативи ринкового впровадження стартап-проекту

| Альтернатива | Опис | Переваги |
|---|---|---|
| Вихід на локальний ринок | Запуск NIDS на ринку країни з метою тестування продукту. | Можливість отримати зворотний зв'язок і вдосконалити продукт без великих ризиків. |
| Міжнародна експансія | Запуск продукту на міжнародних ринках, зокрема в Європі та Північній Америці. | Доступ до ширшої аудиторії та можливість значного зростання. |
| Партнерство з великими компаніями | Співпраця з великими IT-компаніями для просування продукту. | Отримання доступу до ресурсів та технологій партнерів. |
| Фреймворк SaaS | Запропонувати продукт як сервіс з підпискою. | Стійкий дохід від підписок та можливість гнучкого масштабу. |
| Модель freemium | Запропонувати базову версію безкоштовно з платними функціями. | Привернення уваги до продукту та можливість монетизації через додаткові функції. |
| Розробка мобільних додатків | Створення мобільних версій NIDS для розширення доступності. | Забезпечення зручності використання та залучення нових користувачів. |
| Проведення вебінарів та тренінгів | Освітні програми для потенційних клієнтів щодо безпеки мереж. | Зміцнення репутації та довіри до продукту. |

Стратегія компенсації слабких сторін стартапу за рахунок використання наявних ринкових можливостей є одним з найефективніших підходів для забезпечення стабільного зростання та розвитку бізнесу. Цей підхід передбачає максимальне використання сильних сторін стартапу та ринкових умов для нівелювання існуючих недоліків.

## 4.3 Розробка ринкової стратегії проекту

Для роботи з вибраним цільовими групами користувачів ринку (табл. 4.5) необхідно сформувати базову стратегію розвитку (табл. 4.11).

Таблиця 4.11 – Визначення базової стратегії розвитку

| **Елемент стратегії** | **Опис** | **Обгрунтування** |
|---|---|---|
| Продуктовий розвиток | Розширення функціоналу платформи та додавання нових послуг. | Дозволяє задовольнити зростаючі потреби клієнтів і утримувати конкурентні позиції на ринку. |
| Маркетингова експансія | Активне просування на нові ринки через онлайн-канали та партнерства. | Збільшення впізнаваності бренду та залучення нових клієнтів. |
| Залучення інвестицій | Залучення додаткових інвестицій для масштабування та вдосконалення продукту. | Фінансове забезпечення дозволить реалізувати амбітні плани розвитку. |
| Покращення клієнтського досвіду | Запровадження сервісів підтримки, навчання та зворотного зв'язку. | Покращує взаємодію з клієнтами та підвищує їх лояльність до бренду. |
| Технологічні інновації | Інтеграція новітніх технологій, таких як AI та машинне навчання. | Сприяє підвищенню ефективності та забезпечує перевагу над конкурентами. |
| Розширення партнерської мережі | Налагодження співпраці з іншими компаніями для спільного розвитку. | Партнерства відкривають доступ до нових ринків і технологій. |
| Оптимізація операційних процесів | Автоматизація внутрішніх процесів для зниження витрат. | Забезпечує підвищення ефективності та зменшення витрат. |
| Масштабування на міжнародний ринок | Пошук нових можливостей для виходу на іноземні ринки. | Розширює можливості зростання та збільшує прибутки компанії. |

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.12).

Таблиця 4.12 – Визначення базової стратегії конкурентної поведінки

| **Елемент стратегії** | **Опис** |
|---|---|
| Цільовий ринок | Визначення конкретних сегментів ринку, на які буде спрямовано пропозицію NIDS. |
| Конкурентні переваги | Висока ефективність виявлення загроз, інтуїтивно зрозумілий інтерфейс, гнучка налаштування. |
| Основні конкуренти | Основні гравці ринку, такі як Cisco, Fortinet, Palo Alto Networks. |
| Цінова стратегія | Конкурентоспроможні ціни з можливістю гнучкого налаштування ліцензій. |
| Маркетингова стратегія | Використання цифрового маркетингу, соціальних мереж та цільових кампаній. |
| Стратегія продукту | Розробка продукту з фокусом на потреби користувачів і інноваційні технології. |
| Лідерство у витратах | Зосередження на зниженні витрат для досягнення конкурентної переваги. |
| Диференціація | Пропозиція унікальних продуктів або послуг, які відрізняються від конкурентів. |
| Фокусування | Сфокусування на вузькому сегменті ринку. |

На основі вимог споживачів обраного сегменту до постачальника і продукту, а також залежно від стратегії розвитку та стратегії конкурентної поведінки, розробляється стратегія позиціонування, яка визначається у формуванні ринкової позиції, за якою споживачі мають ідентифікувати проект (табл. 4.13).

Таблиця 4.13 – Визначення стратегії позиціонування

| **Елемент стратегії** | **Опис** | **Обґрунтування** |
|---|---|---|
| Цільовий ринок | Малий та середній бізнес, що потребує цифрових рішень для автоматизації процесів. | Цей сегмент часто має обмежені ресурси на великі рішення, тому потребує доступних і ефективних інструментів. |
| Цінова політика | Доступна підписка з варіантами різних пакетів послуг. | Цінова стратегія дозволяє клієнтам починати з базового рівня і поступово розширювати функціональність за потреби. |
| Унікальна пропозиція | Інноваційна платформа з простим інтерфейсом для автоматизації процесів. | Пропонує легкість у використанні та високу адаптивність до специфічних потреб бізнесу. |
| Позиціонування бренду | Технологічний партнер для зростання і розвитку бізнесу. | Надає образ надійного і сучасного рішення, яке сприяє розвитку компаній-клієнтів. |
| Канали розповсюдження | Онлайн-продажі, партнерські програми, виставки та галузеві заходи. | Ці канали дозволяють швидко охоплювати цільову аудиторію та розвивати мережу партнерів. |
| Стратегія просування | Цифровий маркетинг, контент-маркетинг, SEO та SMM. | Ефективне використання онлайн-каналів дозволяє досягати широкої аудиторії та будувати довіру до бренду. |
| Конкурентні переваги | Гнучкість, простота використання, адаптивність до потреб клієнтів. | Переваги, які важливі для цільової аудиторії, що хоче швидких і зручних рішень. |
| Довгострокові цілі | Розширення на міжнародні ринки,.вдоск. платформи. | Спрямовані на масштабування бізнесу та закріплення позицій на глобальному рівні. |

Проведемо розробку маркетингової програми стартап-проекту CyberGuard.

Під час розроблення маркетингової програми доцільно здійснити розробку маркетингової концепції CyberGuard, яку отримає споживач. У табл. 4.14 підсумовуються результати аналізу конкурентоспроможності CyberGuard.

Таблиця 4.14 – Визначення ключових переваг концепції товару (системи)

| Ключові переваги | Опис | Вплив на стартап |
|---|---|---|
| Висока продуктивність | Забезпечення швидкої обробки трафіку для своєчасного виявлення загроз. | Покращення реакції на загрози та зменшення потенційних збитків. |
| Гнучкість | Можливість налаштування системи під конкретні вимоги клієнтів. | Збільшення залученості клієнтів та підвищення їхньої задоволеності. |
| Інтеграція з існуючими рішеннями | Легкість інтеграції NIDS з уже існуючими системами безпеки. | Спростити процес впровадження та зменшити витрати на адаптацію. |
| Відкритий вихідний код | Доступність для спільноти та можливість кастомізації. | Залучення більшої кількості користувачів та розробників для підтримки системи. |
| Сильна спільнота користувачів | Активна підтримка та обмін досвідом між користувачами. | Швидке вирішення проблем та обмін знаннями, що підвищує надійність системи. |

Результатом даного підрозділу є система рішень щодо ринкової поведінки компанії, вона визначає в якому напрямі буде працювати компанія на ринку задля успішного виходу і захоплення аудиторії.

## Висновки до розділу 4

Глибокий аналіз ринку підтвердив високий попит на інноваційні рішення в сфері кібербезпеки, особливо в сегменті SDN. Зростання кількості та складності кіберзагроз створює значний попит на ефективні системи захисту. Запропонований продукт CyberGuard, завдяки своїм унікальним характеристикам та перевагам, зможе завоювати значну частку ринку та задовольнити потреби наших клієнтів.

Запропонований стартап-проект, спрямований на просування NIDS (систем виявлення вторжень в мережі), пройшов глибокий аналіз ринку та конкурентного середовища. У ході дослідження були детально вивчені ключові гравці на ринку NIDS, їхні пропозиції, сильні та слабкі сторони. Цей всебічний аналіз дозволив нам чітко визначити ніші та можливості для нашого продукту.

# ВИСНОВКИ

У роботі було здійснено комплексне дослідження технологій програмно-визначених мереж та можливостей їх поєднання з сучасними підходами глибокого навчання для оптимізації управління мережею, забезпечення її безпеки та підвищення надійності виявлення кіберзагроз.

Основною концептуальною особливістю SDN є розділення площини керування і площини передачі даних, що дозволяє досягти нових рівнів гнучкості та зручності управління мережею. Контролер SDN зосереджує інформацію про всі мережеві пристрої, централізовано здійснюючи завдання управління, такі як маршрутизація та комутація, що спрощує адміністрування мережі та знижує залежність від апаратних обмежень кожного окремого елементу мережі. Використання протоколу OpenFlow дозволяє стандартизувати управління потоками трафіку через різноманітні мережеві пристрої, незалежно від їхнього виробника, що сприяє інтеграції гетерогенних апаратних платформ в єдину цілісну систему. Це забезпечує не лише простоту налаштування, але й можливість гнучкого перенаправлення трафіку відповідно до заданих політик безпеки та ефективності, що є важливим для забезпечення стійкості мережі до змінних умов навантаження.

Також аналіз технології глибокого навчання, показало його значний потенціал у вирішенні складних завдань, пов'язаних із розпізнаванням образів, обробкою сигналів і генерацією послідовних даних. Технологія глибокого навчання здатна автоматично виділяти ознаки з необроблених даних, що скорочує витрати на розробку та підвищує загальну продуктивність систем. Це підтверджує необхідність інтеграції технологій глибокого навчання в SDN для підвищення ефективності систем управління трафіком.

Аналіз наукової літератури свідчить про динамічний розвиток підходів забезпечення безпеки SDN, зокрема, шляхом впровадження інноваційних алгоритмів глибокого навчання для виявлення і нейтралізації кіберзагроз, таких як DDoS-атаки. Програмно-визначені мережі стикаються з низкою безпекових

викликів, які потребують нових підходів для їхньої повної реалізації. Дослідження показують, що протокол OpenFlow, хоч і ефективно забезпечує контроль потоків трафіку, містить певні вразливості, які потребують подальших розробок для їхньої оптимізації. Тому забезпечення захисту мережі має враховувати не тільки ефективні алгоритми виявлення загроз, а й вдосконалені методи автентифікації, авторизації та шифрування.

Запропонована системи виявлення вторгнень, що побудована на базі алгоритмів ML та DL, значно підвищують здатність SDN забезпечувати безпеку мережі. Ця система не лише може виявляти вторгнення в режимі реального часу, але й ефективно розпізнають нові, раніше невідомі загрози. Особливістю запропонованої системи є те, що вона базуються на наборі даних NSL-KDD, який дозволяє моделювати різноманітні сценарії атак.

Окрему увагу в дослідженні приділено проблемі виявлення та запобігання DDoS-атакам. Ці атаки становлять суттєву загрозу для SDN, адже вони можуть вивести з ладу будь яку захищену мережу. Використання методів інформаційної ентропії та машинного навчання дозволяє ефективно аналізувати структуру трафіку, виявляючи аномальні патерни й оперативно реагуючи на можливі загрози. Це відкриває нові можливості для створення стійких і гнучких SDN-систем, здатних працювати в умовах високих навантажень і підвищених вимог до безпеки.

Отже, результати дослідження показали високу перспективність використання технології DL в програмно-визначених мережах для забезпечення ефективного управління, надійного моніторингу трафіку та захисту від кіберзагроз. Застосування DL дозволяє створювати адаптивні системи, здатні динамічно змінювати конфігурацію мережі та реагувати на можливі загрози у реальному часі. Це особливо важливо в контексті розвитку мереж наступних поколінь (5G, IoT), де забезпечення безпеки й якості обслуговування стає визначальним фактором для функціонування критично важливих інфраструктур.

Проведений аналіз доводить необхідність подальших досліджень у галузі інтеграції сучасних алгоритмів DL в SDN з метою побудови масштабованих, надійних і стійких мережевих рішень, які здатні адаптуватися до умов сучасного кіберпростору.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кононова І.В., Могилевич В.Д. (2023). Комплексне використання програмно-визначеної мережі та глибокого навчання для виявлення атак. Матеріали VI науково-практичної конференції курсантів (студентів), аспірантів, докторантів та молодих учених "Актуальні питання застосування Спеціальних інформаційно-комунікаційних систем"

2. Haji, Saad & Zeebaree, Subhi & Saeed, Rezgar & Ameen, Siddeeq & Shukur, Hanan & Omar, Naaman & M. Sadeeq, Mohammed & Ageed, Zainab & Mahmood, Ibrahim & Yasin, Hajar. (2021). Comparison of Software Defined Networking with Traditional Networking. Asian Journal of Computer Science and Information Technology. 9. 1-18. 10.9734/AJRCOS/2021/v9i230216. https://www.researchgate.net/publication/351936824

3. Гніденко М.П., Вишнівський В.В., Ільїн О.О. Побудова SDN мереж. – Навчальний посібник. – Київ: ДУТ, 2019. – 190 с.

4. Уривський Л.О., Могилевич В.Д. (2024). АНАЛІЗ АТАК В ПРОГРАМНО-КЕРОВАНИХ МЕРЕЖАХ. Збірник матеріалів Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ», 46–49. https://conferenc-journal.its.kpi.ua/article/view/301994

5. Кононова І.В., Могилевич В.Д. (2024). Аналіз вразливостей протоколів комутації у програмно-визначених мережах. Матеріали III Міжнародної науково-практичної конференції "Кібербезпека державних інституцій та подолання кризових станів" (Том 1)

6. Admassu, Tsehay. (2020). Software defined network emulation with OpenFlow protocol. International Journal of Advances in Applied Sciences. 9. 70. 10.11591/ijaas.v9.i1.pp70-76. https://www.researchgate.net/publication/339708791

7. Hussain M, Shah N, Amin R, Alshamrani SS, Alotaibi A, Raza SM. Software-Defined Networking: Categories, Analysis, and Future Directions. *Sensors*. 2022; 22(15):5551. https://doi.org/10.3390/s22155551

8. Al Somaidai, Mohammed. (2014). Survey of Software Components to Emulate OpenFlow Protocol as an SDN Implementation. American Journal of Software Engineering and Applications. 3. 10.11648/j.ajsea.20140306.12.

9. Дашкевич А.О.. Дослідження багатошарових нейронних мереж для автоматичного виділення ознак при вирішенні задачі розпізнавання образів. Науковий вісник ТДАТУ, 6 (2), с.134-139: 2016.

10. Лебідь О., Кіпоренко С., Вовк В.. Виявлення кібератак та підвищення інформаційної безпеки на основі технології нейронних мереж в умовах кібервійни. Наука і техніка сьогодні, №1 (15). - Київ: 2023.

11. Шемет С.В. Застосування згорткової нейронної мережі для обробки та аналізу МРТ-зображень. Електронний ресурс: https://openarchive.nure.ua/server/api/core/bitstreams/446c21ec-457f-433c-919d-e49c7b1ca8dd/content

12. Das, S., Tariq, A., Santos, T., Kantareddy, S.S., Banerjee, I. (2023). Recurrent Neural Networks (RNNs): Architectures, Training Tricks, and Introduction to Influential Research. In: Colliot, O. (eds) Machine Learning for Brain Disorders. Neuromethods, vol 197. Humana, New York, NY. https://doi.org/10.1007/978-1-0716-3195-9_4

13. Li Q, Liu Y, Shang Y, Zhang Q, Yan F. Deep Sparse Autoencoder and Recursive Neural Network for EEG Emotion Recognition. Entropy (Basel). 2022 Aug 25;24(9):1187. doi: 10.3390/e24091187. PMID: 36141073; PMCID: PMC9497873.

14. Liu, Shuanglong & Zhang, Chao & Ma, Jinwen. (2016). Stacked Auto-Encoders for Feature Extraction with Neural Networks. 377-384. 10.1007/978-981-10-3611-8_31.

15. Ali Haider, Arman Rasool. Security Issues and Challenges in SDN. Advances in Cyber Security. 2021. https://researchgate.net/publication/356742762

16. Maham Iqbal, Farwa Iqbal, Fatima Mohsin, Muhammad Rizwan, Fahad Ahamd. Security Issues in Software Defined Networking (SDN): Risks, Challenges and Potential Solutions. 2019. https://www.researchgate.net/publication/338019274

17. Farooq M.S., Riaz S., Alvi, A. Security and Privacy Issues in Software-Defined Networking (SDN): A Systematic Literature Review. Electronics 2023, 12, 3077. https://doi.org/10.3390/electronics12143077

18. Ayodele B., Buttigieg V. SDN as a defence mechanism: a comprehensive survey. Int. J. Inf. Secur. 23, 141–185 (2024). https://doi.org/10.1007/s10207-023-00764-1

19. Pulkit Ohri, Subhrendu Guha Neogi. Software-Defined Networking Security Challenges and Solutions: A Comprehensive Survey. 2022. https://dx.doi.org/10.12785/ijcds/120131

20. Hussein A., Louma Chadad, Nareg Adalian, Ali Chehab, Imad H. Elhajj, and Ayman Kayssi. 2019. Software-Defined Networking (SDN): The Security Review. Journal of Cyber Security Technology 4 (1): 1–66. doi:10.1080/23742917.2019.1629529.

21. M. B. Jiménez, D. Fernández, J. E. Rivadeneira, L. Bellido and A. Cárdenas, A Survey of the Main Security Issues and Solutions for the SDN Architecture, in IEEE Access, vol. 9, pp. 122016-122038, 2021, doi: 10.1109/ACCESS.2021.3109564

22. Shaik Razia, Venkata Ramani Varanasi. Intrusion Detection using Machine Learning and Deep Learning. International Journal of Recent Technology and Engineering (IJRTE) 8(4). 2019. doi:10.35940/ijrte.D9999.118419

23. Hindy H. Intrusion Detection Systems using Machine Learning and Deep Learning techniques. 2021. https://rke.abertay.ac.uk/en/studentTheses

24. Emad-ul-Haq Qazi, Muhammad Imran, Noman Haider, Muhammad Shoaib, Imran Razzak. An intelligent and efficient network intrusion detection system using deep learning. Computers and Electrical Engineering. Volume 99. 2022. 107764. ISSN 0045-7906. https://doi.org/10.1016/j.compeleceng.2022.107764

25. Asharf Javed, Nour Moustafa, Hasnat Khurshid, Essam Soliman Debie, Waqas Haider and Abdul Wahab. A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions. Electronics (2020): n. pag. DOI:10.3390/electronics9071177

26. Alotaibi A, Rassam M.A. Adversarial Machine Learning Attacks against Intrusion Detection Systems: A Survey on Strategies and Defense. Future Internet. 2023; 15(2):62. https://doi.org/10.3390/fi15020062

27. Thanh-Huy Nguyen, Van Son Nguyen, Thai Thanh Tung, Tran Tien Dung, Le Thi Thanh Thuy, Nguyen Liem Hieu, Nguyen Minh Dung, Nguyen Van Ba. USING MACHINE LEARNING AND DEEP LEARNING TO IMPROVE ANOMALY ATTACK. 2023. https://doi.org/10.35741/issn.0258-2724.58.4.40

28. Mohammed Mynuddin, Sultan Uddin Khan, Zayed Uddin Chowdhury, Foredul Islam, Md Jahidul Islam, Mohammad Iqbal Hossain, Dewan Mohammed Abdul Ahad. Automatic Network Intrusion Detection System Using Machine Learning and Deep Learning. 2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS), Bandung, Indonesia, 2024, pp. 1-9, doi: 10.1109/AIMS61812.2024.10512607

29. Madhusudhan R., Thakur S.K., Pravisha P. (2024). Enhancing Intrusion Detection System Using Machine Learning and Deep Learning. In: Barolli, L. (eds) Advanced Information Networking and Applications. AINA 2024. Lecture Notes on Data Engineering and Communications Technologies, vol 201. Springer, Cham. https://doi.org/10.1007/978-3-031-57870-0_29

30. Chandrapal Singh, Ankit Kumar Jain. A comprehensive survey on DDoS attacks detection & mitigation in SDN-IoT network. e-Prime - Advances in Electrical Engineering, Electronics and Energy. Volume 8. 2024. https://doi.org/10.1016/j.prime.2024.100543

31. Fan Cong, Nitheesh Murugan Kaliyamurthy, Shi Chen, He Jiang, Yiwen Zhou, and Carlene Campbell. 2022. Detection of DDoS Attacks in Software Defined Networking Using Entropy Applied Sciences 12, no. 1: 370. https://doi.org/10.3390/app12010370

32. Oyucu Saadin, Onur Polat, Muammer Türkoğlu, Hüseyin Polat, Ahmet Aksöz, and Mehmet Tevfik Ağdaş. 2024. Ensemble Learning Framework for DDoS Detection in SDN-Based SCADA Systems Sensors 24, no. 1: 155. https://doi.org/10.3390/s24010155

33. T. Alasali, O. Dakkak. Exploring the landscape of SDN-Based DDoS Defense: A Holistic Examination of Detection and Mitigation Approaches, Research Gaps and Promising Avenues for Future Exploration, Ijanser, Vol. 7, No. 4, Pp. 327–349, May 2023. https://doi.org/10.59287/ijanser.726

34. Songa A.V., Karri G.R. An integrated SDN framework for early detection of DDoS attacks in cloud computing. J Cloud Comp 13, 64 (2024). https://doi.org/10.1186/s13677-024-00625-9

35. Priyanka Kujur, Subhra Priyadarshini Biswal, Sanjeev Patel. Security Challenges and Analysis for SDN-Based Networks. Software Defined Networks: Architecture and Applications. 2022 https://doi.org/10.1002/9781119857921.ch10

36. Kumar, Satish & Sunanda, & Arora, Sakshi. (2020). A Statistical Analysis on KDD Cup'99 Dataset for the Network Intrusion Detection System. 10.1007/978-981-15-3852-0_9

37. Zehui Zhao, Laith Alzubaidi, Jinglan Zhang, Ye Duan, Yuantong Gu. A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations. Expert Systems with Applications. Volume 242. 2024. https://doi.org/10.1016/j.eswa.2023.122807

38. S. J. Stolfo, Wei Fan, Wenke Lee, A. Prodromidis, and P. K. Chan. Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project. In DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings, volume 2, pages 130–144 vol.2, 2000

39. R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman. Evaluating Intrusion Detection Systems: the 1998 DARPA Off-line Intrusion Detection Evaluation. In DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings, volume 2, pages 12–26 vol.2, 2000

40. Özgür, Atilla & Erdem, Hamit. (2016). A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. 10.7287/PEERJ.PREPRINTS.1954.

41. Hesford, J., Cheng, D., Wan, A., Huynh, L., Kim, S., Kim, H., & Hong, J.B. (2024). Expectations Versus Reality: Evaluating Intrusion Detection Systems in Practice. *ArXiv, abs/2403.17458*.

42. Singh, Anshuman & Gupta, Brij. (2022). Distributed Denial-of-Service (DDoS) Attacks and Defense Mechanisms in Various Web-Enabled Computing Platforms: Issues, Challenges, and Future Research Directions. International Journal on Semantic Web and Information Systems. 18. 1-43. 10.4018/IJSWIS.297143.

43. Cisco Systems. (n.d.). Cisco IOS NetFlow. https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html

44. Muhammad, Afaq & Rehman, Shafqat & Song, Wang-Cheol. (2015). Large Flows Detection, Marking, and Mitigation based on sFlow Standard in SDN. Journal of Korea Multimedia Society. 18. 189-198. 10.9717/kmms.2015.18.2.189.

45. Prashant Kumar Shukla, Priti Maheshwary, E.K. Subramanian, V. Jean Shilpa, P. Ravi Kiran Varma. Traffic flow monitoring in software-defined network using modified recursive learning. Physical Communication. Volume 57. 2023. https://doi.org/10.1016/j.phycom.2022.101997.

46. He, Daojing & Chan, Sammy & Ni, Xiejun & Guizani, Mohsen. (2017). Software-Defined-Networking-Enabled Traffic Anomaly Detection and Mitigation. IEEE Internet of Things Journal. PP. 1-1. 10.1109/JIOT.2017.2694702.

47. Tcpdump. (n.d.). *Tcpdump*. http://www.tcpdump.org

48. Bit-Twist. (n.d.). *Bit-Twist*. http://bittwist.sourceforge.net

49. Tcpreplay. (n.d.). *Tcpreplay*. http://tcpreplay.synfin.net

50. Hping3. (n.d.). *Hping3*. http://wiki.hping.org

51. Sadhwani, Sapna, Baranidharan Manibalan, Raja Muthalagu, and Pranav Pawar. 2023. "A Lightweight Model for DDoS Attack Detection Using Machine Learning Techniques" *Applied Sciences* 13, no. 17: 9937. https://doi.org/10.3390/app13179937

52. Гніденко М.П., Вишнівський В.В., Ільїн О.О. Побудова SDN мереж. – Навчальний посібник. – Київ: ДУТ, 2019. – 190 с.

53. C., Bai J. Sun Q Software-Defined Wide Area Network Architectures and Technologies: training manual. CRC Press, 2013,460 p.

54. Abdulsamad A.A., Salih, T.A. IoT security improvement based on SDN Controller. Eurasian Journal of Engineering and Technology, 2023. № 14. 49 – 56. https://geniusjournals.org/index.php/ejet/article/view/3199

55. Aayush Pradhan, Rejo Mathew, Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN),Procedia Computer Science, Volume 171, 2020,Pages 2581-2589,ISSN 1877-0509, https://doi.org/10.1016/ j.procs.2020.04.280

56. Ali Nadim Alhaj, Nitul Dutta. Analysis of Security Attacks in SDN Network: A Comprehensive Survey. Contemporary Issues in Communication, Cloud and Big Data Analytics, 2022. 27–37. https://researchgate.net/ publication/356688838

57. P. Karthika, Dr. A. Karmel. Analysis of Different Attacks on Software Defined Network and Approaches to Mitigate using Intelligent Techniques. International Journal of Advanced Computer Science and Applications 12(9), 2021. https://www.researchgate.net/publication/355085556

58. Anmol Mahajan, Abhinav Bhandari. Attacks in Software-Defined Networking: A Review. Proceedings of the International Conference on Innovative Computing & Communications (ICICC) , 2020. https://papers.ssrn.com/ sol3/papers.cfm? abstract_id=3564048

59. Anh Tuan Phu, Bo Li, Faheem Ullah, Tanvir Ul Huque, Ranesh Naha, Muhammad Ali Babar, Hung Nguyen. Defending SDN against packet injection attacks using deep learning. Computer Networks Volume 234, 2023. https://www.sciencedirect.com/ science/article/ pii/S1389128623003808

60. Jin Wang, Liping Wang. SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN. Sensors, 2022. https://www.mdpi.com/1424-8220/22/21/8287

61. Farooq MS, Riaz S, Alvi A. Security and Privacy Issues in Software-Defined Networking (SDN): A Systematic Literature Review. *Electronics*. 2023; 12(14). https://doi.org/10.3390/electronics12143077

62. Jagdeep S., Sunny B..Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions, Computer Science Review, Volume 37, 2020. sciencedirect.com/science/article/pii/S1574013720301647

63. Alsaghier, H. M. (2019). Attack on sdn infrastructure and security measures. Journal of Engineering and Applied Sciences, 1-17. https://doi.org/10.5455/jeas.2019090101

64. Yassine Maleh, Youssef Qasmaoui, Khalid El Gholami, Yassine Sadqi, Soufyane Mounir. A comprehensive survey on SDN security: threats, mitigations,and future directions. Journal of Reliable Intelligent Environments, 2022. https://www.researchgate.net/ publication/358380811

65. Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega. Security in SDN: A comprehensive survey. Journal of Network and Computer Applications Volume 159, 2020. https://sciencedirect.com/science/article/abs/pii/S1084804520300692

66. Алєксєєв М.О., Сінько В.В., Могилевич В.Д. (2024). Оцінка впливу атак в програмно-керованих мережах. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. https://www.tech.vernadskyjournals.in.ua/archive?id=136

67. Розроблення стартап-проекту: Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ: НТУУ «КПІ», 2016. – 28 с.

68. Основи виробничого підприємництва: Навчальний посібник / під ред. Підлісної О.А., Янкового В.В. –К.: ІВЦ «Видавництво Політехніка», НТУУ«КПІ», 2010. – 287 с

69. Економіка підприємства: курс лекцій: у 2 кн./ під заг. ред. П.В. Круша, К.В. Шелехова. -К.: НТУУ «КПІ», 2012. – Кн.2. Теорія і практика господарювання. – Ч.1. – 280 с. – Бібліогр.: у кінці тем.

70. Економіка підприємства: курс лекцій: у 2 кн./ під заг. ред. П.В. Круша, К.В. Шелехова. -К.: НТУУ «КПІ», 2012. – Кн.2. Теорія і практика господарювання. – Ч.2. – 342 с. – Бібліогр.: у кінці тем.

# ДОДАТОК А

У рамках магістерської дисертації була розроблена модель системи виявлення мережевих вторгнень (NIDS) за допомогою Python. Основним інструментом для навчання моделі був широко використовуваний датасет NSL-KDD, який є розширеною версією оригінального набору даних KDD Cup 1999 і містить покращену якість даних для тренування систем кібербезпеки. Датасет NSL-KDD містить різноманітні зразки нормальної та аномальної поведінки в мережі, зокрема атаки різних типів, що дозволяє моделі навчитися розрізняти трафік.

Після етапу тренування моделі, була виконана симуляція атаки типу "відмова в обслуговуванні" (DoS). Це одна з найпоширеніших форм кіберзлочинів, яка спрямована на перевантаження системи чи сервера великою кількістю запитів, що унеможливлює нормальне функціонування. Використовуючи інструмент Wireshark, можна проаналізувати мережеві пакети під час цієї атаки, візуалізуючи її вплив на мережу.

Навчена модель NIDS повинна автоматично виявляти ознаки DoS-атаки та видавати відповідні сповіщення, що є ключовою функцією систем захисту мереж.

Для реалізації системи виявлення мережевих вторгнень на Python, було обрано кілька ключових бібліотек, які допоможуть виконати різні завдання, від роботи з даними до аналізу мережевого трафіку та побудови моделі машинного навчання.

Pandas — це бібліотека для роботи з даними, яка надає зручні інструменти для зберігання, обробки та аналізу великих наборів даних. Ми будемо використовувати Pandas для завантаження датасету NSL-KDD, його попередньої обробки (очищення, нормалізація, перетворення категоріальних даних у числові формати тощо) та підготовки до навчання моделі машинного навчання. Завдяки Pandas, можна легко виконувати операції над стовпцями та рядками даних.

Scikit-learn — це одна з найпопулярніших бібліотек для машинного навчання на Python. Вона надає широкий набір алгоритмів класифікації, регресії та кластеризації. Scikit-learn використовується для побудови та тренування моделі виявлення атак. Ми застосуємо алгоритм класифікації для навчання моделі на основі NSL-KDD датасету. Крім того, ця бібліотека пропонує корисні інструменти для розбиття даних на навчальні та тестові вибірки, стандартизації та оцінки точності моделі.

Scapy – це бібліотека для роботи з мережевими пакетами, яка дозволяє з легкістю створювати, відправляти, захоплювати та аналізувати мережеві пакети. Ми використовуватимемо Scapy для симуляції DoS-атаки шляхом генерування великої кількості запитів до сервера, що може спричинити перевантаження мережі. Окрім того, Scapy дозволить нашій системі NIDS перехоплювати мережеві пакети в режимі реального часу для аналізу трафіку та виявлення потенційних атак.

NumPy – це бібліотека для наукових обчислень на Python, яка пропонує багатовимірні масиви та функції для їх обробки. Навіщо вона потрібна: NumPy забезпечить швидкі та ефективні обчислення при обробці великих обсягів даних. Наприклад, її масиви можуть використовуватися для векторизації операцій у моделі машинного навчання або при обробці наборів даних під час їхньої підготовки.

Time – це стандартна бібліотека Python для роботи з часом. Бібліотека Time допоможе вимірювати час виконання різних операцій, що може бути корисним при моніторингу швидкості обробки пакетів або під час вимірювання продуктивності моделі та системи виявлення вторгнень загалом.

Опрацювання датасету та тренування моделі є одним із ключових етапів створення системи виявлення мережевих вторгнень. Ми використовуємо алгоритм логістичної регресії для класифікації, тренуємо модель на навчальних даних та перевіряємо її точність на тестових даних.

Створюємо модель логістичної регресії з максимальною кількістю ітерацій 2000

model = LogisticRegression(max_iter=2000)

Тренуємо модель на навчальних даних (X_train - ознаки, y_train - цільові мітки)

model.fit(X_train, y_train)

Прогнозуємо мітки для тестових даних (X_test)

y_pred = model.predict(X_test)

Виводимо звіт про якість класифікації, порівнюючи передбачення з реальними мітками (y_test)

print(classification_report(y_test, y_pred))

Далі ми запускаємо аналіз мережевого трафіку за допомогою спеціальної функції packet_callback(packet), яка була створена для моніторингу та аналізу кожного перехопленого пакету. Ця функція служить основним механізмом для виявлення потенційних атак у режимі реального часу. Вона обробляє кожен отриманий пакет, перевіряючи його на наявність підозрілої активності або ознак шкідливої атаки. Якщо функція ідентифікує пакет, який відповідає критеріям загрози (наприклад, DoS-атака), вона виведе відповідне попередження, сповіщаючи користувача про виявлену атаку.

Потім, необхідно провести симуляцію атаки.

```
import time                # Імпортуємо бібліотеку time для роботи з часом
from scapy.all import *    # Імпортуємо всі модулі з бібліотеки Scapy для роботи з мережевими пакетами
def syn_flood(target_ip, target_port, duration):
    """
    Функція для проведення атаки.

    Параметри:
    target_ip (str): IP-адреса цілі, на яку буде надіслано трафік.
    target_port (int): Порт цілі, на якому відбуватиметься атака.
    duration (int): Тривалість атаки в секундах.
    """
```

start_time = time.time()  # Запам'ятовуємо час початку атаки

while time.time() - start_time < duration:  # Цикл, що виконується протягом визначеного часу

ip = IP(dst=target_ip)  # Створюємо IP-пакет, вказуючи IP-адресу цілі

tcp = TCP(sport=RandShort(), dport=target_port, flags="S")  # Створюємо TCP-пакет з випадковим вихідним портом та встановленим прапорцем "SYN"

send(ip/tcp, verbose=False)  # Надсилаємо пакет без виведення інформації в консоль

print("Атака зупинилася через", duration, "секунд")  # Повідомляємо, коли атака закінчилася

# Викликаємо функцію з конкретними параметрами

syn_flood("111.111.1.11", 8888, 20)  # Заміна "111.111.11.11" на IP-адресу вашої цілі та 8888 на порт, на який буде здійснено атаку, тривалість атаки - 20 секунд

Приклад виконання:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.63 | 0.93 | 0.75 | 9711 |
| 1 | 0.92 | 0.59 | 0.72 | 12833 |
| accuracy | | | 0.74 | 22544 |
| macro avg | 0.77 | 0.76 | 0.74 | 22544 |
| weighted avg | 0.79 | 0.74 | 0.73 | 22544 |

Звіт про класифікацію моделі для оцінки її ефективності. Він містить ключові метрики, такі як точність (precision), повнота (recall), F1-міра (f1-score) та підтримка (support) для двох класів: `0` (нормальний трафік) і `1`(атака). Ось розбір метрик:

Клас 0:

- Точність: 0.63 – З усіх передбачених класів 0, 63% виявились правильними.

- Повнота: 0.93 – З усіх фактичних класів 0, 93% було передбачено правильно.

- F1-міра: 0.75 – Гармонічне середнє точності та повноти.

- Підтримка: 9711 – Кількість фактичних зразків класу 0.

Клас 1:

- Точність: 0.92 – З усіх передбачених класів 1, 92% виявились правильними.

- Повнота: 0.59 – З усіх фактичних класів 1, 59% було передбачено правильно.

- F1-міра: 0.72 – Гармонічне середнє точності та повноти.

- Підтримка: 12833 – Кількість фактичних зразків класу 1.

Інші метрики:

- Точність (accuracy): 0.74 – Загальна точність моделі, що показує відсоток правильних передбачень.

- Середня макро-міра (macro avg): 0.77 для точності, 0.76 для повноти, 0.74 для F1-міри – це середні значення по всіх класах, без урахування дисбалансу між класами.

- Зважене середнє (weighted avg): 0.79 для точності, 0.74 для повноти, 0.73 для F1-міри – це середні значення, зважені відповідно до кількості зразків у кожному класі.

Загальний результат моделі: точність 0.74, що означає, що модель правильно передбачила приблизно 74% зразків.

Ввімкнений аналізатор трафіка:

```
>>> scapy.sniff(iface="Беспроводная сеть", prn=packet_callback, store=False)
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []
```

Нормальний трафік:

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 12 | 3.018782 | Routerboardc_80:40:… | Intel_2a:3b:f9 | ARP | 42 | 192.168.88.1 is at 48:8f:5a:80:40:93 |
| 13 | 3.292570 | 192.168.88.79 | 77.74.181.36 | TCP | 66 | 1148 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM |
| 14 | 4.008437 | Intel_2a:3b:f9 | Broadcast | ARP | 42 | Who has 192.168.88.42? Tell 192.168.88.79 |
| 15 | 4.301846 | 192.168.88.79 | 77.74.181.36 | TCP | 66 | [TCP Retransmission] 1148 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM |
| 16 | 4.661204 | 192.168.88.1 | 255.255.255.255 | MNDP | 169 | 5678 → 5678 Len=127 |
| 17 | 4.882125 | Intel_2a:3b:f9 | Broadcast | ARP | 42 | Who has 192.168.88.42? Tell 192.168.88.79 |
| 18 | 5.339399 | 192.168.88.79 | 192.168.88.1 | DHCP | 358 | DHCP Request - Transaction ID 0x95ccaddb |
| 19 | 5.362407 | 192.168.88.1 | 192.168.88.79 | DHCP | 342 | DHCP ACK - Transaction ID 0x95ccaddb |
| 20 | 5.411029 | 192.168.88.79 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Leave group 224.0.0.252 |
| 21 | 5.587981 | 192.168.88.79 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.252 for any sources |
| 22 | 5.598756 | 192.168.88.79 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Leave group 224.0.0.252 |
| 23 | 5.599200 | 192.168.88.79 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.252 for any sources |
| 24 | 5.614527 | 192.168.88.79 | 224.0.0.251 | MDNS | 81 | Standard query 0x0000 ANY LAPTOP-AP6OR3EM.local, "QM" question |
| 25 | 5.616683 | 192.168.88.79 | 224.0.0.252 | LLMNR | 75 | Standard query 0x1332 ANY LAPTOP-AP6OR3EM |
| 26 | 5.617580 | 192.168.88.79 | 224.0.0.251 | MDNS | 91 | Standard query response 0x0000 A 192.168.88.79 |
| 27 | 5.624058 | 192.168.88.79 | 224.0.0.251 | MDNS | 81 | Standard query 0x0000 ANY LAPTOP-AP6OR3EM.local, "QM" question |
| 28 | 5.624940 | 192.168.88.79 | 224.0.0.251 | MDNS | 91 | Standard query response 0x0000 A 192.168.88.79 |
| 29 | 5.632797 | 192.168.88.79 | 239.255.255.250 | SSDP | 179 | M-SEARCH * HTTP/1.1 |
| 30 | 5.797524 | 192.168.88.79 | 77.74.181.141 | TCP | 66 | 1149 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM |
| 31 | 5.877216 | Intel_2a:3b:f9 | Broadcast | ARP | 42 | Who has 192.168.88.42? Tell 192.168.88.79 |
| 32 | 5.877462 | 192.168.88.79 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.252 for any sources |
| 33 | 6.802341 | 192.168.88.79 | 77.74.181.141 | TCP | 66 | [TCP Retransmission] 1149 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM |
| 34 | 6.812673 | 192.168.88.79 | 20.199.120.85 | TLSv1.2 | 154 | Application Data |
| 35 | 6.872240 | 20.199.120.85 | 192.168.88.79 | TLSv1.2 | 225 | Application Data |
| 36 | 6.923071 | 192.168.88.79 | 20.199.120.85 | TCP | 54 | 1401 → 443 [ACK] Seq=101 Ack=172 Win=516 Len=0 |
| 37 | 7.047644 | 192.168.88.79 | 62.128.100.49 | TCP | 66 | 1150 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM |

Запускаємо атаку:

```
>>> syn_flood("192.168.88.79", 8080, 20)
Атака зупинилася через 20 секунд
```

Вивід програми:

```
[16 rows x 18 columns]
D:\Python311\Lib\site-packages\sklearn\base.py:486: UserWarning: X has feature names, but LogisticRegression was fitted without feature names
  warnings.warn(
УВАГА! Бачу атаку
    service  flag  logged_in  count  srv_count  ...  dst_host_diff_srv_rate  dst_host_same_src_port_rate  dst_host_srv_diff_host_rate  dst_host_serror_rate  dst_host_srv_serror_rate
0       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
1       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
2       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
3       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
4       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
5       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
6       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
7       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
8       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
9       0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
10      0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
11      0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
12      0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
13      0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
14      0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
15      0.8   0.8        0.8    0.8        0.8  ...                     0.2                          0.2                          0.2                   0.2                      0.2
```

Атака виявлена!