Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему
Метод виявлення шкідливих пакетів та DDoS атак на основі аналізу мережевого трафіку з використанням глибоких згорткових нейронних мереж

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

КРМКБ.220188.22.01.16 ПЗ

Виконав: студент 2 курсу, група КБм-22-1 Майор Є.В.
Підпис

Керівник доц., к.т.н, доцент Джулій В.М.
Підпис

Нормоконтролер старший викладач Мостовий С.В.
Підпис

До захисту допускаю:
Зав. кафедри кібербезпеки, к.т.н., доц Кльоц Ю.П.
Підпис

15 грудня 2023 р.

Хмельницький, 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Освітній рівень Магістр

Галузь знань 12 Інформаційних технологій

Спеціальність 125 Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

"30" 08 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Майор Євгену Віталійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод виявлення шкідливих пакетів та DDoS атак на сонові аналізу мережевого трафіку з використанням глибоких згорткових нейронних мереж

Керівник роботи Джулій Володимир Миколайович

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

кандидат технічних наук, доцент

Затверджена наказом № 30 ректора університету, додаток №25 від 15.08.2023

2. Строк подання студентом проєкту (роботи) на кафедру 15.11.2023

3. Вихідні дані до проекту (роботи) Розробка методу аналізу мережевого трафіку в основні якого стоять глибокі згорткові нейронні мережі для виявлення шкідливих пакетів та DDoS-атак із подальшим машинним навчанням отриманої моделі

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Виділення проблеми шкідливих пакетів у мережевому трафіку. Аналіз глибоких згорткових мереж у питанні виялвення шкідливих пакетів. Постановка задачі дослідження. Виведення архітектури створення нейронної мережі та моделі. Системна реалізація методу. Апробація отриманих результатів. Висновки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали і посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В. Старший викладач кафедри кібербезпеки		

7. Дата видачі завдання «01» вересня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Вибір напряму дослідження і узгодження тематики КРМ з керівником	01.06.2023	
2	Ознайомлення з предметною областю; формулювання мети і задач дослідження; визначення об'єкта і предмета дослідження	04.09.2023	
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	18.09.2023	
4	Робота над розділом 2 – розробка моделей і методів для вирішення поставленої задачі	02.10.2023	
5	Робота над розділом 3 – розробка алгоритмів і технологій, їх аналіз	16.10.2023	
6	Робота над розділом 4 – апробація запропонованих рішень	06.11.2023	
7	Робота над науковою публікацією	10.11.2023	
8	Узгодження отриманих результатів, оформлення пояснювальної записки згідно вимог	15.11.2023	
9	Попередній захист роботи	17.11.2023	
10	Захист роботи на засіданні ЕК	06.12.2023	

Студент

Підпис Є.В. Майор
Ініціали, прізвище

Керівник проекту (роботи)

Підпис В.М. Джулій
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод виявлення шкідливих пакетів та DDoS-атак на основі аналізу мережевого трафіку з використанням глибоких згорткових нейронних мереж.

Автор роботи: Майор Євген Віталійович

Керівник роботи: к.т.н., доц. Джулій Володимир Миколайович

Загальний обсяг роботи: 74 сторінок, 24 рисунків, 7 таблиць, 4 додатки, 56 посилання.

Ключові слова: безпека мережі, глибокі згорткові нейронні мережі, шкідливі пакети та DDoS-атаки, методи виявлення.

Забезпечення інформаційної безпеки корпоративних мереж неможливе без систематичного моніторингу та аналізу мережевої активності з метою виявлення та блокування потенційно шкідливих дій. Цей підхід дозволяє оперативно реагувати на кібератаки та аномальну активність, що допомагає попереджати можливі інциденти та захищати ресурси мережі від небажаної втрати або пошкодження. Аналіз трафіку і виявлення загроз стають важливими елементами захисту, забезпечуючи безпеку та стійкість корпоративних мереж у сучасному цифровому середовищі.

У цій роботі розглянути загальний алгоритм проведення атак на мережу, шляхом надсилання шкідливих пакетів, методи протидії і специфіку використання глибоких згорткових нейрнонних мереж для виявлення несанкціонованого трафіку в мережі. Результатом є розроблений алгоритм побудови нейронної мережі, яка аналізує мережу і дає оцінку надходженим пакетам поступово навчаючись на цьому.

14.12.2023

ANNOTATION

Theme of qualification work: A method of detecting malicious packets and DDoS attacks on the main analysis of network traffic using deep convolutional neural networks.

Author of the work: Maior Yevhen Vitaliiovich

Mentor: Ph.D. Dzhuliy Vladimir Mykolaiovych

Total volume of work: 74 pages, 24 figures, 7 tables, 3 appendices, 56 links.

Keywords: network security, deep convolutional neural networks, malicious packets and DDoS attacks, detection methods.

Ensuring the information security of corporate networks is impossible without systematic monitoring and analysis of network activity in order to detect and block potentially malicious actions. This approach allows for rapid response to cyber attacks and anomalous activity, which helps to prevent possible incidents and protect network resources from unwanted loss or damage. Traffic analysis and threat detection are becoming important elements of protection, ensuring the security and resilience of corporate networks in the modern digital environment.

This work considers the general algorithm for carrying out attacks on the network by sending malicious packets, methods of countering them, and the specifics of using deep convolutional neural networks to detect unauthorized traffic in the network. The result is a developed algorithm for building a neural network that analyzes the network and gives an assessment of incoming packets by gradually learning from it.

11.12.2023

ЗМІСТ

ВСТУП

Враховуючи швидкість та масштабне зростання мережевих атак у сучасному світі, виявлення шкідливих пакетів та DDoS-атак стає проблемою для будь-якої організації. Це лише забезпечує безпеку мережі, а й зберігає репутацію підприємства, захищає конфіденційні дані, та забезпечує надійність і доступність сервісів для користувачів.

Через низку наступних причин постає необхідність у передчасному виявленні шкідливих пакетів для швидкої реакції на загрозу, яка може виникнути, якщо завчасно не подолати атаку:

– зупинки роботи важливих онлайн-сервісів, що призводить до великих економічних втрат для підприємств;

– виявлення шкідливих пакетів дозволяє оптимізувати пропускну здатність мережі, розподіляти ресурси та забезпечити нормальну роботу легітимних користувачів;

– шкідливі пакети можуть містити спроби витоку конфіденційної інформації. Ефективне виявлення цих пакетів забезпечує захист конфіденційних даних;

– швидке виявлення та відвертання атак допомагає уникнути великих витрат на відновлення та відшкодування збитків.

Поява нових, інноваційних методів DDoS-атак створює серйозні труднощі для існуючих методів протидії. У цьому контексті використання різних методів машинного навчання виявляє перспективи в боротьбі з DDoS-атаками. Ці методи дозволяють виявляти атаки з високою точністю і низьким рівнем помилкових спрацьовувань. Розробка та вдосконалення таких систем машинного навчання стає важливим кроком у напрямку забезпечення надійності та безпеки комп’ютерних мереж у умовах постійно зростаючого рівня загрози DDoS-атак.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Виділення проблем шкідливих пакетів та DDoS-атак у мережевому трафіку

Вивчення сучасних загроз кібербезпеці виявляє важливість ефективного виявлення шкідливих пакетів та атак DDoS у мережевому середовищі. Ця проблематика становить серйозний виклик для забезпечення безпеки мережі та збереження відповідності принципам конфіденційності, цілісності та доступності даних. Визначення ефективних методів виявлення та протидії таким загрозам вимагає комплексного дослідження, адаптованих підходів та розвитку інноваційних технологічних рішень. Вирішення цього питання відіграє критичну роль у забезпеченні безпеки мереж та зміцненні їх стійкості перед сучасними кіберзагрозами [1].

Атака DoS (Denial of Service – «відмова в обслуговуванні») – це кібератака, спрямована на систему з метою призвести її до непрацездатності, спричинити відмову у обслуговуванні. Її суть полягає в створенні умов, при яких звичайні користувачі не можуть отримати доступ до конкретних сервісів або звертатися до них ускладнено [1,2].

DDoS (Distributed Denial of) — це еволюційна форма атаки DoS, де напади відбуваються паралельно з різних пристроїв, можливо, заражених ботнетами. Основною метою є зруйнування роботи системи таким чином, щоб її послуги стали недоступними. Існують різноманітні варіанти та типи таких атак, але суть полягає в одному: зловмисник використовує вразливості для управління численними машинами [3,4]. На рисунку 1.1 можна побачити відмінність між цими типами атак.

Атаки з використанням протоколу - це категорія атак передбачає використання вразливостей мережевого протоколу для виснаження ресурсів сервера [5].

Зараз активно розвиваються нові методи та технології для виявлення та запобігання атакам DoS і DDoS. Найпопулярніші мають на меті використання штучного інтелекту, щоб аналізувати трафік та ідентифікації аномальних паттернів, що допомагає вчасно реагувати на потенційні загрози [5].

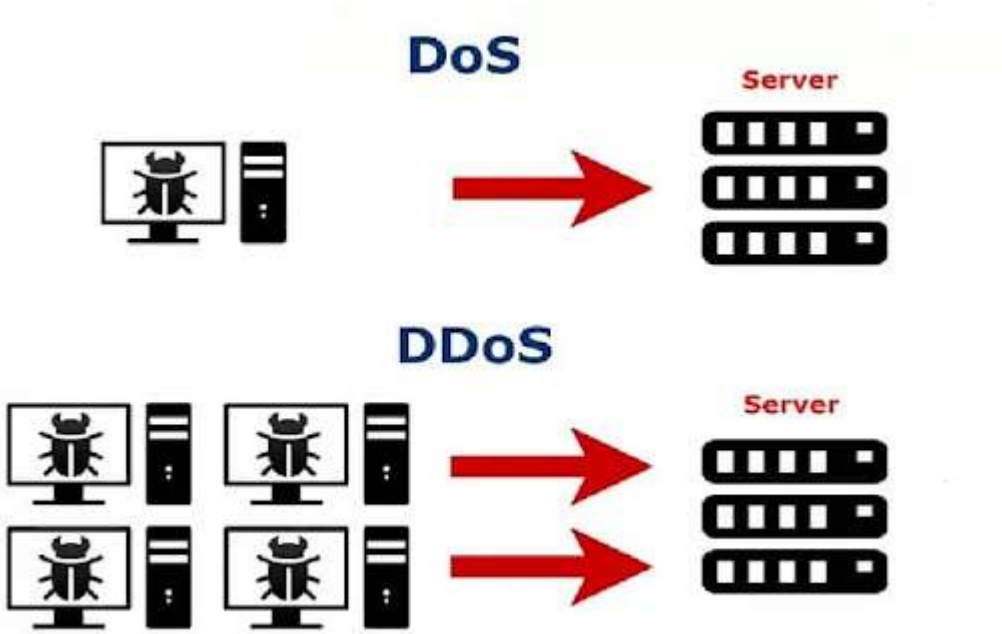


Рисунок 1.1 – Різниця між DoS і DDoS атаками

У прямих атаках зловмисник зазвичай використовує ботнет для запуску атаки, відправляючи великий потік пакетів на цільовий об'єкт. Відбивачі атаки включають в себе використання ботнетів спільно з іншими пристроями, відомими як рефлектори, для направлення шкідливого трафіку на жертву.

Атаки на основі відображення передбачають зловмисний трафік, де IP-адреса зловмисника замінюється IP-адресою жертви. Шкідливі пакети спрямовуються на інші вузли мережі, які потім надсилають свої відповіді жертві. Це дозволяє зловмиснику залишатися анонімним. Крім того, існує інша концепція під назвою «ампліфікація», де короткі запити генерують довші відповіді. Атаки з відображенням використовують це посилення, тому їх називають DDoS-атаками з відображенням. Рисунок 1.2 ілюструє процес атаки, під час якого відбивачі

спрямовують більш інтенсивний трафік до жертви порівняно з трафіком, що надсилається від зловмисника до відбивача [8].

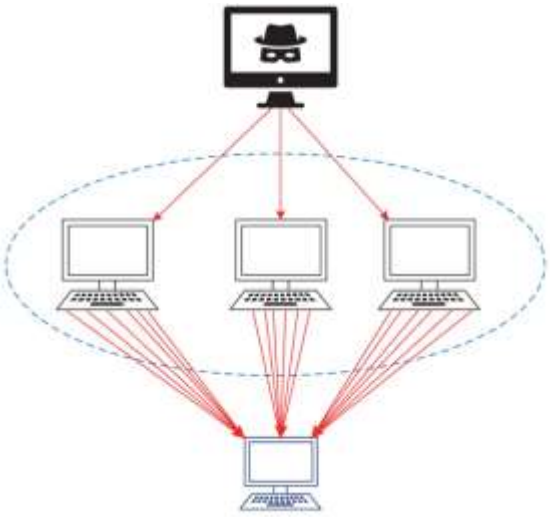


Рисунок 1.2 - Діаграма атаки розподіленої відмови в обслуговуванні (DDoS)

Зараз активно розвиваються нові методи та технології для виявлення та запобігання атакам DoS і DDoS. Найпопулярніші мають на меті використання штучного інтелекту, щоб аналізувати трафік та ідентифікації аномальних паттернів, що допомагає вчасно реагувати на потенційні загрози [5].

Типи DDoS-атак розділені на різні рівні в залежності від типу атаки позначенні на рисунку 1.3, який демонструє класифікацію атак і те який тип протоколу застосовується при їх застосуванні.

Рисунок 1.3 – Види DDoS-атак

Детальний опис різних видів DDoS-атак [11]:

– UDP атака відбувається великою кількістю UDP-пакетів перевантажують мережеву пропускну здатність, намагаючись перекрити легітімний мережевий трафік і тим самим має на меті сповільнити або спричинити зупинку цільової машини;

– ICMP надсилається велику кількість ICMP-пакетів (Ping-запити) до цільового сервера, перевантажуючи його мережевий стек;

– HTTP перевантаження ресурсів сервера обробки запитів;

– DNS використовуються вразливості в DNS-системі для відправки запитів з фальшивою адресою цільовому DNS-серверу. Відповідь на ці запити є набагато більшою, ніж запит, що збільшує мережеве навантаження;

– Slowloris спроба утримувати відкриті з'єднання з цільовим сервером, надсилаючи часткові запити та займаючи доступні ресурси сервера;

– SYN Flood надсилається багато запитів на початок рукостискання (SYN), не закінчуючи його, що призводить до заповнення черги очікування і перевантаження ресурсів сервера;

– NTP використовуються NTP-сервери для відправки запитів до цільового сервера зі збільшенням відповіді, збільшуючи тим самим обсяг трафіку, який не є необмеженим;

– SSDP використовуються SSDP-сервери для відправки фальшивих запитів зі збільшенням відповіді до цільової адреси, перевантажуючи її мережевий стек.

Атака на мережу може виглядати як велика кількість пакетів, які надходять на цільовий сервер або мережу. Ці пакети містять різні заголовки, запити та інші дані. Різниця між різними видами атак полягає у типах пакетів [10]. Ці різні типи пакетів використовують вразливості протоколів і можуть виконувати атака по декількох ключових напрямках, щоб ще більше перевантажити мережу і в більшості випадків, це виконується з великої кількості систем. Загальний алгоритм роботи DDoS-атаки зображено на рисунку 1.4.

Цей алгоритм можна поділити на 3 етапи, в залежності від складності або особливості атаки чи підготовки до атаки, ці етапи можуть відрізнятися, але загальний алгоритм залишається незіним, а отже наведений далі рисунок демонструє стандартний спосіб проведення такого типу атак.

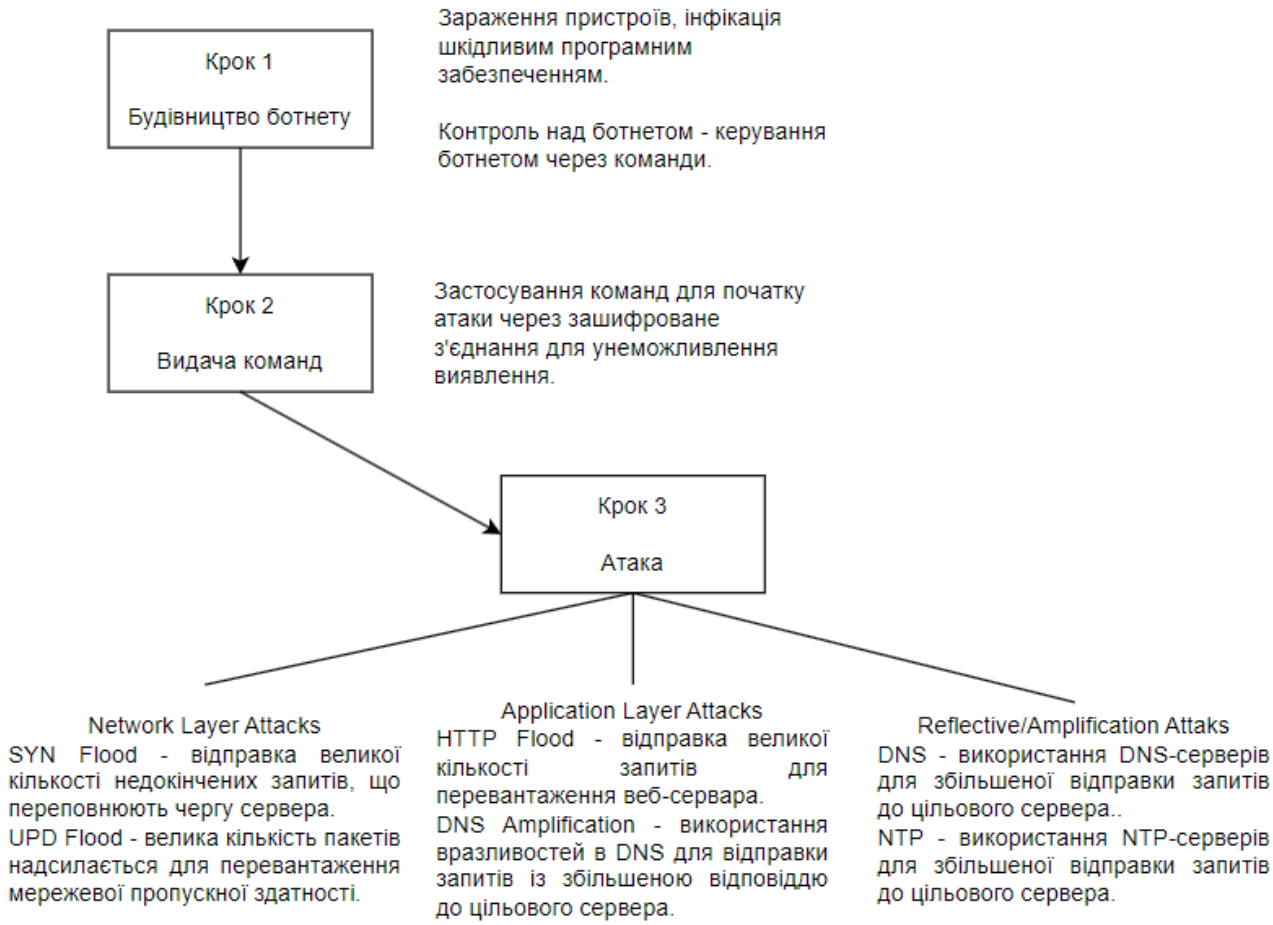


Рисунок 1.4 – Структурований опис DDoS-атак

Кожен вид атаки використовує відмінні типи запитів та заголовків для перевантаження мережевих ресурсів цільового об'єкта. Різниця полягає в тому, на якому рівні мережі чи застосунків атака спрямована та які протоколи використовуються для генерації великої кількості запитів. Ці заголовки відрізняються в залежності від типу атаки, чи це атака на мережу чи це атака з додатку, це має різну специфіку, яку потрібно розуміти і також передбачувати. Приклад заголовків та запитів у DDoS-атаках показано у таблиці 1.1

Таблиця 1.1 – Приклад заголовків і запитів під час DDoS-атак.

Атака	**Заголовок**	**Запит**
SYN Flood	Прапорець SYN (встановлення з'єднання)	Багато недокінчених TCP-запитів із встановленням з'єднання.
UDP Flood	Відправка UDP-пакетів без потреби встановлення з'єднання (нетранспортний протокол).	Велика кількість UDP-запитів, які можуть бути спрямовані на різні порти та служби.
HTTP Flood	Містить методи запитів (GET, POST тощо) та інші HTTP-параметри.	Багато HTTP-запитів, що схожі на легітімний трафік, але великою кількістю.
DNS Amplification	Містить DNS-запити з підміненою IP-адресою цільового сервера.	DNS-запити до відомих DNS-серверів зі збільшеною відповіддю, спрямовані на цільовий сервер.

Як видно, основною метод є надсилання аномально великої кількості пакетів, щоб перезавантажити мережу і тим самим спиричини збої у стистемі або отримати доступ до неї.

Приклад мережевого трафіку із DDoS-атакою з використанням UDP-флуду, такий трафік складається з великої кількості UPD-пакетів, які надсилаються з різних джерел на один порт [12]. Такі пакети не мають в собі корисного навантаження, а використовуються для перевантаження сервера показано на рисунку 1.5.

Наведений приклад UPD флуд пакету демонструє спробу виконати запит по бстандартному 80-тому порту, який в більшості випадків є відкритим і може бути підхоплений пакетом, щоб спрямовувати туда велику кількість трафіку, також як видно, змінюється стандартна IP-адреса, проходячи по усіх можливих відкритих каналах, щоб зачепитися за один.

```
UDP
Source IP: 192.168.1.1
Destination IP: 192.168.1.2
Source Port: 1234
Destination Port: 80

UDP
Source IP: 192.168.1.2
Destination IP: 192.168.1.3
Source Port: 1234
Destination Port: 80

UDP
Source IP: 192.168.1.3
Destination IP: 192.168.1.4
Source Port: 1234
Destination Port: 80

...
```

Рисунок 1.5 – Приклад UPD-флуд DDoS-атаки

Приклад трафіка HTTP-запитів зображено на рисунку 1.6 які виконують надсилання запитав або запитів на авторизацію на веб-сайті. Ці запити містять неправильні дані, але вони все одно можуть перевантажити веб-сервер [13].

```
GET /index.html HTTP/1.1 Host: www.example.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.84 Safari/537.36 Accept: text/html,application/xhtml +xml,application/xml;q=0.9,image/avif,image/webp,image/apng,/;q=0.8,application/signed-exchange;v=b3;q=0.9 Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Connection: close

POST /login HTTP/1.1 Host: www.example.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.84 Safari/537.36 Accept: text/html,application/xhtml +xml,application/xml;q=0.9,image/avif,image/webp,image/apng,/;q=0.8,application/signed-exchange;v=b3;q=0.9 Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Connection: close
username=admin&password=password
```

Рисунок 1.6 – Приклад HTTP-флуд DDoS-атаки

1.2 Глибокі згорткові нейронні мережі (CNN) у виявленні шкідливих пакетів та DDoS-атак

Мережі з конволюційними нейронними мережами (CNN) навчаються за допомогою наборів даних, що включають як законні, так і шкідливі пакети. Ці дані використовуються для навчання мережі розпізнавати візуальні особливості, які відрізняють шкідливі пакети від законних. Після завершення навчання мережу можна використовувати для виявлення шкідливих пакетів у реальному часі. Мережа аналізує трафік і, користуючись знанням про відзнаки шкідливих пакетів, визначає, чи є певний пакет шкідливим [15, 16].

CNN мають численні переваги порівняно з іншими методами виявлення шкідливих пакетів та атак DDoS. Вони є надзвичайно точними, спроможними виявляти різноманітні шкідливі атаки і перевершують статистичні методи за цією характеристикою. Крім того, CNN є більш масштабованими, можуть впоратися з великим обсягом трафіку і використовуються для виявлення шкідливих пакетів у великому масштабі. Це робить їх ефективним і надійним інструментом в боротьбі з кіберзагрозами та забезпечує високий рівень безпеки мережі [16].

Глибокі згорткові нейронні мережі можуть автоматизувати процес виявлення аномальних мережевих активностей, що може бути особливо корисним у великих мережах, де ручне виявлення атак є складним завданням.

CNN вивчає складні зв’язки між об’єктами, ідентифікувати об’єкти чи об’єкти незалежно від їхнього положення та зменшувати обчислювальну складність мережі. Мережа складається з кількох шарів, включаючи згорткові шари, шари об’єднання та пов’язані шари, приклад зображено на рисунку 1.7.

CNN працює на локальних рецептивних полях і деяких варіантах RNN, такі як довготривала короткочасна пам’ять (LSTM), використовують спільний доступ до параметрів. Згорткові нейронні мережі в сучасних інформаційних технологіях також все частіше використовуються в поєднанні з іншими методами машинного навчання, що покращує можливість мережі тренуватися на наборах даних і дає можливість до динаміки у самостійному розвитку.

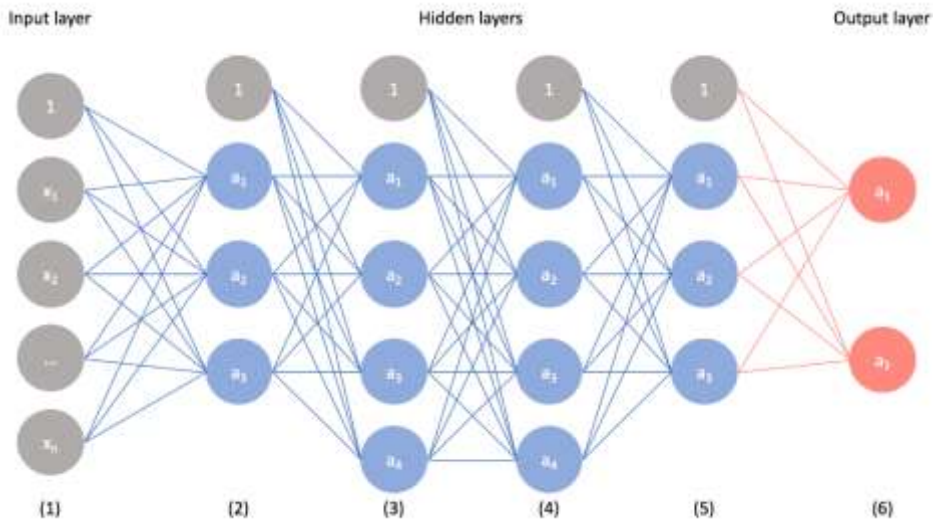


Рисунок 1.7 – CNN багатошаровий перцептрон

Машинне навчання (ML) — це галузь штучного інтелекту (AI), яка передбачає створення алгоритмів, які можуть навчатися на основі даних без додаткового програмування. Зрештою, системи ML здатні вчитися на даних і робити прогнози чи рішення без необхідності явного дотримання правил чи інструкцій [17, 18].

Аспекти машинного навчання, які будуть корисними для CNN і завдання виявлення DDoS-атак:

– класифікація використовується для класифікації даних у категорії. Наприклад, ML використовується в системах кредитного ризику та системі виявлення шахрайства;

– регресія використовується для прогнозування значення змінної. Наприклад, ML використовується в системах прогнозування погоди та системах рекомендацій;

– узагальнення використовується для узагальнення знань із набору даних на нові дані. Наприклад, ML використовується в системах машинного перекладу та системах виявлення захворювань [18, 37].

Методи виявлення DDoS, засновані на машинному навчанні (ML), можна класифікувати на три основні групи: контрольовані, неконтрольовані та гібридні, і

кожна з них включає кілька підкатегорій. Комплексна систематика методів виявлення DDoS на основі ML зображена на рисунку 1.8.

Архітектура методу включає три ключові компоненти - попередню обробку даних, вилучення ознак і класифікацію. Під час попередньої обробки даних видаляються зайві та нерелевантні функції мережевого трафіку. Модель CNN використовується для виділення просторових ознак з оброблених даних, тоді як модель LSTM відповідає за класифікацію часових ознак, отриманих із CNN. LSTM забезпечує захоплення послідовних залежностей у даних та ідентифікацію шаблонів DDoS-атак. Ефективність запропонованого методу була оцінена на реальному наборі даних мережевого трафіку, зібраному з програмно визначеного тестового стенда IIoT [33, 51].

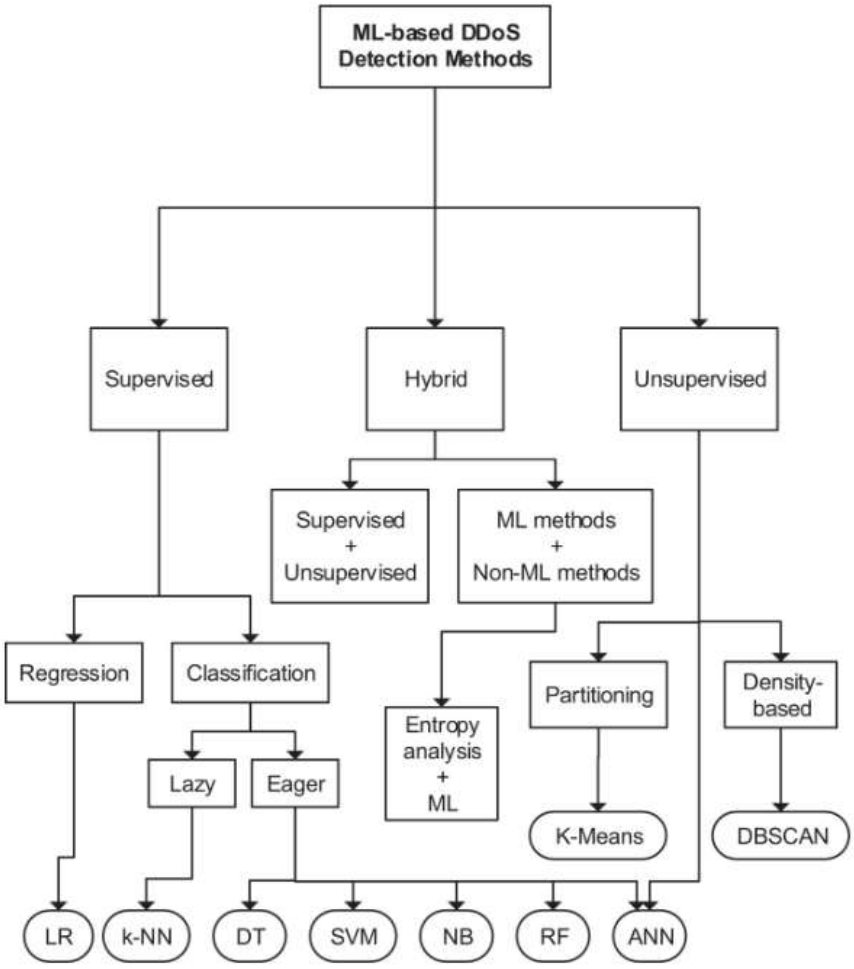


Рисунок 1.8 - Система розподілених методів виявлення відмови в обслуговуванні на основі машинного навчання

Використання машинного навчання як інструменту для виявлення аномалій та розрізнення доброякісного і атакуючого трафіку є актуальною темою досліджень, яка показує вражаючі результати. Один із підходів включає в себе використання фізичної мережі як спеціальної площадки для тестування, де присутні як атакуючі, так і цільові комп'ютери, і контрольовано виконуються численні атаки. Отримані дані про трафік можна використовувати для тренування алгоритмів машинного навчання під наглядом, щоб відрізняти атакуючий трафік від безпечного. Крім цього, алгоритми неконтрольованого навчання можна використовувати для реального часу кластеризації вхідного трафіку, розпізнавши звичайний трафік від атак на основі їхньої поведінки і функціональних особливостей. У обох підходах пакети або потоки трафіку представлені з використанням ключових характеристик, таких як розмір пакета, протокол і інтервал між пакетами.

1.3 Аналіз методів виявлення DDoS-атак на основі глибоких згорткових нейронних мереж

Розробка методів виявлення DDoS-атак на основі глибокого навчання пропонує кілька підходів. Використання глибокої нейронної мережі дозволяє розподілити мережевий трафік на дві категорії: DDoS та не-DDoS. Інший метод передбачає використання CNN для виявлення аномалій у трафіку, що можуть свідчити про можливу DDoS-атаку.

DDoS-Detector - це метод виявлення DDoS-атак на основі CNN, який використовує CNN для класифікації трафіку як DDoS або не-DDoS, на рисунку 1.10 наведені переваги та недоліки такого підходу і подібних до нього.

CNN відрізняються від традиційних методів виявлення DDoS-атак за їхньою вищою точністю та ефективністю. Це робить їх більш привабливими для застосування в області кібербезпеки. Крім того, CNN можуть навчатися впізнавати нові, раніше невідомі види DDoS-атак, розширюючи спектр захисту мережевих систем.

На етапі збору даних збираються зразки DDoS-трафіку та не-DDoS-трафіку. DDoS-трафік можна зібрати з різних джерел, таких як honeypots, sandboxes та аналіз трафікових потоків. Не-DDoS-трафік можна зібрати з реальних мереж [36, 50].

Метод DDoS-Detector складається з наступних етапів [36, 38]:

– збір даних, який містить зразки DDoS-трафіку та не-DDoS-трафіку.
– обробка даних, по специфічним параметрам;
– навчання на наборі даних;
– класифікація трафіку.

На етапі обробки даних дані обробляються таким чином, щоб їх можна було використовувати для навчання CNN та включає в себе наступні кроки, які зображені на рисунку 1.9.

На етапі навчання CNN навчається на наборі даних, що містить оброблені дані. CNN може бути навчена за допомогою різних алгоритмів навчання, таких як backpropagation.

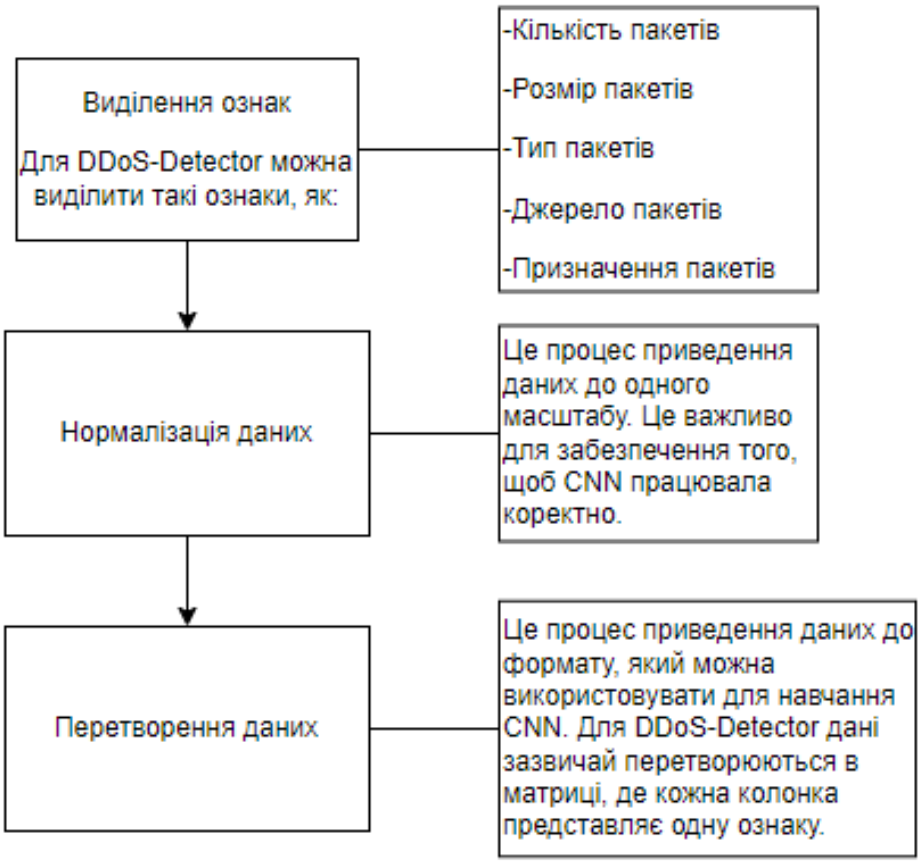


Рисунок 1.9 – Етапи обробки даних для CNN

На етапі класифікації трафіку CNN використовується для класифікації трафіку. CNN видає оцінку того, чи є трафік DDoS-атакою. Ця оцінка може бути використана для прийняття рішення про те, чи потрібно блокувати трафік. Метод DDoS-Detector був протестований на різних наборах даних і показав високу точність. Метод DDoS-Detector є одним із найбільш ефективних методів виявлення DDoS-атак на основі CNN [37].

CNN-DDoS - це метод виявлення DDoS-атак на основі CNN, який використовує CNN для виявлення аномалій у трафіку, які можуть бути ознаками DDoS-атаки.

Метод CNN-DDoS був протестований на різних наборах даних і показав високу точність. Метод CNN-DDoS є одним із найбільш ефективних методів виявлення DDoS-атак на основі CNN [36, 47].

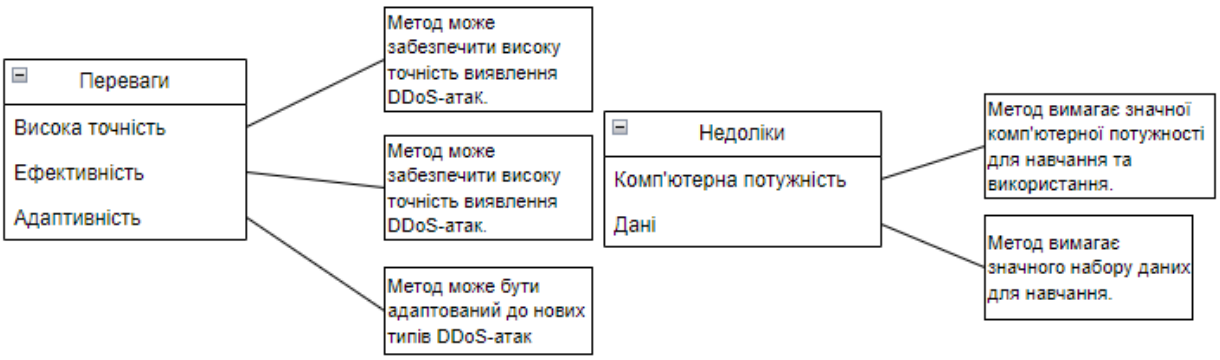


Рисунок 1.10 – Переваги і недоліки методів виявлення DDoS-атак

DeepDDoS - це система виявлення DDoS-атак, яка використовує глибокі нейронні мережі для аналізу мережевого трафіку та виявлення атак. Основні кроки використання DeepDDoS можуть виглядати приблизно так [37, 45]:

– зібрати дані про мережевий трафік;
– підготувати дані для подальшої обробки;
– розробити глибоку згорткову мережу;
– використовувати навчальний набір даних для тренування DeepDDoS;

– використовувати втрати та оптимізатор для налаштування параметрів мережі;

– використовувати тестовий набір даних для оцінки ефективності моделі;

– аналізувати показники точності, чутливості та специфічності системи виявлення DDoS-атак;

– оптимізувати модель для забезпечення швидкості та точності виявлення атак;

– якщо модель показує високу ефективність на тестовому наборі даних, вона може бути впроваджена в реальну систему захисту для нагляду за мережевим трафіком та виявлення DDoS-атак.

Це загальний підхід до використання DeepDDoS або будь-якої аналогічної системи для виявлення DDoS-атак.

1.4 Постановка завдання

Мета кваліфікаційної роботи магістра – полягає у вирішенні проблеми виявлення шкідливих пакетів та DDoS-атак у мережі шляхом аналізу мережевого трафіку з використанням глибоких згорткових мереж та методів машинного навчання. Для підготовки даних для тренування нейронної мережі буде використано метод з учителем, який включає в себе аналіз готових вибірок даних.

Проведено аналіз ефективності CNN у виявленні шкідливих пакетів. Глибокі згорткові нейронні мережі (CNN) є ефективним інструментом для виявлення шкідливих пакетів та DDoS-атак у комп'ютерних мережах. Мережу можна навчити розпізнавати особливості, що відрізняють шкідливі пакети від дозволених.

Використання поєднання машинного навчання з глибокими згортковими мережи є ефективним способом виявлення шкідливих пакетів комп'ютерних мережах.

Аналізуючи існуючі методи виявлення DDoS-атак, було встановлено, що найбільш ефективними - є моделі згорткової нейронної мережі (CNN) та довготривалої короткочасної пам'яті (LSTM).

Вибір наборів даних, використання відповідних алгоритмів навчання та комбінація методів можуть значно покращити ефективність виявлення DDoS-атак. Були розглянути бази DARPA та KDD.

Проведено аналіз методів виявлення DDoS-атак на основі глибоких згорткових нейронних мереж за результатами якого метод DDoS-Detector, який використовує поглиблені нейронні мережі для виявлення DDoS-атак у мережевому трафіку, є дуже ефективним та має численні переваги.

У першому розділі було виконано аналіз стану DDoS-атак і виявлено, що DDoS-атаки є серйозними загрозами для мережевої безпеки, спрямованими на нанесені шкоди системам.

У рамках цієї роботи буде розроблена глибока згорткова мережа, яка здатна аналізувати різні аспекти мережевого трафіку, виявляти патерни та виявляти аномалії. Для досягнення цієї мети, пов'язаної із виявленням шкідливих пакетів

через аналіз мережі за допомогою глибоких згорткових мереж (CNN), необхідно виконати наступні *задачі дослідження*:

1. Розробка оптимальних алгоритмів аналізу – виконати дослідження оптимальних алгоритмів для аналізу мережевого трафіку, що дозволяють точно ідентифікувати шкідливі пакети та DDoS-атаки.

2. Методи підготовки даних – розробити ефективний метод підготовки даних для навчання, які враховують різноманітні аспекти мережевого трафіку та забезпечують стабільні результати під час тренування.

3. Архітектура глибокої згорткової мережі - розробити оптимальну архітектуру глибокої згорткової мережі, яка може адекватно аналізувати та класифікувати різні типи мережевого трафіку, забезпечуючи надійні результати виявлення шкідливих пакетів.

Ці аспекти дослідження спрямовані на розробку ефективної та надійної системи виявлення шкідливих пакетів та DDoS-атак, що базується на глибокому аналізі мережевого трафіку та використанні передових методів машинного навчання.

2 ЕТАПИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

2.1 Опис методу аналізу мережевого трафіка

Попередня обробка має вирішальне значення для використання CNN під час аналізу даних мережевого трафіку. Дані зазвичай відображаються у вигляді пакетів, які містять ряд заголовків і корисних деталей. Щоб відформатувати його відповідно до мережі, необхідна попередня обробка.

Етапи обробки пакетів відбуваються наступним чином - пакети отримуються та сортуються, потім скануються та реєструються в центральному місці. Звідти пакети направляються до відповідного місця призначення та маркуються відповідним чином. Після того, як вони досягли місця призначення, пакети вивантажуються, сортуються та знову скануються для забезпечення точності. Нарешті пакети доставляються адресатам.

Щоб забезпечити певний рівень узгодженості, важливо стандартизувати як формат, так і розмір пакетів, що надсилаються.

Такі методи, як стиснення трафіку або одноразове кодування, можуть бути використані для перетворення даних у числовий формат, доцільний для обробки CNN.

Довжина пакета, IP-адреси джерела та призначення та номери портів є важливими характеристиками, які можна отримати з пакетних даних.

Помноження даних на скаляр таким чином, щоб середнє значення було близьким до 0, а стандартне відхилення було близьким до 1, ось що означає нормалізація, і це часто використовуваний метод попередньої обробки.

Іншою поширеною технікою попередньої обробки є вирівнювання. Вирівнювання полягає в тому, що дані розміщуються в одному масштабі. Це можна зробити, наприклад, шляхом масштабування всіх даних до діапазону.

Запропонована архітектура CNN складається з кількох згорткових рівнів, за якими слідують повністю зв’язані рівні. Згорткові шари витягують із вхідних даних характеристики високого рівня, тоді як повністю пов’язані рівні поєднують ці функції, щоб зробити прогнози щодо типу трафіку.

Навчання моделі CNN передбачає передачу їй великої кількості позначених даних трафіку та налаштування її параметрів, щоб мінімізувати помилку класифікації. Гіперпараметрична оптимізація передбачає вибір оптимальних значень таких параметрів, як кількість згорткових шарів, кількість фільтрів у кожному шарі та функції активації.

Архітектура згорткової нейронної мережі для аналізу мережевого трафіку представлена у вигляді діаграми на рисунку 2.1.

Першим кроком є обробка вхідних даних – серії пакетів, які містять різні заголовки та корисну інформацію. Нормалізація пакетів грає ключову роль у процесі забезпечення консистентності та однорідності даних. Подальший аналіз можна легко спростити шляхом перетворення пакетів у стандартний формат і зведення їх розмірів.

Для отримання оптимальних результатів аналізу мережевого трафіку важливо переконатися, що пакети адаптовані для глибокої згорткової обробки нейронної мережі. Звернення уваги на їхній розмір і форматування дозволить максимально використати можливості мережі для отримання точних і оперативних результатів.

Перетворення пакетних даних у числовий формат, який можуть ефективно обробляти згорткові нейронні мережі (CNN), є вирішальним етапом обробки. Щоб зробити його більш придатним для подальшого аналізу та використання в моделях машинного навчання, потрібно одноразове кодування або стиснення трафіку.

Для досягнення цього метафоричної "мови" мережі необхідно виконати вилучення ключових функцій з пакетних даних. Серед них можуть бути IP-адреси джерела та призначення, номери портів, довжина пакетів та інші параметри, які мають високу інформативність та визначальне значення для подальшого аналізу. Цей етап підготовки даних дозволяє значно скоротити обсяг даних, зберігаючи при

цьому важливу інформацію для подальшого аналізу та виявлення аномалій у мережевому трафіку.

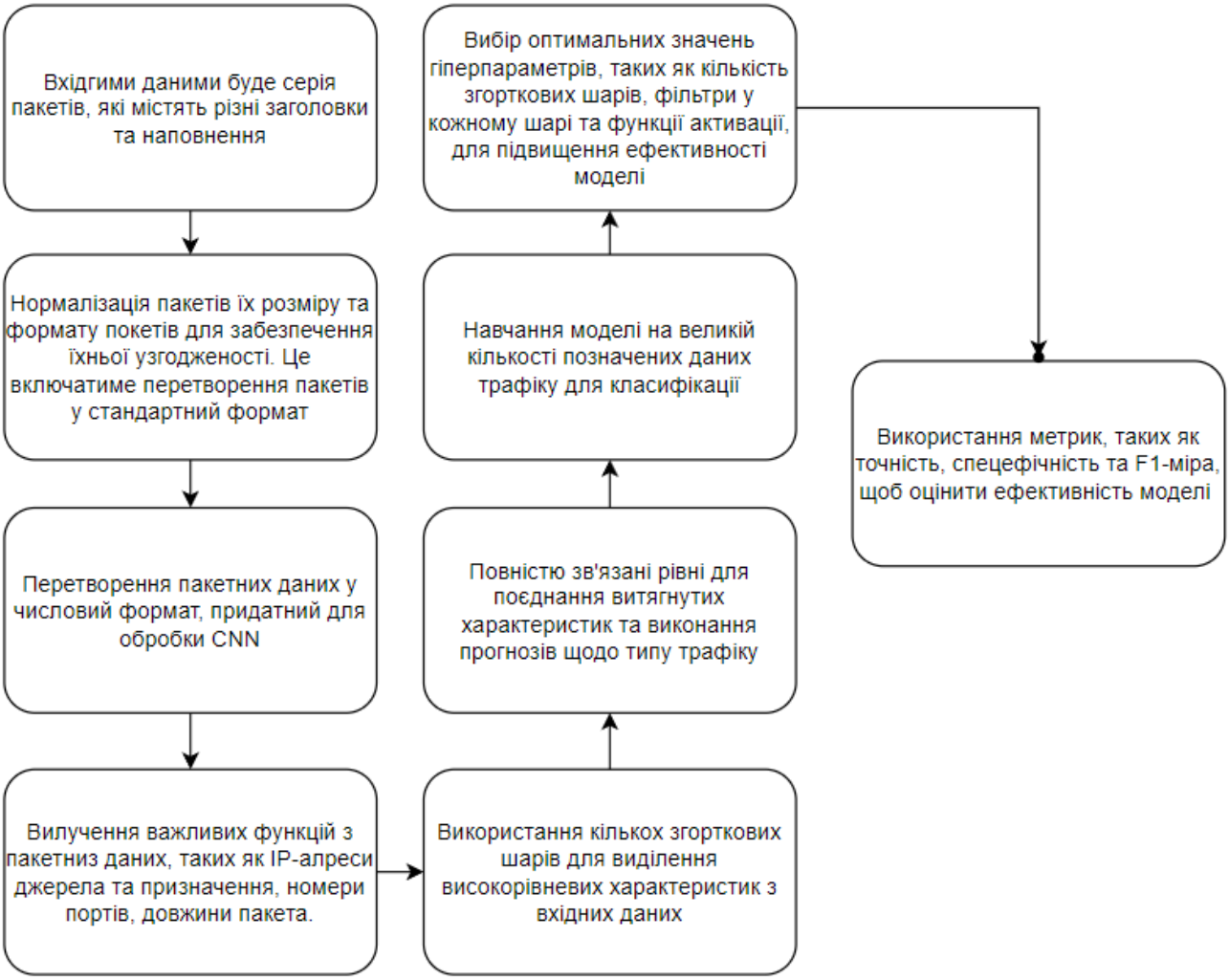


Рисунок 2.1 – Архітектура згорткової мережі

Використання кількох згорткових шарів у нейронних мережах є ключовим для ефективного виділення високорівневих характеристик з вхідних даних. Ці шари відповідають за виявлення абстрактних особливостей та взаємозв'язків у пакетах мережевого трафіку. Вони дозволяють автоматично витягувати та узагальнювати важливі ознаки, що допомагає зменшити розмірність даних та підготувати їх для подальшої обробки.

Наступним етапом є використання повністю зв’язаних рівнів для об’єднання витягнутих характеристик та роботи з ними для прогнозування типу трафіку. Ці

рівні дозволяють створювати зв'язки між отриманими характеристиками та визначати тип трафіку з великою точністю. Вони використовують узагальнені ознаки, зібрані з попередніх етапів обробки, для зроблення прогнозів та класифікації мережевого трафіку на основі виявлених особливостей.

Навчання моделі на великій кількості позначених даних мережевого трафіку є ключовим етапом для досягнення високої точності класифікації. Чим більше різноманітних даних має модель на вході, тим краще вона може вивчати та розрізняти різні типи трафіку, що покращує її здатність виявляти аномальну активність та шкідливі пакети.

Оптимізація нейронної мережі включає вибір оптимальних значень гіперпараметрів. Це важливий крок, який може суттєво вплинути на ефективність моделі. Налаштування кількості згорткових шарів, фільтрів у кожному шарі та функцій активації дозволяє досягти оптимального балансу між точністю прогнозування та обчислювальною складністю. Цей процес вимагає систематичних експериментів та оцінки результатів для вибору найкращих параметрів, що допомагає підвищити ефективність та точність моделі.

Використання метрик для оцінки моделі на тестових даних є важливим етапом у визначенні точності та ефективності алгоритму класифікації мережевого трафіку. Метрики, такі як точність (accuracy), відгук (recall), специфічність (specificity) та F1-мера, надають різноманітний огляд того, наскільки добре модель справляється з класифікацією.

2.2 Формування нових моделей для нейронних мереж

KDD Cup 99 Dataset є одним з найвідоміших та широко використовуваних наборів даних у галузі кібербезпеки. Він був створений для конкурсу KDD Cup 1999, спрямованого на виявлення вторгнень у комп'ютерні мережі. Цей набір даних став стандартом у вивченні та розробці методів виявлення атак у мережах.

KDD є цінним ресурсом для вивчення та розвитку систем виявлення атак у комп'ютерних мережах. Для вашої роботи, пов'язаної з методами виявлення DDoS-

атак та аналізу мережевого трафіку, ви можете використати цей набір даних для навчання моделей машинного навчання та тестування алгоритмів.

Основні характеристики KDD:

– набір даних складається з близько 4,9 мільйонів записів про мережевий трафік, включаючи нормальний трафік та різні типи атак;

– включає понад 40 різновидів атак, таких як Denial of Service (DoS), Probe (розвідувальні атаки), Remote to Local (R2L), User to Root (U2R) тощо;

– містить інформацію про різні протоколи мережі, типи пакетів та їхні характеристики, що можуть бути корисними для класифікації трафіку та виявлення аномалій;

– даний набір даних симулює реальний мережевий трафік з включеними атаками, тому може містити шум та особливості, які важко виявити;

– цей набір даних широко використовується для досліджень та тестування різних алгоритмів та моделей машинного навчання для виявлення вторгнень.

Ця база даних містить велику кількість атак та нормального мережевого трафіку і може бути корисною для наступних аспектів:

– включає широкий спектр атак для створення моделей, які можуть виявляти навіть найскладніші загрози;

– ці дані дозволяють моделям вивчати характеристики різних типів трафіку та атак для кращого розрізнення між ними.

Навчання з учителем – це метод машинного навчання, де модель отримує дані, позначені мітками або класами (у нашому випадку, атаки та нормальний трафік), і навчається робити прогнози чи класифікацію на основі цих міток [45].

Використання міток для атак та нормального трафіку дозволяє розділити дані на дві категорії: атаки та безпечний трафік.

Використання міток атак та нормального трафіку для розподілення даних на класи атак та безпечного трафіку.

Використання різних алгоритмів класифікації, таких як логістична регресія, дерева рішень, або глибокі нейронні мережі, через фреймворк ML.NET у поєднанні з мовою програмування C#, дозволяє створювати моделі, які можуть ефективно розпізнавати та класифікувати трафік як нормальний чи атаку.

Використання метрик, таких як точність, чутливість, специфічність, для оцінки ефективності моделей на основі їхньої здатності правильно класифікувати трафік [41].

Навчання без учителя використовує методи кластеризації для виявлення аномальних паттернів:

– використання методів кластеризації, наприклад, K-means або DBSCAN, для групування подібних об'єктів без наявності міток атак;

– аналіз отриманих кластерів для виявлення аномальних груп, які можуть вказувати на потенційні атаки чи нетипові паттерни у мережевому трафіку;

– оскільки дані в KDD Cup 99 Dataset є реальними або симульованими, структура даних може містити певні загальні ознаки, які можна використати для виявлення аномальних паттернів без прив'язки до певних атак.

KDD включає розглядання ключових характеристик мережевих пакетів:

– визначення небезпечних змін у використанні IP-адрес або портів, що може вказувати на атаку;

– аналіз типів пакетів та їхньої незвичайної частоти для виявлення аномалій;

– врахування незвичайних обсягів передачі даних або несподіваних часових шаблонів.

Отже, навчання без учителя у контексті виявлення аномальних паттернів використовує методи кластеризації, такі як K-means чи DBSCAN, для групування схожих об'єктів без міток атак. Це дозволяє аналізувати отримані кластери, виявляючи аномальні групи, які можуть вказувати на потенційні атаки чи нетипові паттерни у мережевому трафіку.

У KDD Cup 99 Dataset, що містить реальні або симульовані дані, ключові характеристики мережевих пакетів включають визначення небезпечних змін у

використанні IP-адрес та портів, аналіз типів пакетів та їхньої незвичайної частоти для виявлення аномалій, а також врахування несподіваних обсягів передачі даних чи часових шаблонів. Ці характеристики допомагають виявляти та аналізувати потенційно небезпечні або нетипові паттерни, які можуть вказувати на атаки у мережі.

2.3 Оцінка та порівняння моделей

Порівняння моделей проводиться з використанням метрик та стратегій оцінки, які дозволяють визначити, яка модель краще справляється з завданням виявлення DDoS-атак у вашому конкретному випадку.

Точність (Precision) - ідентифікована моделлю загальна кількість позитивних випадків, точність якої дорівнює відношенню кількості правильно класифікованих позитивних випадків [39].

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

TP (True Positives) - виявлені позитивні випадки вважаються справжніми позитивними (TP), що є точною кількістю виявлених атак.

FP (False Positives) - помилково ідентифіковані позитивні випадки, також відомі як помилкові спрацьовування (FP), виникають, коли звичайний трафік помилково позначається як атака.

Втрати (Loss) є ключовою метрикою в оцінці моделей машинного навчання. Вони відображають, наскільки точно модель передбачає реальні значення в порівнянні з фактичними даними.

У контексті нейронних мереж, втрати є числовим значенням, яке визначає, наскільки передбачені значення відрізняються від фактичних. Це вимірюється за допомогою функції втрати (loss function), яка обчислює різницю між передбаченими значеннями і правильними значеннями.

Коли модель навчається, вона спочатку має високі втрати, оскільки її прогнози можуть бути далекими від реальності. Поступово, під час тренування, модель намагається зменшити ці втрати, оновлюючи внутрішні параметри так, щоб прогнози були ближчими до правильних значень.

Зменшення втрат під час тренування вказує на те, що модель стає кращою у передбаченні. Чим менші втрати, тим більше модель відповідає даним тренувального набору.

ROC (Receiver Operating Characteristic): Це графік, що використовується для оцінки якості бінарної класифікації. Він показує відношення між чутливістю (true positive rate) і специфічністю (false positive rate) для різних значень порогу відсічення [40].

Чим більше AUC, тим краще модель вирішує проблему класифікації, оскільки вона вимірює загальну якість класифікації моделі, незалежно від конкретного порогового значення. По суті, AUC означає площу під кривою та обчислюється шляхом визначення площі під кривою ROC.

Процес побудови ROC-AUC:

– TPR - це чутливість моделі, визначена як відношення правильно класифікованих позитивних екземплярів (True Positives) до загальної кількості позитивних екземплярів (True Positives + False Negatives). TPR показує, яку частку позитивних екземплярів модель здатна відслідковувати правильно.

– FPR - це специфічність моделі, визначена як відношення помилково класифікованих негативних екземплярів (False Positives) до загальної кількості негативних екземплярів (False Positives + True Negatives). FPR вказує на частку негативних екземплярів, які помилково класифіковані як позитивні.

ROC крива візуалізує відношення між TPR і FPR для різних значень порогу відсічення. Чим більше відхилення від діагональної лінії (що відповідає випадковому класифікатору), тим краще працює модель [41, 42].

AUC обчислює площу під цією кривою. Якщо модель є ідеальною, AUC буде дорівнювати 1.0. Якщо модель не краща за випадковий класифікатор, AUC буде близько до 0.5.

Отже, ROC-AUC є метрикою, яка дозволяє оцінити якість бінарної класифікації моделі, і чим більше площа під кривою ROC, тим краще модель робить прогнози.

Матриця плутанини (Confusion Matrix) - це таблиця, яка використовується для оцінки ефективності моделі класифікації, особливо в задачах бінарної або багатокласової класифікації. Вона відображає кількість правильних та неправильних класифікацій для кожного класу у прогнозах моделі.

Інформація в матриці плутанини дозволяє отримати різні метрики:

– відношення правильно класифікованих позитивних прикладів до всіх прикладів, які модель визнала як позитивні;

– відношення правильно класифікованих позитивних прикладів до всіх фактичних позитивних прикладів;

– відношення правильно класифікованих негативних прикладів до всіх фактичних негативних прикладів;

Ця матриця допомагає оцінити ефективність моделі у виявленні DDoS-атак. За її допомогою можна обчислити різні метрики, такі як точність, чутливість, специфічність тощо, що вказують на здатність моделі правильно визначати DDoS-атаки та нормальний мережевий трафік [50].

У контексті виявлення DDoS-атак:

– кількість правильно виявлених DDoS-атак;

– кількість правильно виявлених нормальних (не-DDoS) пакетів;

– кількість неправильно класифікованих як DDoS пакетів;

– кількість неправильно класифікованих як не-DDoS (коли насправді це DDoS) пакетів.

F1-міра - це метрика, яка оцінює точність моделі класифікації, особливо в контексті незбалансованих класів. Вона представляє собою гармонічне середнє між двома іншими метриками: точністю (precision) та чутливістю (recall).

Чутливість (Recall/Sensitivity) - це відношення кількості правильно визначених позитивних випадків до загальної кількості існуючих позитивних випадків.

$$Recall = \frac{TP}{TP + FP} \tag{2.2}$$

FN (False Negatives) - кількість неправильно не визначених позитивних випадків (атаки, які були помилково визнані як нормальний трафік).

F1-міра об'єднує точність та чутливість у єдину метрику, що є гармонічним середнім між ними.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.3}$$

F1-міра враховує обидва аспекти класифікації, дозволяючи оцінити модель не тільки з точки зору правильно класифікованих позитивних випадків, а й здатності уникнути пропусків атак (чутливість). Вона дозволяє зрозуміти, наскільки модель здатна досягати балансу між цими двома аспектами у своїй роботі [43].

Специфічність (Specificity) - це метрика, яка визначає здатність моделі правильно ідентифікувати від'ємні випадки, тобто правильно класифікувати дані, які належать до негативного класу (наприклад, нормальний трафік в мережі) відносно загальної кількості даних негативного класу або їх спецефічності на кількість правильних значень.

Специфічність є важливою метрикою, особливо коли маємо справу з незбалансованими даними, де кількість від'ємних випадків (нормальний трафік) може бути значно вищою за кількість позитивних випадків (атаки). Вона допомагає оцінити, наскільки модель ефективно відрізняє нормальний трафік від атак,

забезпечуючи додатковий контроль над прогнозами моделі у відношенні до від'ємних випадків [38, 46].

Формула специфічності:

$$Specificity = \frac{TP}{TP + FP} \quad (2.4)$$

TN (True Negatives) - кількість правильно визначених від'ємних випадків (нормальний трафік, який був правильно визнаний як нормальний).

FP (False Positives) - кількість неправильно визначених позитивних випадків (атака, яка була помилково визнана як нормальний трафік).

Специфічність визначає:

– специфічність дозволяє оцінити, наскільки добре модель відрізняє від'ємні випадки від позитивних. Чим вище значення специфічності, тим менше помилкових позитивних класифікацій у моделі;

– вона важлива у випадках, коли нам потрібно впевнитися, що модель правильно визначає нормальний трафік, щоб уникнути включення в аналіз від'ємних випадків у позитивний клас (наприклад, помилкове класифікування нормального трафіку як атаки).

Комбінування F1-міри та Специфічності дозволяє зберегти баланс між здатністю моделі правильно класифікувати атаки та її здатністю виявляти нормальний трафік. Це допомагає отримати більш повний образ про ефективність моделі у виявленні DDoS-атак без врахування тільки одного аспекту.

Під час навчання одним із важливих параметрів для нейронних мереж є швидкість навчання, яка визначає швидкість, з якою модель змінює свої ваги. Він відіграє важливу роль у збіжності моделі до оптимального рішення або локального мінімуму функції втрат, і його вплив може або прискорити, або перешкодити процесу [51].

Це впливає на процес навчання через оновлення ваг моделі за допомогою градієнтів функції втрати. У звичайному градієнтному спуску оновлення ваг визначається так:

$$нова\ вага = стара\ вага - швидкість\ навчання \times градієнт \quad (2.5)$$

Основна ідея полягає у тому, щоб знаходити оптимальне значення швидкості навчання, яке дозволяло моделі ефективно збігатися до оптимального рішення без перенавчання чи ускладнення навчання.

Вибір швидкості навчання включає:

– збереження константної швидкості навчання;

– зниження швидкості навчання з часом (Learning Rate Decay);

– адаптивні методи швидкості навчання.

Крива навчання (Learning Curve) - це графік, який демонструє зв'язок між кількістю навчальних прикладів і метрикою ефективності моделі (наприклад, точність або втрати) на навчальному і валідаційному наборах даних. Ця крива дає можливість оцінити, як точність моделі змінюється в залежності від обсягу навчальних даних [52].

Основні риси кривої навчання:

– ось X (вісь абсцис), кількість навчальних прикладів;

– ось Y (вісь ординат), показник ефективності моделі.

Кількість прикладів або обсяг навчальних даних, які використовуються для тренування моделі. Це може бути кількість зразків, партій, епох тренування тощо.

Метрика, яка відображається на графіку (наприклад, точність, втрати, F1-міра). Вона показує, наскільки добре модель вирішує поставлену задачу в залежності від кількості навчальних прикладів і по цій метриці можна оцінити якість натренованості і правильність оформлення мережі.

Аналіз кривої навчання може допомогти в прийнятті рішень про подальші кроки у тренуванні моделі.

Недостатня навченість, якщо як на навчальній, так і на валідаційній кривих точність занадто низька, це може свідчити про те, що модель недостатньо складна або потребує більше епох тренування.

Перенавченість, якщо на навчальній кривій точність висока, але на валідаційній низька, можливо, модель перенавчилася на навчальних даних і потребує більше даних або регуляризації.

Оптимальний момент, коли криві для навчального та валідаційного наборів даних збігаються або залишаються стабільними. Це може бути момент, коли модель навчилася.

Здатність моделі узагальнювати нові дані та потребу в повторному навчанні можна оцінити, спостерігаючи за двома кривими на кривій навчання, які графічно представляють зміну в показниках ефективності, оскільки дані, які використовуються для навчання моделі, збільшуються як для наборів даних для навчання, так і для перевірки.

2.4 Аналіз результатів та перевірка ефективності

Для здійснення всебічного порівняння існуючих методів виявлення було проаналізовано найкращі пропозиції ML, використовуючи їхні показники ефективності, які представлені у таблиці 2.1. Крім того, було розглянуто недоліки кожного методу, які відображені у таблиці 2.2. Загалом, аналіз вказує на те, що серед моделей ML найбільш ефективними у виявленні атак є випадковий ліс (RF) та машина опорних векторів (SVM), а серед методів глибокого навчання - моделі згорткової нейронної мережі (CNN) та довготривалої короткочасної пам'яті (LSTM). Варто відзначити, що інші методи також продемонстрували перспективні результати у конкретних випадках [20, 21].

Але не варто забувати, що для нейронних мереж, особливо, якщо використовуються методи машинного навчання, велика значення має формування набору даних. Якщо потрібно виконувати збір на основі сигнатур, то особливість полягає у порівнянні вхідних. Цей метод може виявляти тільки відомі атаки.

Навпаки, система виявлення аномалій виявляє будь-яку поведінку, яка є не типовою для нормальної роботи системи [23].

Щоб отримати набір даних DDoS-атак потрібно застосовувати спеціалізовані програмні засоби, що використовують різні методи, або використовувати вже зібрані дані, які включають як реальний, так і згенерований трафік. Ці дані можуть служити для створення нових систем виявлення вторгнень, які можуть передбачати різні типи DDoS-атак [19,23].

Таблиця 2.1 - Здатність методів на основі машинного навчання (ML) виявляти атаки розподіленої відмови в обслуговуванні (DDoS).

Метод	**Acc.**	**Rec.**	**Prec.**	**F1-score**	**FPR**	**AUC**
CNN+LSTM [21]	0.97	0.99	0.97	0.97	N/A	N/A
CNN+LSTM [20]	0.99	0.99	0.99	0.99	N/A	0.99625

Таблиця 2.2 – Неділоки методів у виявленні DDoS-атак.

Метод	**Недоліки**
CNN+LSTM [21]	Не оцінено на основі специфічного набору даних IoT, не охоплено сучасні типи атак
CNN+LSTM [20]	Не оцінено на основі спеціального набору даних IoT, пропущено багато типів атак і зразків

Виділяють чотири підходи перевірки запропонованого методу виявлення [19, 21]:

– математичні моделі, що символічно описують систему і перевіряються математично;

– моделювання або відтворення структури для експериментів на одній системі, дозволяючи гнучкість і відкидання непотрібних альтернатив;

– умуляція моделювання з реальними системами, об'єднуючи реальні та імітаційні елементи, працюючи в реальному часі, але з обмеженою масштабованістю;

– використання реальних умов мережі, операційних систем, додатків та платформ для експериментів, але з обмеженнями, такими як неможливість зміни топології мережі та вразливість до атак.

CNN мають два основних види навчання [27]:

– навчання на залежному від вчителя методі використовує набір даних, позначений цільовими значеннями;

– навчання без нагляду використовує набір даних, не позначений цільовими значеннями. Мережа уміє виділяти істотні характеристики з даних.

Контрольоване навчання в контексті виявлення атак DDoS передбачає використання алгоритму для вивчення функції:

$$f\ (x) = y \qquad (2.6)$$

Отже, вибір тренувальної бази з атаками - завдання не однозначне. Широко використовувані бази містять застарілі атаки, тоді як новіші бази мають складну структуру та вимагають уважної обробки, і використовуються обмеженим колом дослідників, що ускладнює порівняння результатів. Наразі виділяються дві основні бази з відомими атаками - DARPA [24] і KDD [25]. Опис цих баз наведено в таблиці 1.4.

Одним із найпоширеніших алгоритмів класифікації для виявлення DDoS є дерево рішень, яке є ієрархічною моделлю контрольованого навчання. На рисунку 2.2 представлено схематичну ілюстрацію роботи алгоритму DT, який можна використовувати для виявлення DDoS. У цій моделі кожен вузол є вузлом прийняття рішення, і він реалізує тестову функцію [28,29]:

$$f_m\ (x) \qquad (2.7)$$

З дискретними виходами, які генерують окремі гілки в дереві. Алгоритм DT вимагає нещодавно отриманих вхідних даних (пакет або потік) для проходження вузлів перед тим, як зрештою йому буде призначено мітку в аркуші [30].

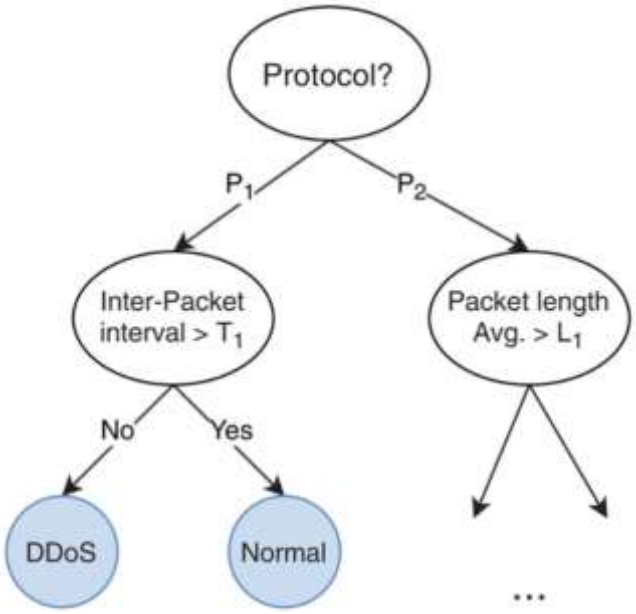


Рисунок 2.2 - Зразок функціональності дерева рішень (DT) для розподіленого виявлення відмови в обслуговуванні (DDoS)

Функція відображає вхідну змінну *x* до вихідної змінної *y*. Вивчення функції відображення досягається на основі набору даних, який містить значну інформацію про мережевий трафік, що дає змогу розробляти прогнозні моделі, які можуть відрізняти звичайний трафік від зловмисного. По суті, модель навчається з використанням набору даних і при поданні нових вхідних даних *x*, він обчислює відповідне вихідне значення *y* [28].

Ледачий класифікатор, у відміну від нетерплячого, відкладає процес побудови моделі та прогнозування до того моменту, коли надходять нові дані. Цей підхід усуває необхідність будувати модель до отримання нової інформації, зменшуючи час, потрібний для її створення. Проте, час для здійснення прогнозів може бути значно великим [29].

k-NN класифікатор обчислює відстань між новим отриманим зразком (наприклад, потоком або пакетом) і зразками в наборі даних. Евклідова відстань,

яка зазвичай використовується для обчислення відстані двох зразків, визначається наступним чином [25]:

$$distance\ (x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{2.8}$$

де x_1, x_2 є значеннями i функції для зразків x_1, x_2, відповідно, і n це кількість ознак.

Потім, k вибираються зразки, найближчі до отриманого зразка, і обчислюється таке рівняння [26]:

$$p(C_i, x) = \frac{k_i}{k} \tag{2.9}$$

де C_i представляє розглянуті мітки класу (наприклад, [DDoS, нормальний]), $p(C_i, x)$ це ймовірність того, що вибірка x належить до класу C_i і k_i це кількість зразків у k-найближчих зразках, які є членами C_i.

KDDCup99 і CICIDS2017 були двома поширеними наборами даних, у дослідженні розділили функції на підмножини, щоб передавати кожну з них на канали CNN [29,30].

Дослідження у якому було виконано порівняльний аналіз різних моделей машинного та глибокого навчання для виявлення DDoS-атак у мережах IoT. Алгоритми машинного навчання включали SVM, RF і NB, тоді як глибокі моделі навчання складалися з MLP, LSTM, CNN і комбінації CNN і LSTM. Використовувався набір даних CICIDS2017 для оцінки результатів. Отримані результати показали, що модель CNN+LSTM виявилася найефективнішою з точки зору виявлення DDoS-атак у мережах IoT [34].

Підсумок останніх досліджень щодо виявлення DDoS-атак за допомогою згаданих вище контрольованих алгоритмів машинного навчання наведено у

таблиці 2.3 [29]. Ці підсумки показують ефективність деяких видів алгоритмів показуючи їх точність і спроможність до тренованості, що також суттєво впливає на якість роботи мережі.

Таблиця 2.3 - Час навчання та тестування алгоритмів.

Алгоритм	**Час навчання (с)**	**Час тестування (с)**	**Середній**
DT	17.43	3.03	10.23
RF	171.11	5.19	88.15
SVM	168,59	1,97	85,28
k-NN	0,13	15957,7	7978,915

Отже, дослідження підтверджує, що комбінація CNN і LSTM виявляється найефективнішою для виявлення DDoS-атак у мережах IoT, дозволяючи досягти високої точності та рівня виявлення у порівнянні з іншими розглянутими моделями.

У дослідженні був розроблений метод виявлення та класифікації DDoS-атак у програмно-визначених мережах промислового Інтернету речей (IIoT), який поєднує у собі дві потужні моделі глибокого навчання - CNN та LSTM.

2.5 Висновок до розділу

Зважаючи на значущість попередньої обробки даних для ефективного використання згорткових нейронних мереж у аналізі мережевого трафіку, виявляється, що ці етапи грають ключову роль. Процес включає отримання, сортування, сканування та реєстрацію пакетів, нормалізацію формату та розміру для забезпечення їхньої узгодженості, а також використання методів, які дозволяють перетворити дані у відповідний числовий формат.

Архітектура згорткової нейронної мережі використовує кілька згорткових та повністю зв’язаних рівнів для виділення важливих характеристик та зроблення прогнозів щодо типу трафіку. Підготовка даних перед навчанням моделі є

вирішальним етапом для досягнення високої точності класифікації, а оптимізація гіперпараметрів впливає на ефективність моделі.

Остаточна оцінка моделі на тестових даних засвідчить, наскільки ефективно вона класифікує мережевий трафік, використовуючи метрики, такі як точність, відгук, специфічність та F1-мера, для оцінки рівня точності та ефективності алгоритму.

Основний набір даних KDD Cup 99 став класичним інструментом для аналізу кібербезпеки, пропонуючи понад 40 типів атак та нормальний трафік для досліджень. Він надає можливість створення моделей, здатних виявляти складні загрози, та дозволяє аналізувати різноманітні аспекти мережевого трафіку.

Навчання з учителем використовує мітки для розділення даних на категорії атак та безпечного трафіку, розвиваючи моделі класифікації з використанням різних алгоритмів. Це дозволяє ефективно розпізнавати та класифікувати трафік, використовуючи метрики для оцінки моделей.

Навчання без учителя, використовуючи KDD Cup 99 Dataset, використовує методи кластеризації для виявлення аномальних паттернів у мережевому трафіку. Аналізуючи ці паттерни, можна виявити потенційні атаки чи нетипові взірці без прив'язки до конкретних видів атак.

Оцінка моделей для виявлення DDoS-атак використовує різноманітні метрики та стратегії, щоб визначити, яка найкраще справляється з цією задачею. Точність вимірює, наскільки модель правильно класифікує позитивні випадки, враховуючи TP та FP. Втрати, в свою чергу, показують, наскільки точно модель передбачає значення порівняно з реальними даними. Ці метрики, зокрема кількість правильно та неправильно класифікованих пакетів DDoS та нормального трафіку, надають можливість оцінити точність та ефективність моделі у виявленні DDoS-атак.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДУ ЧЕРЕЗ МАШИНЕ НАВЧАННЯ ТА ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ ДЛЯ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ

3.1 Розробка архітектури системи

Архітектура системи для виявлення шкідливих пакетів та DDoS атак на основі аналізу мережевого трафіку буде побудована на платформі C# та WinForms. Вона базуватиметься на глибоких згорткових нейронних мережах для аналізу мережевих даних. Діаграма потоку даних системи виявлення DDoS-атак буде мати наступний вигляд, представлений на рисунку 3.1.

Рисунок 3.1 – Діаграма потоку даних системи для виявлення шкідливих пакетів.

Ця діаграма показує, як дані про мережевий трафік надходять на вхід системи виявлення атак. Ці дані потім аналізуються системою, щоб виявити ознаки атак.

Якщо система виявляє атаку, то вона генерує результати виявлення атак, які виводяться на вихід.

Вхідні дані можуть включати як фізичний, так і логічний мережевий трафік. Фізичний мережевий трафік - це дані, які передаються по фізичних мережевих каналах, таких як Ethernet або Wi-Fi. Логічний мережевий трафік - це дані, які передаються по логічних мережевих каналах, таких як IP-пакети. Протокольний мережевий трафік - це дані, які передаються по конкретних мережевих протоколах, таких як TCP, UDP або HTTP. Атрибути мережевого трафіку - це характеристики мережевого трафіку, такі як його джерело, призначення, порти та протоколи. Метадані мережевого трафіку - це додаткові дані про мережевий трафік, такі як час, дата, тривалість та інші.

Мережевий трафік збирається з різних джерел, таких як фізичні мережеві пристрої, логічні мережеві пристрої та інші системи безпеки. Після збору мережевий трафік обробляється для видалення шуму та інших непотрібних даних. Потім мережевий трафік фільтрується для видалення незначних атак. Потім мережевий трафік аналізується системою виявлення атак, щоб виявити ознаки атак.

Якщо система виявлення атак виявляє атаку, то вона генерує результати виявлення атак. Ці результати включають тип атаки, деталі атаки та рекомендації щодо реагування. Результати виявлення атак зберігаються для подальшого аналізу та аудиту.

Для інтерактивної візуалізації результатів аналізу мережевого трафіку вибір C# і WinForms був би комплексним і ефективним підходом. Широкий спектр можливостей, які пропонують ці опції, робить їх швидкими та універсальними для створення настільних програм.

У налаштуваннях .NET ML.NET здатний створювати, навчати та запускати моделі машинного навчання, що робить його потужною технологією машинного навчання. За допомогою цього інструменту я можу зручно поєднувати свої будівельні блоки з іншими розділами системи.

Швидше створення прототипів і швидкі зміни програми можливі за допомогою однієї мови програмування для проекту, а розробка програмного забезпечення, налагодження та обслуговування спрощуються.

У .NET доступна велика кількість бібліотек, фреймворків і інструментів для виконання різноманітних завдань, таких як обробка даних і керування мережевим трафіком. Широкий вибір інструментів у .NET надає значну гнучкість.

Система складається з наступних основних компонентів зображених на рисунку 3.2. Вигляд кожного компонента наведений в Додатку Б.

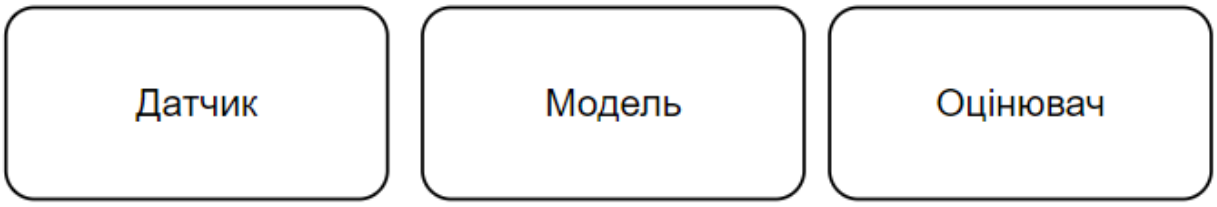


Рисунок 3.2 – Основні компоненти системи.

Датчик збирає дані про мережевий трафік. Дані будуть збиратися із вхідних пакетів, які будуть надходити у систему.

Модель використовується для виявлення атак. Модель реалізована за допомогою нейронної мережі, навченої на наборі даних KDD

Оцінювач використовується для оцінки ефективності моделі. Оцінювач може використовує комбінування метрик F1-міри та Специфічності.

Потік даних у системі зображений на рисунку 3.3. Він показує, яким чином дані будуть проходити крізь систему і які етапи повинен пройти пакет, щоб отримати оцінку і винесення результату.

Датчик збирає дані про мережевий трафік - Це початковий етап, де дані про мережевий трафік збираються.

Отримані дані від датчика передаються до моделі для аналізу у вигляді потоку даних у реальному часі.

Модель аналізує дані та генерує результати виявлення атак за допомогою моделі машинного навчання, тут відбувається класифікація та виявлення потенційно шкідливого трафіку.

Оцінювач отримує результати виявлення атак від моделі для подальшої обробки та аналізу.

Оцінювач обчислює ефективність результатів виявлення атак з використанням різних метрик, таких як F1-міра, специфічність, це дозволяє визначити, наскільки ефективно система впоралася з виявленням атак та якість її роботи.

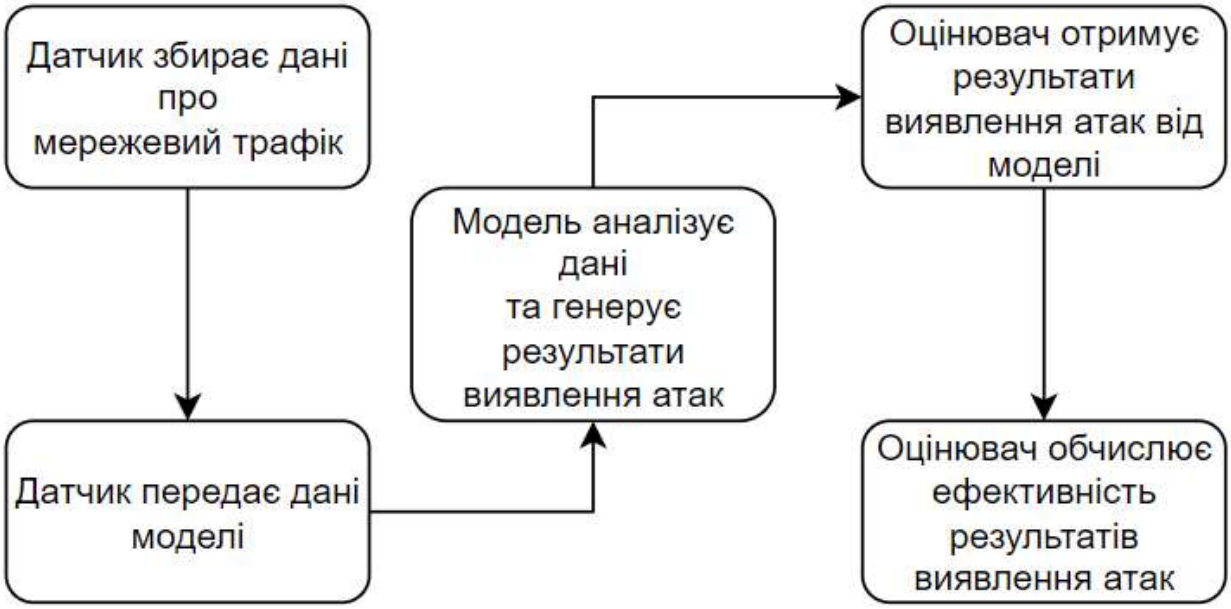


Рисунок 3.3 – Потік даних у системі.

Створення функцій для читання та завантаження даних з файлів KDD Cup 99 Dataset у вигляді, придатному для подальшої обробки.

Виявлення та обробка відсутніх значень, дублікатів або аномальних записів для забезпечення якості даних.

Вибір та виділення важливих ознак, які є ключовими для аналізу мережевого трафіку та виявлення DDoS-атак.

Важливим етапом є підготовка початкової моделі, а саме обробка даних KDD Cup 99 Dataset і це буде впроваджено наступними етапами:

– завантаження даних;
– перевірка та очищення даних;
– фільтрація та вибір ознак;
– перетворення формату даних;
– нормалізація та шкалювання.

Конвертація даних до формату, який можна використовувати в моделях машинного навчання (наприклад, числові представлення). Приведення значень ознак до одного масштабу для оптимізації роботи моделей.

Модуль машинного навчання з використанням ML.NET для виявлення DDoS-атак на основі даних KDD включає наступні етапи:

– використання ML.NET для навчання моделі на підготовлених даних KDD, де вже визначені мітки атак та нормального трафіку;
– перевірка ефективності навченої моделі за допомогою тестових даних;
– застосування навченої моделі до нових даних для виявлення потенційних DDoS-атак.

Реалізація цього модулю здійснена за допомогою мови програмування C# та використанням відповідних бібліотек для роботи з даними, таких як ML.NET для завантаження, обробки та підготовки KDD Cup 99 Dataset.

Отримання F1-міри та Специфічності є важливим етапом для оцінки ефективності моделей, кроки оцінювання представлені нижче:

– обчислення F1-міри;
– обчислення Специфічності (Specificity);
– модуль для обчислення;
– порівняння моделей.

F1-міра об’єднує точність (Precision) і чутливість (Recall) моделі. Вона обчислюється як гармонічне середнє між точністю і чутливістю і буде прекрасним показником ефективності згорткової мережі.

Специфічність (або True Negative Rate) визначає, яку частину істинно негативних зразків модель класифікувала правильно.

Функції або методи, які отримають прогнози вашої моделі для тестових даних та порівняють їх зі справжніми мітками для розрахунку F1-міри та Специфічності.

Після отримання цих метрик для кожної моделі, ви зможете порівняти їхню ефективність. Вибір найкращої моделі може базуватися на високих значеннях F1-міри та Специфічності.

Реалізація цих обчислень виконана у вигляді окремих функцій та класів, які приймають прогнози моделі та справжні мітки тестових даних, обчислюють відповідні метрики та повертають їх для подальшого порівняння та аналізу ефективності моделей.

3.2 Розробка згорткових нейронних мереж

Створення згорткової нейронної мережі (CNN) на C# для аналізу мережевого трафіку за допомогою TensorFlow або ML.NET складається з кількох кроків:

- попередня обробка даних;
- побудова моделі;
- навчання моделі.

Для реалізації нейронної мережі буде використовуватися бібліотека *Microsoft.ML,* яка містить усі необхідні методи, щоб можна було створити, натренувати та використовувати нейронні мережі, також тут доступні методи машинного навчання.

Тепер необхідно визначити клас, який буде відповідати за схему моделі, по якій і буде визначатися пакет для мережі, пакет міститиме великий набір параметрів і буде використовуватися для навчання і в подальшому трафік буде перетворюватися у поля цього класу, щоб проводити порівняння.

Клас міститиме 41 поле, тому повний вигляд із усіма властивостями наведений у Додатку В, він визначає схему набору даних.

Не повний вигляда класу, через велику кількість полів буде представлено нижче:

```
using Mucrosoft.ML.Data

public class NetworkData
{
        [LoadColumn(0)]
        Public float Duration {get;set;}

        [LoadColumn(1)]
        Public float Duration {get;set;}

        [LoadColumn(2)]
        Public string Service{get;set;}

        //Other fields

        [LoadColumn(41)]
        Public float Duration {get;set;}
}
```

Коли уже визначена схема моделі і після завантаження тестових даних KDD, потрібно реалізувати інтеграцію даних і налаштувати конвеєр попередньої обробки даних, після якого буде доступна натренована модель.

Отже, створення згорткової мережі включатиме наступні етапи:

– підготовка даних;

– побудова моделі;

– навчання моделі;

– оцінка моделі;

– збереження моделі.

Для завантаження даних, використовується *MLContext* із бібліотеки ML.NET, а попередня обробка даних, включатиме розділення на навчальні та тестові дані і це визначатиме потік обробки даних, який перетворить значення та складе їх у вектори для моделі.

Після чого, попередньо оброблений потік даних, буде використовуватися у згортковій нейронній мережі, де потрібно вказати параметри: колонки з мітками, особливості, кількість класів і епох навчання.

Побудований піплайн для навчання моделі з навчальними даними буде використовуватися для оцінки результатів, оцінка ефективності моделі виконуватиметься за допомогою метрик.

Вигляд сервісу, який займається завантаженням даних, розділенням цих даних та створює потік обробки даних, має наступний вигляд:

```
public class DataLoaderService: IDataLoaderService
{
	pubic IDataView LoadData(MLContext mlContext, string dataPath)
	{
		return mlContext.Data.LoadFromTextFile<NetworkData>(dataPath, hasHeader: true, separatorChar: ‘,’);
	}
	public (IDataView trainData, IDataView testData) SplitData(MLContext mlContext, IDataView dataView)
	{
		var splitData = nlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);
		return (splitData.TrainSet,splitData.TestSet);
	}
	public IEstimator<ITramsformer> CreateDataProcessingPipeline(MLContext mlContext)
	{
		return mlContext.Transforms.Conversion.MapValueToKey("Label", "IsMalicious")
			.Append(mlContext.Transforms.Concatenate("Features", "Duration" "ProtocolType", "Service", "Flag", "SrcBytes", "DstBytes", "Land"));
	}
}
```

У даному класі відбувається завантаження даних моделі для подальшого проведення машинного навчання, також відбувається розбивка даних, оскільки вхідний файл, по стандарту повинен містити формат .cvs. Наступний метод створює пайплайн процес, який виконує трансформацію по полях вхідного файлу, щоб бібліотека і нейронна мережа розуміла на що звертати увагу і яким чином проставляти коефіцієнти для навчання, це насправді самий важливий метод у цьому класі, оскільки від нього залежить якість навченої мережі.

Важливим методом для навчання моделі є перетворення файлу із набором тестових даних у клас, який використовується для навчання, такий метод матиме наступний вигляд:

```
public static List<T> ReadFromCSV<T>(string filePath)
{
	var config = new CsvConfiguration(CultureInfo.InvariantCulture)
	{
		HasHeaderRecord = true,
		Delimiter = ","
	};

	using var reader = new StreamReader(filePath);
	using var csv = new CsvReader(reader, config);
	var records = csv.GetRecords<T>().ToList();
	return records;
}
```

Тепер необхідно розробити сервіс для побудови та навчання моделі, яка буде навчатися за попередньо визначеним конвеєром, також сервіс виконуватиме оцінку моделі і збереження результатів, і матиме наступний вигляд:

```
public class ModelTrainerService: IModelTrainerService
{
	public ITransformer TrainModel(MLContext mlContext, IDataView trainData, IEstimator<ITransformer> pipeline)
	{
		return pipeline.Fit(trainData)
	}

	public (double microAccuracy, double macroAccurace) EvaluateModel(MLContext mlContext, ITransformer trainedModel, IDataView testData)
	{
		var predictions = trainedModel.Transform(testData);
		var metrics = mlContext.MulticlassClassification.Evaluate(predictions, labelColumnName: "Label", predictedLabelColumnName: "PredictedLabel");
		return (metrics.MicroAccuracy, metrics.MacriAccuracy);
	}

	public void SaveModel(MLContext mlContext,  ITransformer trainedModel, DataViewSchema schema, string modelPath)
	{
		mlContext.Model.Save(trainedModel, schema, modelPath);
	}
}
```

Тут використовується впровадження залежностей, щоб в подальшому можна було розділяти реалізацію, від основних методів, які буде використовувати система, вигляд інтерфейсів наведено нижче:

```
public interface IDataLoaderService
{
	IDataView LoadData(MLContext mlContext, string dataPath);
	(IDataView trainedModel, IDataView testData) SplitData(MLContext mlContext, IDataView dataView);
	IEstimator<ITransformer> CreateDataProcessingPipeline(MLContext mlContext);
}

public interface IModelTrainerService
{
	ITransformer TraineModel(MLContext mlContext, IDataView trainedModel, IEstimator<ITransformer> pipeline);
	(double microAccuracy, double macroAccuracy) EvaluateModel(MLContext mlContext, ITransformer trainedModel, IDataView testData);
	void SaveModel(MLContext mlContext, ITransformer trainedModel, DataViewSchema schema, string modelPath);
}
```

Для використання Dependency Injection потрібно додати налаштування сервісів, це виконується за допомогою контейнера *ServiceCollection.* Основна форма отримує екземпляри сервісів через конструктор, що дає можливість використати їх у подіях, які будуть прив’язані до візуальних елементів форми. Це найкращий спосіб організації модулів та компонентів додатку, що в подальшому дозволить протестувати ці компоненти окремо.

Взаємодія із описаними сервісами виконується через змінні інтерфейсів, які створюють під час компіляції додатку і будуть відповідати лише одному значенні на рівні тієї форми, де вони викликаються, це матиме наступний вигляд:

```
private void ConfigureServices(IServiceCollection services)
{
    services.AddSingleton<IDataLoaderService, DataLoaderService>();
    services.AddSingleton<IModelTrainerSerivce, ModelTrainerSerivce();
}
```

При подальшій розробці, кількість сервісів буде збільшуватися, а зараз розглянемо код, який відповідатиме за тренування моделі:

```
public void TrainModelButton_Click(object sender, EventArgs e)
{
	var mlContext = new MLContext9);
	var dataPath = Path.Combine(Enviroment.CurrentDirectory, "data.cvs");
	var dataView = _dataLoaderService.LoadData(mlContext, dataPath);
	var dataProcessPipeline = _dataLoaderService.CreateDataProcessingPipeline(mlContext);
	var pipeline = dataProcessPipeline
		.Append(mlContext.MulticlassClassification.Trainers.ConvolutionalNeuralNetwork(
		labelColumnName: "Label", featureColumnName: "Features", numberOfClasses: 2, epochs: 10));
	var trainedModel = _modelTrainerService.TrainModel(mlContext, trainData, pipeline);
	var (microAccuracy, macroAccurace) = _modelTrainerService.EvaluateModel(mlContext, trainedModel, testData)

	MessageBox.Show($"Micro Accuracy: {microAccuracy}\n Macro Accurace: {macroAccurace}");

	_modelTrainerService.SaveModel(mlContext, trainedModel, trainedData.Sche,a, "model.zip");
}
```

Отже, щоб побудувати згорткову нейронну мережу, потрібно виконати наступні дії, це підготувати дані і розділити на набори, які будуть задіяні у навчанні та тестуванні, використовуватися для цього буде набір даних та додаткові налаштування машинного навчання, код наведено нижче:

```
var mlContext = new MLContext();
var dataView = mlContext.Data.LoadFromEnumerable(dataList);

var trainTestSplit = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);
var trainingData = trainTestSplit.TrainSet;
var testingData = trainTestSplit.TestSet;
```

Тут відбувається перетворення колонок у числові значення, які використовуються для прогнозування у натренованій моделі. Також виконується об’єднання ознак і перетворення їх у вектор, щоб використати для навчання. Наступний етап це нормалізація даних до діапазону між 0 та 1. Далі відбувається завантаження моделі TensorFlow, яка використовується в ML.NET.

Після чого відбувається визначення архітектури CNN за допомогою TensorFlow, де виконується побудова пайпалайну для бінарної класифікації, код наведено нижче:

```
var pipeline = mlContext.Transforms.Conversion.MapValueToKey(
"Label", nameof(NetworkData.IsMalicious))
   .Append(mlContext.Transforms.Concatenate("Features",
      nameof(NetworkData.Duration),
      nameof(NetworkData.ProtocolType),
      // Include other features...
   ))
   .Append(mlContext.Transforms.NormalizeMinMax("Features"))
   .Append(mlContext.Model.LoadTensorFlowModel("Path/To/Your/TensorFlowModel"))
   .Append(mlContext.BinaryClassification.Trainers.LbfgsLogisticRegression("Label"));
```

Також тут використовується алгоритм логістичної регресія, який потрібен для бінарної класифікації моделі. Це дозволяє створити послідовний пайплайн операцій обробки та моделювання даних, який побудує моделі класифікації на основі даних, і потрібен для передбачення на нових даних, у нашому випадку для аналізу мережевого трафіку.

Важливим кроком є процес навчання моделі та її застосування на тестових даних, щоб оцінити її точність, використавши бінарні метрики класифікації, вигляд процесу навчання наведено нижче:

```
var trainedModel = pipeline.Fit(trainData);

var predictions = trainedModel.Transform(testData);
var metrics = mlContext.MulticlassClassification.Evaluate(predictions, labelColumnName: "Label", predictedLabelColumnName: "PredictedLabel");
```

Тут використовується побудований пайплайн, який містить попередньо визначені кроки підготовки даних та моделювання, результат навчання має на виході навчену модель, яка використовуватиметься для оцінки трафіку.

Щоб використати навчену модель, потрібно створити екземпляр класу, який відповідатиме полям нашої визначеної моделі *NetworkData*, важливо, що поля

повинні відповідати усім колонкам, і розробити клас для виведення і представлення результатів, код наведено нижче:

```
var newData = new NetworkData();
var predictionEngine = mlContext.Model.CreatePredictionEngine<NetworkData,
        YourPredictionClass>(trainedModel);
var prediction = predictionEngine.Predict(newData);
```

Клас результатів, який створюватиме навчена модель під час аналізу мережі міститиме поля:

– логінчка властивість, яка вказує, чи пакет класифікується як зловмисний чи нормальний;

– оцінка достовірності, пов’язана із аналізом;

– властивість для представлення типу атаки.

У цьому висновку після аналізу пакету, фіксуватиметься результат роботи згорткової мережі, та додаватимиться додаткова інформація пов’язана із типом атаки, яка допоможе інтерпретувати та проаналізувати результати аналізу, ці результати будуть представленні у вигляді візуальної інтерпретації, щоб можна було відслідкувати атаку.

Щоб оцінити модель спрогнозувавши тестові дані, буде використовуватися *BinaryClassificationMetrics* для обчислення різних показників. Також тут буде виконуватися розранунох спецефічності та оцінка F1-метрики в цьому ж об’єкті, код оцінювання наведений нижче:

```
var metrics = mlContext.BinaryClassification.Evaluate(predictions, labelColumnName: "Label");
var f1Score = metrics.F1Score;
var specificity = metrics.ConfusionMatrix.GetSpecificity();

var precision = metrics.Precision;
var recall = metrics.Recall;
var accuracy = metrics.Accuracy;
```

Ці показники оцінки допомагають оцінити продуктивність моделі в завданнях двійкової класифікації, надаючи розуміння її точності, запам’ятовування та загальної точності як позитивних, так і негативних прогнозів класу.

Далі ці результати будуть зберігатися і відображатися у вигляді графіку, який розділиться відповідно до метрик, для цього потрібен клас:

```
public class EvaluationResult
{
   public string Metric { get; set; }
   public double Value { get; set; }
}
```

Результати цього класу будуть компонуватися у список, який передаватиметься для відображення на графіку.

Також потрібно надати можливість поступового навчання, щоб оновлювати та додатково навчати модель. Коли надходять нові пакети, можна викликати метод *RetrainOnline*, щоб виконувати поступове навчання, після внесення змін у модель її потрібно оцінити і зберегти оновлення, якщо результати будуть задовільняти наші потреби у аналізі.

Щоб ефективно налаштувати навчання моделі, потрібно розглянути підхід до оновлення моделі, який буде мати стратегію або навчання з кожного пакету або навчання через деякий час.

Перевагою поступового навчання – є можливість безперервно навчатися та адаптуватися до нових шаблонів даних.

Поступове навчання може призвести до погіршення моделі, якщо вхідні дані не опрацьовувати і не робити правильну оцінку результатів.

Переваги пакетного навчання – це забезпечує більш стабільні та контрольовані оновлення моделі, тобто ми можемо відсіяти непотрібно пакети, які можуть погіршити нашу модель, попередньо оцінивши їх.

Недоліком – є неможливість зафіксувати миттєві зміни даних і цей метод може вимагати повторної обробки великих наборів даних.

Найкращий метод навчання залежить від конкретного випадку використання, обмежень і поведінки ваших даних. Тут буде використовуватися пакетний метод оцінювання, тобто нові пакети, які будуть надходити компонуються у список, який перед початком додаткового тренування буде оцінений і пакети, які можуть нашкодити моделі, будуть вилучені із нього.

3.3 Розробка системи і тренування моделі

C# надає великий спектр можливостей в розробці програмного забезпечення і інструментів для розробників, як вже зазначалося раніше, основними елементами системи будуть: датчик, модель, оцінювач. Оскільки вже була проведена реалізація створення тренованої моделі і перетворення результатів аналізу у структуровані дані, тобто була виконана основна робота з нейронною мережею, залишається реалізувати програмні компоненти, щоб можна було протестувати модель і побачити зрозумілий результат.

Розпочнемо із створення компонента датчик, який повинен приймати вхідний трафік і перетворювати його у потрібну нам модель, щоб нейронна мережа могла надати своє передбачення.

Потрібно вибрати пристрій на який будуть прийматися вхідні пакети, тобто в системі може бути декілька таких пристроїв, потрібно вивести їх список і дати можливість вибрати, з чим завданням допоможе бібліотека *SharpPcap*, як це зробити наведено нижче:

```
var devices = CaptureDeviceList.Instance;
```

Буде використовуватися бібліотека *SharpPcap* для захоплення пакетів з мережевого інтерфейсу. Перехоплення пакетів передбачає перехоплення та аналіз пакетів даних, які проходять через мережевий інтерфейс. Цей дозволяє перевіряти необроблені дані в цих пакетах, витягувати корисну інформацію та потенційно виконувати дії на основі отриманих даних.

Далі необхідно відкрити обраний девайс для спостереження, визначити метод обробки захоплених пакетів, та за допомогою *PacketDotNet* отримати усю інформацію про пакет, код наведено нижче:

```
var device = devices[0];

device.OnPacketArrival += (sender, e) =>
{
   var packet = Packet.ParsePacket(e.Packet.LinkLayerType, e.Packet.Data);
};

device.Open(DeviceMode.Promiscuous);
device.StartCapture();
```

Отже, потрібно відокремити спосіб вилучення ознак з пакету, оскільки мережеві пакети мають особливості у своїх заголовках, також потрібно проаналізувати вміст корисного навантаження у пакеті, щоб виконувати більш детальний аналіз, реалізація наведена нижче:

```
private static void OnPacketArrival(object sender, CaptureEventArgs e)
{
   var packet = Packet.ParsePacket(e.Packet.LinkLayerType, e.Packet.Data);

   var ipPacket = packet.Extract<IPv4Packet>();
   if (ipPacket != null)
   {
      var sourceIP = ipPacket.SourceAddress;
      var destinationIP = ipPacket.DestinationAddress;
      var protocol = ipPacket.Protocol;
      var packetSize = e.Packet.Data.Length; // Total packet size
      var networkData = new NetworkData
      {
         SourceIP = sourceIP,
         DestinationIP = destinationIP,
         Protocol = protocol,
         PacketSize = packetSize
         // Add other relevant features to your NetworkData class
      };
   }
}
```

Варто зауважити, що такий тип роботи вимагає надання системі певних дозволів, щоб додаток працював правильно, оскільки перехоплення мережевого трафіку може класифікуватися, як зловмисне використання.

Пакетні дані знаходяться в центрі виявлення та класифікації вторгнень, де необхідно витягти відповідні функції для розробки моделі машинного навчання. Основне завдання полягує у розробці формату, який готує необроблені пакетні дані для навчання.

Отже, реалізація самого датчика, де *PacketSencosr* прослуховує надходження пакетів по IPv4, який буде нашим основним шлюзом для відстеження і коли може його розшифрувати перетворишви у модель з якою вже може працювати нейронна мережа, то система виконує подію запхоплення пакету, буде мати наступний вигляд:

```
public class PacketSensor
{
   private CaptureDevice _device;

   public event EventHandler PacketArrived;

   public PacketSensor()
   {
      var devices = CaptureDeviceList.Instance;
      _device = devices.FirstOrDefault();

      if (_device != null)
      {
         _device.OnPacketArrival += OnPacketArrival;
         _device.Open(DeviceMode.Promiscuous);
         _device.StartCapture();
      }
      else
      {
         Console.WriteLine("No capture device found.");
      }
   }

   private void OnPacketArrival(object sender, CaptureEventArgs e)
   {
      PacketArrived?.Invoke(this, EventArgs.Empty);
   }
}
```

Цей код фіксує пакети за допомогою SharpPcap, витягує основні характеристики з пакетів IPv4 (IP-адреса джерела, IP-адреса призначення, протокол, розмір пакета) і повертає значення у потрібному форматі.

Підготовка даних для навчання передбачає перетворення отриманих пакетних даних у формат, придатний для навчання моделі машинного навчання. У контексті аналізу мережевих пакетів для виявлення вторгнень це включає вилучення відповідних функцій і впорядкування їх у структурований набір даних.

У системі передбачений лог вивід даних, куда буде записуватися інфомрація про кожен пакет, дані будуть відображатися в реальному часу коли запускається датчик перехоплення, код наведено нижче:

```
public MainForm()
{
   InitializeComponent();
   _packetSensor = new PacketSensor();
   _packetSensor.PacketLogged += OnPacketLogged;
}

private void OnPacketLogged(object sender, string packetInfo)
{
   listBoxPacketLog.Items.Insert(0, packetInfo); // Display the packet info in the listbox
}
```

Також в режимі реального часу буде показано результати роботи аналізатора і виводитиметься кількість отриманих пакетів разом із їх оцінкою. У PacketSensor потрібно додати нові поля, які будуть виконувати функцію збіру даних, код наведено нижче:

```
public event EventHandler<Tuple<int, int>> StatisticsUpdated; // Event to pass statistics (malicious, trusted)

private int _maliciousPacketCount = 0;
private int _trustedPacketCount = 0;
```

Щоб дані на гістрорамі оновлювалися в реальному часі, потрібно додати подію, яка буде викликатися кожен раз, як оцінка моделі буде успішно відбуватися і нейронна мережа видаться результат перевірки.

Вигляд графіка на головній сторінці системи, а також загальна кількість перевірених пакетів за сеанс, буде мати вигляд, як на рисунку 3.1, користувацький інтерфейс наведений у Додатку Б.

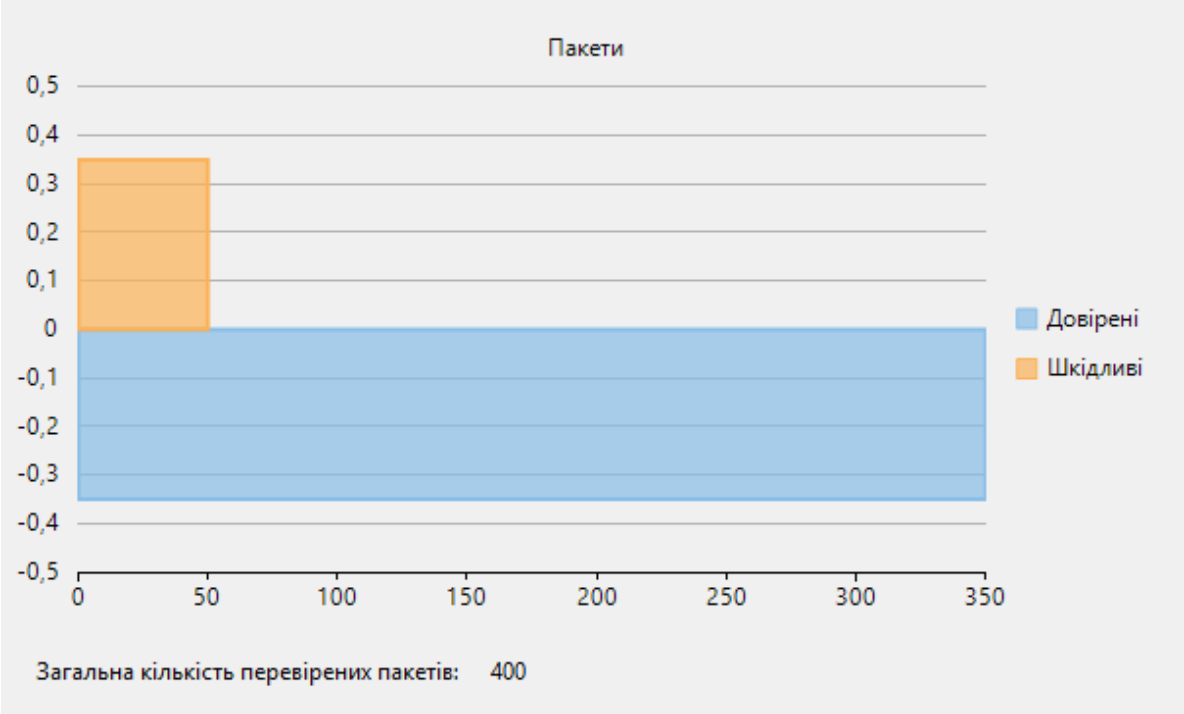


Рисунок 3.1 – Вигляд графіка на головній сторінці.

3.4 Висновок до розділу

У цьому розділі була реалізлована система для аналізу мережевого трафіка на онснові шлибоких згорткових нейронних мереж з використанням машиного навчання без учителя, яка буде додатково навчатися на наборах пакетів, які будуть надходити пілся проведення основного тренування моделі. Додаткове тренування буде проводитися поступово і буде підвантажувати у модель великий набір вхідних даних, після чого відбуватиметься оцінка цих даних і у випадку високого показника результативності відбуватиметься оновлення моделі.

Для ефективної роботи системи, була продумана архітектура додатку та спосіб створення згорткової мережі. Використовуючи механізми впровадження залежностей і техніки машиного навчання, кодова база системи стала простою в

тестуванні та має великий потенціал для розширення і додавання нового функціоналу.

Додано можливість вибирати мережевий адаптер з якого буде аналізуватися вхідний трафік на наявність шкідливих пакетів, а також можливість налаштування додаткового навчання моделі на основі тих самих відних даних з адаптера. Оскільки модель має певні стандарти, яких потрібно притримуватися, щоб правильно оцінити мережевий пакет, було прийнятно рішення відмовитися від можливості додавання свого тестового набору даних, це допоможе відштовхуватися від одних і тих самих початкових даних і досліджувати ефективність додаткового навчання нейронної мережі.

У додатку були використанні WinForms, це надало можливість швидкої розробки користувацького інтерфейсу і дало можливість сконцентруватися на створенні глибокої згорткової мережі і відображенні результатів оцінювання пакетів у зручний спосіб, щоб можна було самостійно оцінювати на графіках ефективність методу. Також додано загальний звіт за сеанс із усіма необхідними метриками, які потрібні для оцінки ефективності запропонованого методу.

4 ОГЛЯД РЕЗУЛЬТАТІВ РОБОТИ МЕТОДУ ВИЯВЛЕННЯ ШКІДЛИВИХ ПАКЕТІВ ЧЕРЕЗ АНАЛІЗ МЕРЕЖЕВОГО ТРАФІКУ З ВИКОРИСТАННЯМ ГЛИБОКИХ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

4.1 Розробка компонентів тистування системи

Для того, щоб перевірити ефективність роботи системи, потрібно підготувати ще один проєкт, який буде зберігати в собі тестові дані для аналізу і який буде працювати через сокети в даному випадку імітуючи поведінку мережі, але у локальному варіанті для більшої контрольованості процесу.

У локальному середовищі знадобиться кінцева точка прийому для надсилання пакетів. Потрібно створити прослуховуючу програму на нашому порту, який може отримувати ці пакети.

Перший етап буде відправляти випадкові пакети і система буде їх усі відловлювати та оцінювати, без додаткового навчання, після чого буде виконана спроба додатково навчити модель цими випадковими даними, які завжди будуть унікальні.

Після чого будуть запропоновані тест-кейси, які потрібні для більш конкретної оцінки і будуть надсилатися вручну з програми тестування, а вже за її результатами будуть виводитися результати роботи.

Аномальний трафік буде основною складовою перевірки, для цього:

- пакети будуть формуватися з неправильними заголовкам;
- у пакетів буде порушена структура;
- вхідні порти будуть нестандартні цим типам протоколів;
- додавання незвихчин даних у пакети;
- різний розмір пакетів і їх навантаження;
- час надсилання буде кожен раз різний і проміжки між ними.

Для створення пакетів із зазначеними вижче параметрами, знадобиться додаткова бібліотека *PcapDotNet*, яка надає доступ до мережі нижнього рівня і має великий функціонал по роботі із пакетами.

Отже, щоб надсилати велику кількість різних пакетів, було написано компонент по генерації випадкових пакетів, який має наступний вигляд:

```
for(int I = 0; I < 100; i++)
{
    Packet packet = RandomizeInputAndCraftPacket();
    SendTcpPacket(packet, ProtocolType.Tcp);
}
```

Тут можна змінити кількість пакетів, які потрібно надіслати і для прикладу представлений пакет по протоколу TCP, а сам код генерації пакету приведений нижче:

```
static Packet RandomizeInputAndCraftPacket()
{
   Random random = new Random();

   // Generate random IP addresses
   string srcIp = $"{random.Next(256)}.{random.Next(256)}.{random.Next(256)}.{random.Next(256)}";
   string dstIp = $"{random.Next(256)}.{random.Next(256)}.{random.Next(256)}.{random.Next(256)}";

   // Generate random port numbers
   ushort srcPort = (ushort)random.Next(ushort.MaxValue + 1);
   ushort dstPort = (ushort)random.Next(ushort.MaxValue + 1);

   // Generate random payload
   byte[] payloadBytes = new byte[10];
   random.NextBytes(payloadBytes);
   string payload = Encoding.ASCII.GetString(payloadBytes);

   return CraftTcpPacket(srcIp, dstIp, srcPort, dstPort, payload);
}
```

У нас є можливість власноруч налаштовувати куда буде відправлятися пакет, тобто в який саме мережевий адаптер сокет відправить дані, що потрібно, оскільки

система буде прослуховувати лише вибраний адаптер в налаштуваннях, а отже потрібно правильно співставити кінцеві точки.

Для того щоб забезпечети непердбачуваний трафік і протестувати модель на складних і не структурованих, непослідовних, даних, цей метод випадковим чином генерує дані про пакет, у прикдаді тут використовується лише один тип протоколу, але їх варіативність також описана і вибирається випадково, також потрібно визначити спосіб відправки завчасно створених тестових пакетів із передвачуваною відповіддю для нас.

Тепер потрібно створити пакет для відправки, метод буде заповнювати заголовки і усе необхідне корисне навантаження пакету враховуючи випадково згенеровані дані:

```
static Packet CraftTcpPacket(string srcIp, string dstIp, ushort srcPort, ushort dstPort, string payload)
{
    EthernetLayer ethernetLayer = new EthernetLayer
    {
       Source = new MacAddress("00:00:00:00:00:00"),
       Destination = new MacAddress("00:00:00:00:00:00"),
       EtherType = EthernetType.None, // Change if required
    };
    IpV4Layer ipV4Layer = new IpV4Layer
    {
       Source = new IpV4Address(srcIp),
       CurrentDestination = new IpV4Address(dstIp),
       Protocol = IpV4Protocol.Tcp,
    };
    Random random = new Random();
    TcpControlBits randomControlBits = (TcpControlBits)random.Next(Enum.GetValues(typeof(TcpControlBits)).Length);

    TcpLayer tcpLayer = new TcpLayer
    {
       SourcePort = srcPort,
       DestinationPort = dstPort,
       ControlBits = randomControlBits,
    };
    PayloadLayer payloadLayer = new PayloadLayer
    {
       Data = new Datagram(Encoding.ASCII.GetBytes(payload)),
    };
    PacketBuilder builder = new PacketBuilder(ethernetLayer, ipV4Layer, tcpLayer, payloadLayer);
    return builder.Build(DateTime.Now);
  }
```

На основі цього сформують тестові дані і будуть передаватися для відправки у адаптер, а можливість надсилати велику кількість пакетів абсолютно різних типів це непогана перевірка адаптивності мережі і схоже на DoS атаку. Оскільки ми тестуємо в локальній мережі, то для відправки пакету будем використовувати сокети, а по замовчуваню задіяний основний мережевий адаптер і система прослуховуватиме саме його, код наведено нижче:

```
static void SendTcpPacket(string dstIp, int dstPort, string payload)
{
      try
      {
         Socket sender = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
         IPAddress ipAddress = IPAddress.Parse(dstIp);
         IPEndPoint remoteEP = new IPEndPoint(ipAddress, dstPort);

         sender.Connect(remoteEP);

         byte[] data = Encoding.ASCII.GetBytes(payload);
         sender.Send(data);

         sender.Shutdown(SocketShutdown.Both);
         sender.Close();
      }
      catch (Exception e)
      {
         Console.WriteLine($"Exception: {e}");
      }
   }
```

Отже ми отримали компонент для відправки пакетів, тепер розглянемо конкретні прикладти роботи системи і оцінки взідних мережевих пакетів. Перший крок - створення власного тестового набору даних, який буде включати як шкідливі, так і звичайні мережеві пакети. Ми можемо симулювати різні типи атак та нормальний мережевий трафік. алі, ми можемо використати нашу модель виявлення шкідливих пакетів, щоб класифікувати ці дані. Результати класифікації допоможуть нам оцінити ефективність моделі, після чого ми зможемо скористатися метриками, такими як F1-мера, точність, специфічність.

4.2 Тестування отриманої моделі

Підготуємо та завантажимо тестові дані, які будуть використовуватися для тренування моделі, після їх завантаження проведемо тренування глибокої згорткової нейронної мережі і оцінимо якість навченої моделі, яку збережему і будемо використовувати для подальших оцінок пакетів.

Давайте розглянемо результат навчання моделі на наборі даних, який має близько п’ятиста тисяч пакетів для перевірки, резлультати приведені на рисунку 4.1, який демонструє графік, звіт у системі, який показує якість навченої моделі у F1-мірі, спецефічності та точності. Результати та їх опис наведені у таблиці 4.1, яка опише значення кожної метрики, яке отримала модель і визначить ступінь ефективності запропонованого методу. Якщо модель буде мати високу ефективність, то метрики будуть наближені до значення 1.00, для F1 – це 0.9, для Спецефічності – це 0.95, а для Точності – це 1.00.

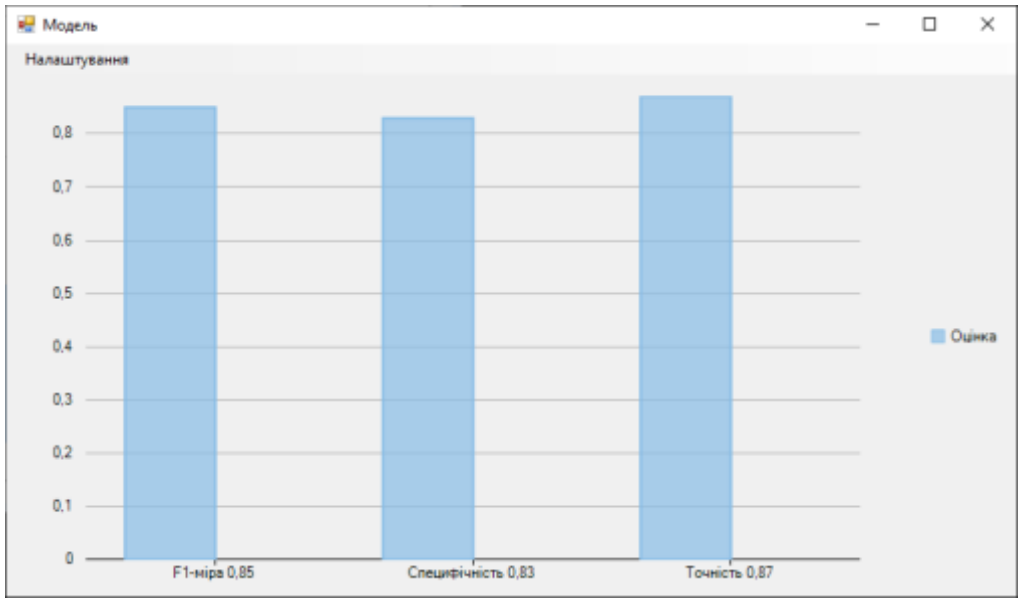


Рисунок 4.1 – Оцінка моделі після використання вхідного тестового набору.

Це дані нашої навченої моделі від якої ми будемо відштовхуватися, оскільки вона показала дуже прийнятні результати після навчання, прогнорзується, що і якість визначення пакетів буде відповідною.

Таблиця 4.1 – Результати оцінки моделі на тестових даних

Метрика	Оцінка	Опис
F1-міра	0.85	Значення 0.85 свідчить про те, що модель добре працює як у точності передбачень, так і в покритті справжніх позитивних випадків.
Специфічність	0.83	Значення 0.83 також є досить хорошим показником, свідчить про те, що модель правильно ідентифікує нешкідливі пакети. Здатності моделі визначати негативні класи можна оцінити, як високу.
Точність	0.87	Значення 0.87 є досить високим і може свідчити про те, що модель правильно класифікує обидва класи (шкідливі та нешкідливі пакети) з високою точністю.

Якщо точність досягла значення 0.87, це може бути показником того, що модель добре навчилася і показує високу правильність передбачень.

Також доступна функція додаткового навчання, яке буде проводитися по нових відловлених пакетах і можна налаштувати, параметри навчання. Форма налаштувань зображена на рисунку 4.2.

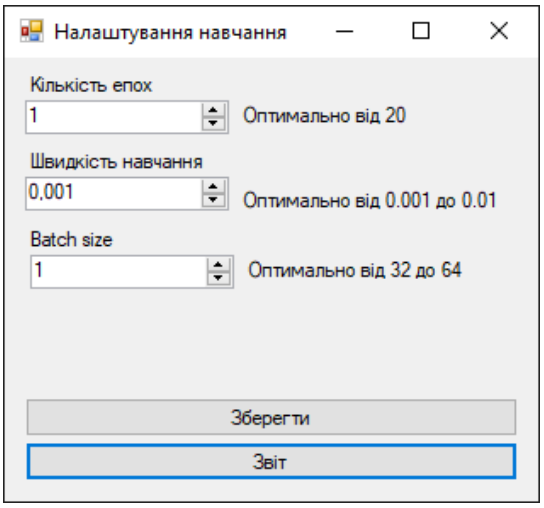


Рисунок 4.2 – Форма налаштувань навчання моделі.

Саме тут можна буде налаштувати додаткове навчання моделі і натиснувши на звіт, отримаємо зміни у оцінці моделі після додаткового навчання, для прикладу візьмемо оптимальні значення у вигляді: 20 епох, 0,01 швидкість та 64 розмір пакету і переглянемо результати. І перевіремо тренованість моделі на 3200 пакетах, які будуть абсолютно хаотично згенеровані нашим компонентом по генерації пакетів, але є важливий момент, значення можуть повторюватися, що буде негативно впливати на якість навчання.

Як наслідок повторюваності пакетів, можна припустити, що це спричинить лише погіршення ефективності моделі. Навіть якщо, згідно з графіком, спочатку може виникнути враження, що якість моделі зросла, це, швидше за все, зумовлене збільшенням кількості однотипних пакетів. Однак, в такому випадку, ефективність моделі буде зменшуватися поступово з часом, оскільки зростання числа однакових пакетів призведе до втрати різноманітності та репрезентативності даних для моделі. Отримані результати оцінки відображені на рисунку 4.3.

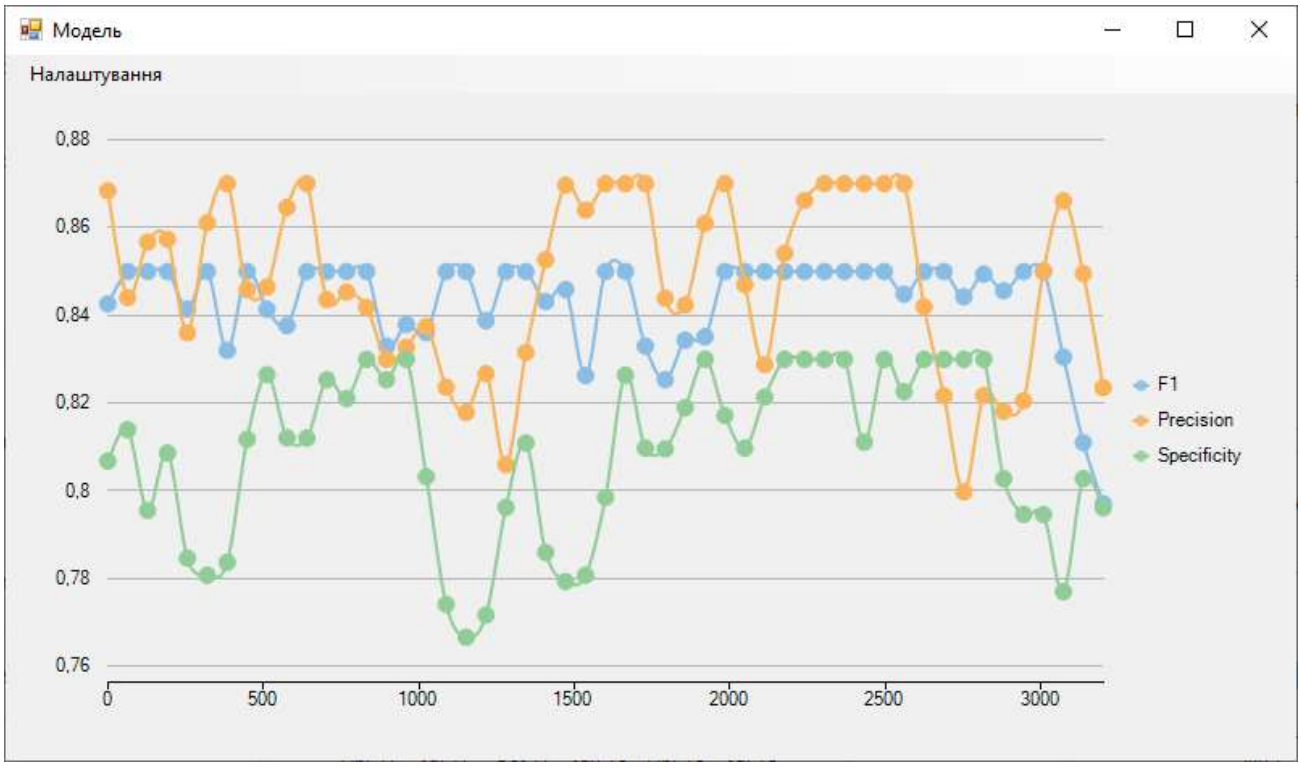


Рисунок 4.3 – Результати додаткового навчання на випадкових пакетах.

Давайте збільшимо кількість пакетів для тренування, прогнозуючи ще більше падіння оцінок моделі. Візьмемо 6400 пакетів, для прикладу, насправді їх можн бути довільна кількість і заново спробуємо навчити модель. Результатти будуть приведені на рсиунку 4.4.

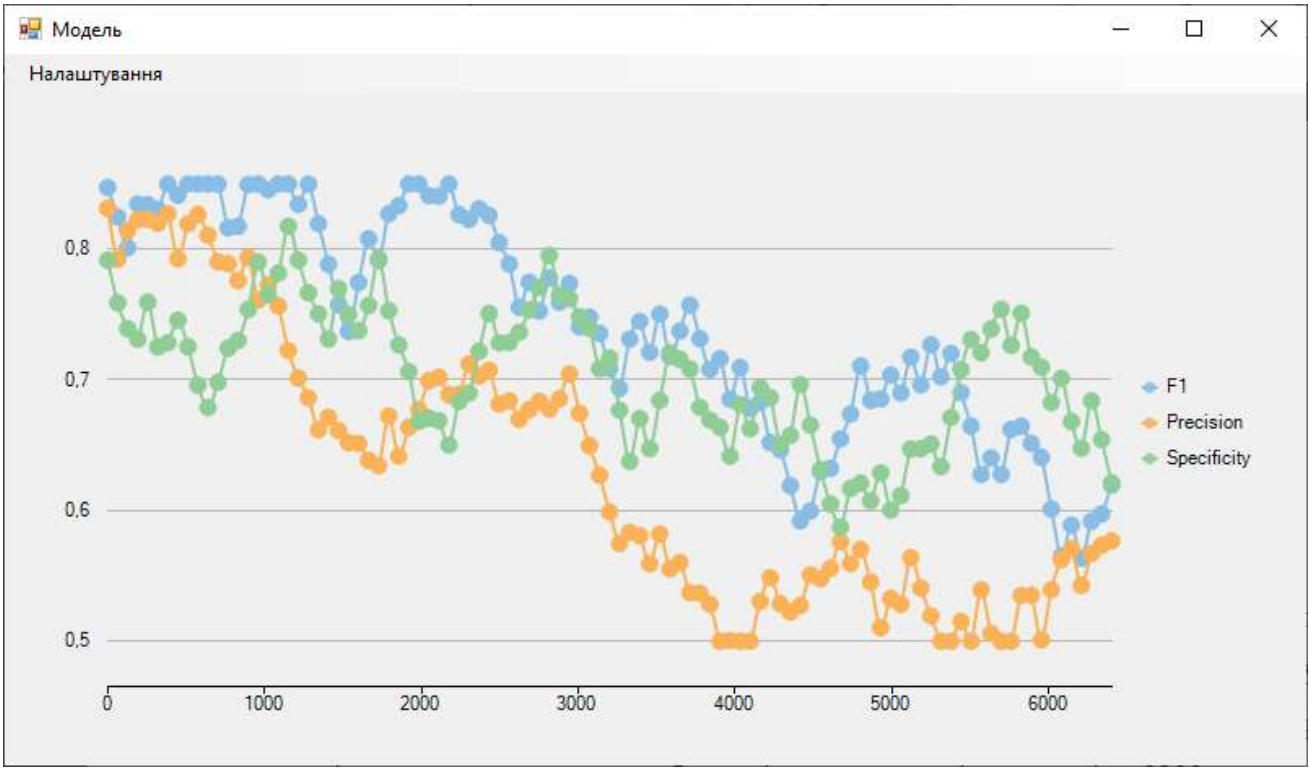


Рисунок 4.4 – Результати навчання на більшій кількості пакетів.

Робимо висновок, що якість вхідних даних вкрай важлива для точності моделі. Хоча на перший погляд може здатися, що на початку спостережень модель покращує свою ефективність, проте на довгостроковій відстані виявляється зниження її якості.

Розглянемо конкретний приклад, використовуючи реальні дані з мережевого трафіку, що надходить на мережевий адаптер. Ці дані є більш релевантними та мають більший потенціал для додаткового навчання моделі. Результати цих спостережень наведені на рисунку 4.5.

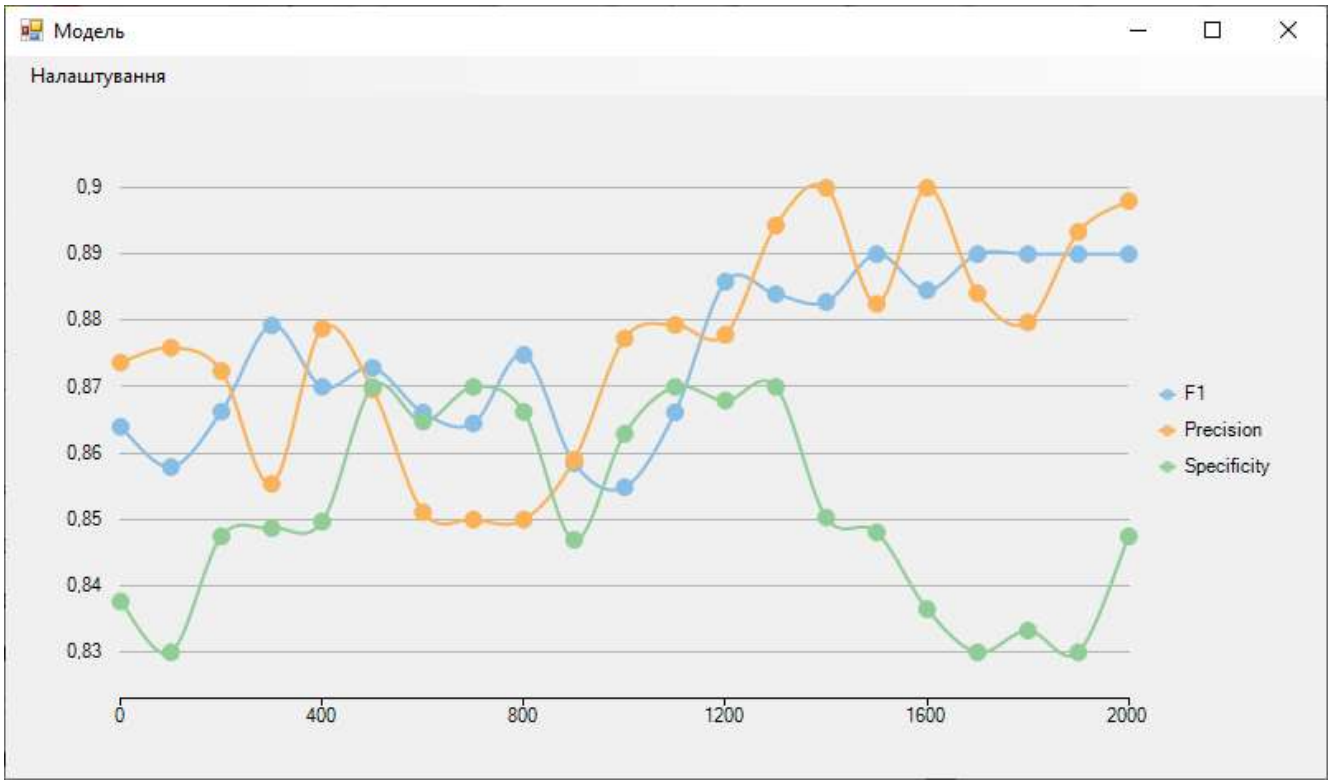


Рисунок 4.5 – Результати навчання на реальних мережевих пакетах.

Підсумовуючи, можна стверджувати, що досягнення значних покращень у роботі моделі можливе лише за умови високої якості вхідних даних. Якщо дані правильно підібрані, може спостерігатися лише незначна зміна у якості оцінки, що є нормальним явищем при навчанні моделі. Крім того, з аналізу графіків можна виокремити стабільну тенденцію, що свідчить про ефективність базової моделі.

Це може вказувати на те, що подальше покращення моделі може потребувати більш спеціалізованого підбору та формування набору даних. Однак, на даному етапі базова модель проявляє високу ефективність, що може вимагати більш глибокого та спеціалізованого підходу до подальшого навчання.

4.3 Оцінка ефективності система на основі аналізу мережевого трафіку з використанням глибоких згорткових нейронних мереж

Підготуємо набір тестових даних, що буде складатися зі звичайних мережевих пакетів, а також включатиме у себе пакети, що імітують спроби атак на мережу. Набір даних включатиме інформацію про потенційно шкідливі пакети, які можуть стати причиною атак на систему.

В таблиці 4.2 перший тестовий кейс складатиметься з пакетів із відомими вразливостями. Цей набір даних допоможе відстежити, як система реагує на шкідливі пакети та чи вдається їм пройти перевірку. Для цього створено віртуальний адаптер, на який будуть відправлятися пакети, після чого сама система буде слухати його та здійснювати оцінку отриманих даних.

В реальному прикладі роботи системи, потрібно підключатися і налаштувати основний адаптер, якщо система має декілька і потрібно класифіковувати усі, то варто запускати декілька додатків, оскільки він може перехоплювати лише і оцінювати лише один потік даних.

Таблиця 4.2 – Тест-кейс із відомими шкідливими пакетамию

Пакет	**Опис**
Звичайний	Відправник - 192.168.1.30, призначення - 8.8.8.8, тип - HTTP запит, порт - 80
	Відправник - 192.168.1.40, призначення - 8.8.8.8, тип - DNS запит, порт - 53
Шкідливий	Відправник - 192.168.1.10, призначення - 8.8.8.8, тип - SQL Injection, вміст - SQL запит, порт - 80
	Відправник - 192.168.1.20, призначення - 8.8.8.8, тип - Cross-Site Scripting (XSS), вміст - JavaScript код, порт - 443

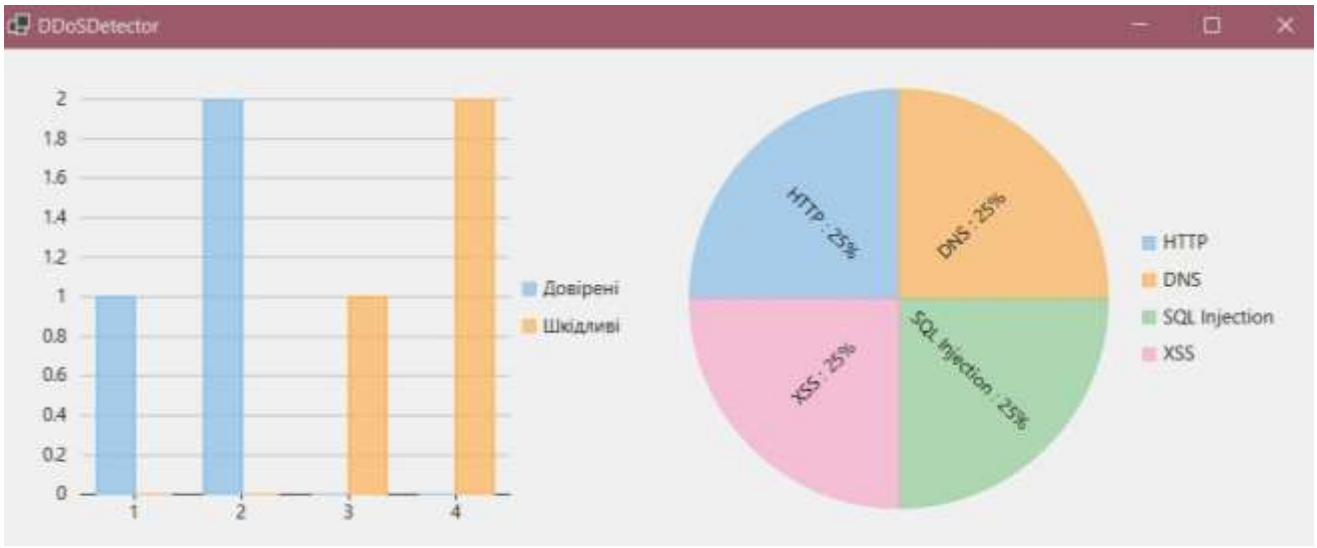


Рисунок 4.6 – Результати аналізу.

Як видно з діаграм, мережа виявилась ефективною у виявленні і класифікації вхідних пакетів, вірно показавши, який тип атаки був. Перейдемо до наступного тест кейсу. У цьому тест кейсі створимо вручну велику кількість шкідливих пакетів, використовуючи реальні приклади, та розбавимо їх звичайними, приклад, як будуть виглядати пакети знаходиться у табилці 4.3.

Таблиця 4.3 – Тест-кейс масштабна атака ботнету.

Пакет	**Опис**
Звичайний	Відправник - 192.168.1.38, призначення - 8.8.8.8, тип - HTTP запит, порт - 80
	Відправник - 192.168.1.100, призначення - 8.8.8.8, тип - DNS запит, порт - 53
Шкідливий	IP адреси з ботнету, призначення - 8.8.8.8, тип - TCP SYN Flood атака
	IP адреси з ботнету, призначення - 8.8.8.8, тип - UDP атака на DNS сервери

Пакети відправлялися хаотично і не послідовно, тобто міксувалися із звичайними, щоб більше приховати роботу ботнета і по відомим портам.

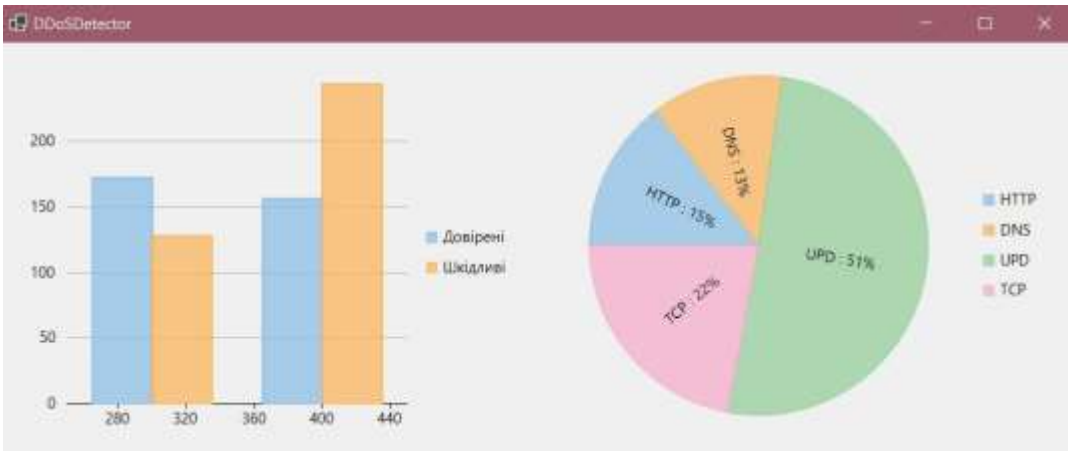


Рисунок 4.7 – Результати аналізу ботнет атаки.

Отже, мережа класифікувала практично усі шкідливі пакети, а решту відправила у поле довірені, судячи з діаграм, по великому сплеску пакетів і їх шкідливості, можна було би зробити висновок, що відбувається атака на мережу, і що потрібно реагувати, якщо дані таким чином будуть оновлюватися в реальному часі і логуватися, то можна буде отримувати повний список усіх надходжених пакетів із їх попередньою оцінкою.

Давайте тепер розглянемо, як буде виглядати звичайний мережвий трафік, який не мітсить ніяких загроз, тобто графік із клаифікацією шкідливих пакетів буде мати лише поле для довірениз, а отже саме такий вигляд мативе в більшості випадків, приклад зображено на рисунку 4.8.

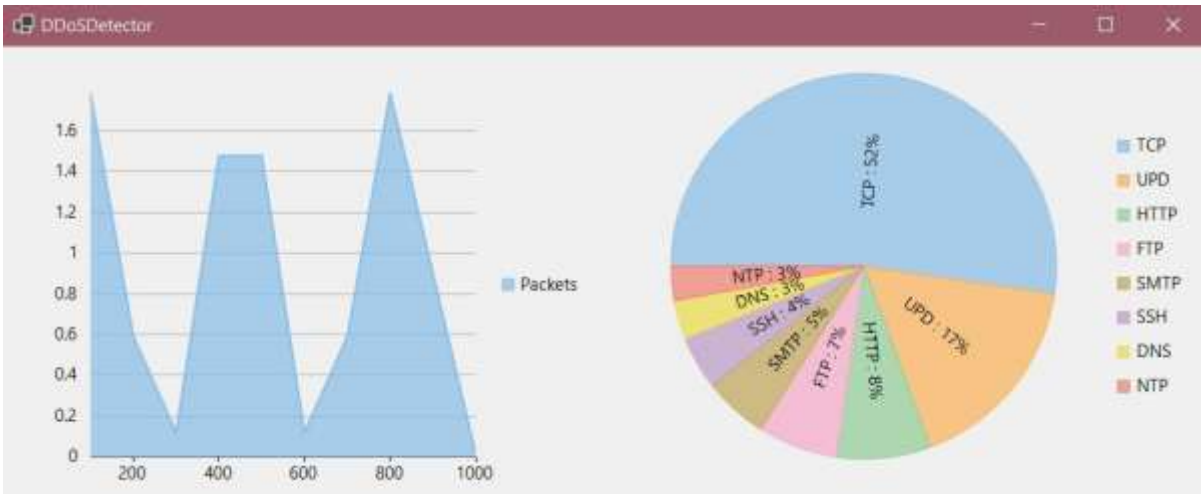


Рисунок 4.8 –Класифікатор звичайного мережевого трафіка.

4.4 Висновок до розділу

У четвертому розділі була детально розглянута і протестована створенна система, щоб провести тестування було створенно додаткове програмне забезпечення, яке імітувалу надходження трафіку і генерувало пакети, або випадково, або із тестових даних, щоб перевірити якість оцінки трафіку глибокою згортковою нейронною мережею.

Системи розроблялися на мові програмування C# і на технології WinForms, що дало можливість візуально відобразити результати роботи системи та дало можливість впровадити зручний користувацький інтерфейс.

За результатами аналізу мережевого трафіку можна зробити висновки, що натренована модель показала гарні результати і хороші оцінки, але важливим стоїть питання додаткового навчання моделі, оскільки вхідний потік даних може бути не настільки якісних, як передбачувалося, а отже за додатковим навчанням потрібно слідкувати, щоб модель не втрачала свою ефективність і продовжувала показувати ефективні результати.

Компоненти системи пройшли тестування на тестових наборах даних, для перевірки правильності роботи кожного із них і щоб переконатися, що усі функції та методи працюють правильно.

Дослідження показало, що аналіз мережевого трафіку з використанням глибоких згорткових нейронних мереж показує гарні результати при оцінці і правильному навчанні, а також має потенціал для додаткового навчання і покращення моделі.

ВИСНОВКИ

Кваліфікаційна робота магістра розв'язує задау аналізу мережевого трафіка з використанням глибоких згорткових нейрнонних мереж на предмет шкідливих пакетів та DDoS-атак. Для ефективної оцінки вхідних пакетів використовується завчасно навчена модель нейрнонної мережі, яка пройшла навчання на одному з найкращих запропонованих наборах даних по даній тематиці, щоб відобразити результати роботи системи, використовувалися метод візуальної аналітики.

У першому розділі проведено аналіз DDoS-атак, встановлено, що CNN є ефективним інструментом для виявлення шкідливих пакетів. Моделі CNN та LSTM виділяються серед існуючих методів виявлення DDoS-атак. Метод DDoS-Detector на базі глибоких згорткових мереж виявився високоефективним. Розділ сформулював мету дослідження - виявлення атак у мережі за допомогою глибокого аналізу трафіку та машинного навчання.

У другому розділі визначено, що отримання та підготовка даних є ключовим етапом для успішного використання згорткових нейронних мереж у виявленні шкідливих пакетів. Застосування архітектури CNN для аналізу мережевого трафіку передбачає використання кількох рівнів згорткових та повністю зв'язаних шарів для виявлення важливих характеристик. Підготовка даних перед навчанням моделі включає нормалізацію та перетворення інформації у числовий формат, що відіграє важливу роль у досягненні високої точності класифікації. Оцінка ефективності моделей на тестових даних дозволяє визначити їхню точність та здатність класифікувати мережевий трафік за допомогою різних метрик, таких як точність, відгук, специфічність та F1-мера.

Основний набір даних KDD Cup 99 відіграє важливу роль у виявленні складних загроз та аналізі різних аспектів мережевого трафіку. Використання методів машинного навчання з учителем та без учителя засноване на цьому наборі даних дозволяє розпізнавати аномалії в мережевому трафіку та класифікувати їх без прив'язки до конкретних видів атак. Оцінка моделей для виявлення DDoS-атак

використовує різні метрики для визначення точності та ефективності їх роботи, що дозволяє вибрати найкращий алгоритм для вирішення цієї проблеми в мережах.

У третьому розділі було реалізовано систему для аналізу мережевого трафіку, використовуючи глибокі згорткові нейронні мережі. Модель пройшла початкове навчання, а також отримала можливість поступового підвантаження нових даних та оновлення після оцінки їхньої ефективності, що робить її більш адаптивною до змін у мережевому трафіку.

Архітектура додатку була створена з урахуванням ефективного функціонування системи, надаючи можливість вибору мережевого адаптера та налаштування додаткового навчання моделі з використанням одних і тих же даних. Відмовившись від можливості додавання власного тестового набору даних, система дозволяє визначати ефективність додаткового навчання моделі без переоцінки даних тестового набору.

У четвертому розділі було впроваджено та протестовано систему, використовуючи додаткове програмне забезпечення для імітації трафіку та оцінки роботи системи. Аналіз мережевого трафіку показав, що натренована модель має великий потенціал, але потребує постійного навчання, особливо при обробці неякісних вхідних даних, щоб підтримувати ефективність та точність результатів.

Результати наукової роботи були написанні дві наукових публікацій, які були опубліковані на тему «Метод виявлення DDoS-атак на основі глибоких згорткових нейронних мереж» на XIX Міжнародна науково-практична конференція «Військова освіта і наука: сьогодення та майбутнє», також у доповіді на тему «Дослідження методів оцінки інформаційної безпеки програмного забезпечення» на IX Міжнародна науково-технічнка конференція «Захист інформації і безпека інформаційних систем». Ці дві публакції опубліковані під час роботи над кваліфікаційною роботою [55, 56].

ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАНЬ

1. Li, Q.; Meng, L.; Zhang, Y.; Yan, J. DDoS attacks detection using machine learning algorithms. In *International Forum on Digital TV and Wireless Multimedia Communications*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 205–216.

2. Mohammad Najafimehr, Sajjad Zarifzadeh, Seyedakbar Mostafavi (2023). DDoS attacks and machine-learning-based detection methods: A survey and taxonomy. [Доступ до ресурсу] - https://onlinelibrary.wiley.com/doi/full/10.1002/eng2.12697.

3. Nick Barney, Ben Lutkevich. Network Security. [Електроний ресурс] - https://www.techtarget.com/searchnetworking/definition/network-security.

4. What is network security? [Електроний ресурс] - https://www.cloudflare.com/learning/network-layer/network-security.

5. "DDoS, Machine Learning, Measures". // "Understanding Denial-of-Service Attacks". / , 2016. – (Taylor & Francis Group). – (ISBN:13: 978-1-4987-2965-9). – С. 12–34.

6. Peng T., Leckie C., and Ramamohanarao K. Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems // ACM Computing Surveys. – 2007. – Vol. 39, N 1. – P. 31 – 42.

7. "Network Security: Private Communication in a Public World" - Автор: Charlie Kaufman, Radia Perlman, Mike Speciner (2021).

8. M. Tayyab, B. Belaton, and M. Anbar, ‘‘ICMPv6-based DoS and DDoS attacks detection using machine learning techniques, open challenges, and blockchain applicability: A review,’’ IEEE Access, vol. 8, pp. 170529–170547, 2020.

9. Ahmed, Sheikh. (2021). A Study of ML Algorithms for DDoS Detection. International Journal for Research in Applied Science and Engineering Technology.

10. Das, Saikat & Mahfouz, Ahmed & Venugopal, Deepak & Shiva, Sajjan. (2019). DDoS Intrusion Detection through ML Ensemble.

11. Mirkovic J, Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Comput Commun Rev. 2004; 34(2): 39-53.

12. Bhattacharyya, D. K., & Kalita, J. K. (2016). DDoS attacks: Evolution, detection, prevention, reaction, and tolerance. Routledge.

13. Gupta, S., & Dahiya, R. (2023). Distributed denial-of-service (DDoS) attacks: Classification, attacks, challenges, and countermeasures. Routledge.

14. Masdari M, Jalali M. A survey and taxonomy of DoS attacks in cloud computing. Secur Commun Netw. 2016; 9(16): 3724-3751. doi:10.1002/sec.1539.

15. Pew Research Center. Artificial Intelligence and the Future of Humans. URL: https://www.pewresearch.org/internet/2018/12/10/artificial-intelligence-and-the-future-of-humans.

16. Freecodecamp. Deep Learning Neural Networks Explained in Plain Englishhttps://www.freecodecamp.org/news/deep-learning-neural-networks-explained-in-plain-english.

17. Investopedia. What Is a Neural Network? URL: https://www.investopedia.com/terms/n/neuralnetwork.asp.

18. Ethem Alpaydin (2020). Introduction to Machine Learning (англ.) (вид. Fourth). MIT. с. xix, 1–3, 13–18. ISBN 978-0262043793.

19. Zainudin A , Ahakonye LAC , Akter R , Kim DS , Lee JM . Ефективний гібридний DNN для виявлення та класифікації DDoS у програмно визначених мережах IIoT . IEEE Internet Things J. 2023 ; 10(10):8491-8504 . doi: 10.1109/JIOT.2022.3196942.

20. Roopak M , Tian GY , Chambers J. Моделі глибокого навчання для кібербезпеки в мережах IoT. Доповідь, представлена на: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), IEEE . 2019 : 0452-0457.

21. Ali, K.. Algorizmi: A Configurable Virtual Testbed to Generate Datasets for O ine Evaluation of Intrusion Detection Systems. Ph.D. thesis – 2010.

22. Bhuyan, Monowar H. et al. “Towards Generating Real-life Datasets for Network Intrusion Detection.” I. J. Network Security 17: 683-701. – 2015.

23. Shiravi, A., Shiravi, H., Tavallaee, M. and Ghorbani, A.A. Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection. Computers & Security, 2012, 357-374.

24. DARPA Intrusion Detection Data Sets [Electronic resource] -Access mode: https://www.ll.mit.edu/ideval/data/.

25. KDD Cup 1999 Data [Electronic resource] -Access mode: http://kdd.ics.uci.edu/databases/kddcup99.

26. Kayacik, H. G. Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets / H. G. Kayacik, A. N. Zincir-Heywood, M. I. Heywood // Proceedings of the Third Annual Conference on Privacy, Security and Trust (PST-2005) —2006. —P. 85–89.

27. Han J, Pei J, Kamber M. Data Mining: Concepts and Techniques. 3rd ed. Elsevier; 2012.

28. Alpaydin E. Introduction to Machine Learning. 2nd ed. The MIT Press; 2010.

29. Rahman O, Quraishi MAG, Lung C. DDoS attacks detection and mitigation in SDN using machine learning. Paper presented at: 2019 IEEE World Congress on Services (SERVICES). 2019:184-189.

30. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Paper presented at: 4th International Conference on Information Systems Security and Privacy. 2018:108-116.

31. Chen J, Yang YT, Hu KK, Zheng HB, Wang Z. DAD-MCNN: DDoS attack detection via multi-channel CNN. Paper presented at: ICMLC'19. Association for Computing Machinery, New York, NY, USA. 2019:484-488.

32. Roopak M, Tian GY, Chambers J. Deep learning models for cyber security in IoT networks. Paper presented at: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), IEEE. 2019:0452-0457.

33. Zainudin A, Ahakonye LAC, Akter R, Kim D-S, Lee J-M. An efficient hybrid-DNN for DDoS detection and classification in software-defined IIoT networks. IEEE Internet Things J. 2023; 10(10):8491-8504. doi:10.1109/JIOT.2022.3196942.

34. Ashraf, J.; Moustafa, N.; Bukhshi, A.D.; Javed, A. Intrusion Detection System for SDN-enabled IoT Networks using Machine Learning Techniques. In Proceedings of the 2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW), Gold Coast, Australia, 25–29 October 2021.

35. Dora, V.R.S.; Lakshmi, V.N. Optimal feature selection with CNN-feature learning for DDoS attack detection using meta-heuristic-based LSTM. Int. J. Intell. Robot. Appl. 2022, 6, 323–349.

36. M. Zekri, S. El Kafhali, N. Aboutabit, and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," in 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), pp. 1-7, 2017.

37. Makori, D.O. Machine Learning Based Ddos Attack Detection for Software-Defined Networks: Yazılım Tanımlı Ağlar Için Makine Öğrenme Esaslı Ddos Attack Algılama. M.Sc. Thesis, Akarya Üniversitesi, Sakarya, Turkey, 2018.

38. Detection and recovery against deep neural network fault injection attacks based on contrastive learning / Wang C. [et al.] // Proceedings of the 3rd Workshop on Adversarial Learning Methods for Machine Learning and Data Mining at KDD, Singapore, 14 Aug. 2021. – 2021.

39. Zhang L. Self-Distillation: Towards Efficient and Compact Neural Networks / Linfeng Zhang, Chenglong Bao, Kaisheng Ma // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2021. – Vol. 44, no. 8. – P. 4388-4403. – DOI: https://doi.org/10.1109/tpami.2021.3067100.

40. Olowononi F. O. Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS / F. O. Olowononi, D. B. Rawat, C. Liu // IEEE Communications Surveys & 109 Tutorials. – 2020. – Vol. 23, no. 1. – DOI: https://doi.org/10.1109/comst.2020.3036778.

41. Hospedales T. Meta-Learning in Neural Networks: A Survey. / T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2021. – DOI: https://doi.org/10.1109/TPAMI.2021.3079209.

42. A survey on missing data in machine learning / Tlamelo Emmanuel [et al.] // Journal of Big Data. – 2021. – Vol. 8, no. 1. – DOI: https://doi.org/10.1186/s40537- 021-00516-9.

43. A taxonomy and survey of attacks against machine learning / Nikolaos Pitropakis [et al.] // Computer Science Review. – 2019. – Vol. 34. – DOI: https://doi.org/10.1016/j.cosrev.2019.100199.

44. Doke A. Survey on Automated Machine Learning (AutoML) and Meta learning / Ashwini Doke, Madhava Gaikwad // 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021. –2021. – DOI: https://doi.org/10.1109/icccnt51525.2021.9579526.

45. Gong Z. Diversity in Machine Learning / Zhiqiang Gong, Ping Zhong, Weidong Hu // IEEE Access. – 2019. – Vol. 7. – DOI: https://doi.org/10.1109/access.2019.2917620.

46. Bengio Y. Representation Learning: A Review and New Perspectives / Y. Bengio, A. Courville, P. Vincent // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2013. – Vol. 35, no. 8. – DOI: https://doi.org/10.1109/tpami.2013.50.

47. Scalable Quantitative Verification for Deep Neural Networks / Teodora Baluta [et al.] // 2021 IEEE/ACM 43rd International Conference on Software 116 Engineering: Companion Proceedings (ICSE-Companion), Madrid, ES, 25–28 May 2021. – 2021. – DOI: https://doi.org/10.1109/icse-companion52605.2021.00115.

48. Doon R. Cifar-10 Classification using Deep Convolutional Neural Network / Raveen Doon, Tarun Kumar Rawat, Shweta Gautam // 2018 IEEE Punecon, Pune, India, 30 Nov. – 2 Dec. 2018. – 2018. – P. 1-5. – DOI: https://doi.org/10.1109/punecon.2018.8745428.

49. Detection and recovery against deep neural network fault injection attacks based on contrastive learning / Wang C. [et al.] // Proceedings of the 3rd Workshop on Adversarial Learning Methods for Machine Learning and Data Mining at KDD, Singapore, 14 Aug. 2021. – 2021.

50. Valtchev S. Z. Domain randomization for neural network classification / Svetozar Zarko Valtchev, Jianhong Wu // Journal of Big Data. – 2021. – Vol. 8, no. 1. – DOI: https://doi.org/10.1186/s40537-021-00455-5.

51. Hoang L.-H. TRe-Map: Towards Reducing the Overheads of Fault-Aware Retraining of Deep Neural Networks by Merging Fault Maps / Le-Ha Hoang, Muhammad Abdullah Hanif, Muhammad Shafique // 2021 24th Euromicro Conference on Digital System Design (DSD), Palermo, Italy, 1–3 September 2021. – 2021. – DOI: https://doi.org/10.1109/dsd53832.2021.00072.

52. Huang K. Functional Error Correction for Robust Neural Networks / Kunping Huang, Paul H. Siegel, Anxiao Jiang // IEEE Journal on Selected Areas in Information Theory. – 2020. – Vol. 1, no. 1. – P. 267–276. – DOI: https://doi.org/10.1109/jsait.2020.2991430.

53. СОУ 207.01:2017. Текстові документи. Загальні вимоги. Хмельницький: ХНУ, 2017. – 46с. URL: http://surl.li/mpjra (дата звернення 20.09.2023).

54. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. [Чинний від 2016-07-1]. Вид. офіц. Київ: Державна наукова установа “Книжкова палата України імені Івана Федорова”, 2016. – 20 с.

55. Майор Є.В., Джулій В.М. Метод виявлення DDoS-атак на основі глибоких згорткових нейронних мереж. XIX Міжнародна науково-практична конференція "Військова освіта і наука: сьогодення та майбутнє" 10 листопада 2023. С. 35-36.

56. Майор Є.В., Джулій В.М., Мостовий С.В., Чешун В.М., Дослідження методів оцінки інформаційної безпеки програмного забезпечення. IX Міжнародна науково-технічнка конференція "Захист інформації і безпека інформаційних систем" 25-26 травня, 2023. С. 135-1

Додаток А

Світлини наукових публікацій, виконаних при роботі над кваліфікаційною роботою магістра

(ксерокопії титульної сторінки, сторінки змісту та всіх сторінок із публікацією)

Перелік наукових публікацій:

1. Майор Є.В., Джулій В.М. МЕТОДИ ВИЯВЛЕННЯ DDoS-АТАК НА ОСНОВІ ГЛИБОКИХ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ. Збірник тез доповідей XIX Міжнародної науково-практичної конференції «Військова освіта і наука: сьогодення та майбутнє». Київ, 2023. с. 35-36.
2. Євген МАЙОР, Володимир ДЖУЛІЙ, Віктор ЧЕШУН, Сергій МОСТОВИЙ ДОСЛІДЖЕННЯ МЕТОДІВ ОЦІНКИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. Матеріали IX Міжнародної науково-технічної конференції «Захист інформації і безпека інформаційних систем». Львів, 2023. с. 135-136.
3. Алгоритми прогнозування вразливостей та загроз інформаційної безпеки на основі тематичних інтернет-ресурсів./ Майор Є., Джулій В., Чешун В., Петляк Н. Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах». 2023. Випуск 4. С.49-56.

ЗМІСТ

ВІЙСЬКОВИЙ ІНСТИТУТ
КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

XIX Міжнародної науково-практичної конференції

«Військова освіта і наука: сьогодення та майбутнє»

10 листопада 2023 року

Київ – 2023

До методу безпеки даних від витоку даних, пред'являються наступні вимоги: використання статистичних методів, незалежних від характеристик контейнерів передачі та зберігання даних; класифікація лише підозрілих послідовностей; формат аналізованих даних не важливий, дані надходять в бінарному вигляді; точність класифікації стислих та зашифрованих даних має досягати максимального значення; можливість протидії загрозам безпеки корпоративних мереж підприємства, також протидія та виявлення botnet мережам; час виконання класифікації має досягати до мінімуму.

Здійснено формальну постановку задачі, визначено мету, проведено аналіз об'єкта та предмета дослідження. Відсоток інцидентів порушення безпеки, конфіденційних даних, пов'язаних із витоком інформації, причиною яких є внутрішні зловмисники, склав понад 78%, що підтверджує актуальність дослідження. Проведено аналіз вразливостей та загроз DLP–систем та засобів захисту даних, визначено недоліки та переваги використовуваних підходів класифікації стислих та зашифрованих даних.

Обгрунтовано вибір статистичних методів проведення аналізу переданих даних для побудови класифікатора, сформульована наукова задача магістерського дослідження у формальному виді. Показано практичну проблему, яка полягає в низькій точності класифікації стислих і зашифрованих псевдовипадкових послідовностей.

к.т.н., доц. Джулій В.М. (ХмНУ)
Майор Є.В. (ХмНУ)

МЕТОДИ ВИЯВЛЕННЯ DDoS-АТАК НА ОСНОВІ ГЛИБОКИХ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Існує декілька методів виявлення DDoS-атак на основі технології глибокого навчання, з використанням згорткових нейронних мереж (CNN).

CNN мають численні переваги порівняно із традиційними методами виявлення DDoS-атак. Вони відзначаються вищою точністю та ефективністю, що робить їх більш привабливими для використання в області кібербезпеки. CNN можуть бути навчені розпізнавати нові, раніше невідомі типи DDoS-атак, що дозволяє забезпечити більш широкий спектр захисту в мережевих системах. Переваги використання CNN для виявлення DDoS-атак: точність; ефективність; адаптивність - CNN можна навчити розпізнавати нові типи DDoS-атак.

DDoS-Detector - метод виявлення DDoS-атак на основі CNN. Використовує CNN для класифікації трафіку DDoS, не-DDoS. На етапі навчання CNN навчається на наборі даних, що містить оброблені дані. CNN може бути навчена за допомогою різних алгоритмів навчання, таких як backpropagation. На етапі класифікації трафіку CNN використовується для класифікації трафіку. CNN видає оцінку чи є трафік DDoS-атакою. Оцінка може бути використана для прийняття рішення про те, чи потрібно блокувати трафік. Метод DDoS-Detector протестований на різних наборах даних і показав високу точність і є одним із найбільш ефективних методів виявлення DDoS-атак на основі CNN .

CNN-DDoS - метод виявлення DDoS-атак на основі CNN, який використовує CNN для виявлення аномалій у трафіку, які можуть бути ознаками DDoS-атаки. Відмініть від DDoS-Detector полягає у фінальному етапі, метод

виконує виявлення аномалій, де CNN використовується для виявлення аномалій у трафіку. CNN-DDoS протестований на різних наборах даних і показав високу точність. Метод CNN-DDoS є одним із найбільш ефективних методів виявлення DDoS-атак на основі CNN.

DeepDDoS - система виявлення DDoS-атак, яка використовує глибокі нейронні мережі для аналізу мережевого трафіку та виявлення атак. Цей підхід базується на використанні CNN для виявлення аномалій у мережевому трафіку, що може бути спричинене DDoS-атаками.

Проведено аналіз методів виявлення DDoS-атак на основі глибоких згорткових нейронних мереж та методів машинного навчання. Задача полягає у вирішенні проблеми виявлення шкідливих пакетів та DDoS-атак у мережі шляхом проведення аналізу мережевого трафіку мереж.

Жидков Д.В. (НЮУ)

АКТУАЛЬНІ ПРОБЛЕМИ АВТОМАТИЗАЦІЇ БПЛА З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Останніми роками широко почали впроваджуватися технології штучного інтелекту (далі-ШІ) у військову сферу. Повна автоматизація механічних процесів призводить до економії ресурсів — соціальних , економічних, адміністративних, інформаційних. ШІ почали широко використовувати в безпілотних літальних апаратах, що являє собою значний прорив у сфері військових технологій. І такий прорив характеризується наступним: по-перше, БПЛА зі штучним інтелектом можна використовувати для спостереження та моніторингу і самостійного визначення об'єктів ураження; по-друге, БПЛА із ШІ можна використовувати для збору та аналізу великого обсягу даних. Ці дані можна отримати з різних джерел (камери, датчики), а використати їх можна, наприклад, для оцінки кількості особового складу та техніки супротивника на визначеній території. Аналізуючи отримані дані, ШІ здатен швидко приймати рішення та скеровувати їх на певні дії БПЛА; по-третє, поєднання ШІ з БПЛА дозволило підсилити безпеку та надійність. Також він здатен коригувати рух маршруту, безпечно сідати у надзвичайних ситуаціях, що значно убезпечить БПЛА від руйнування. Існують декілька типів ШІ, які зараз застосовують в БПЛА. Перший тип – комп'ютерне бачення, воно полягає у використанні камери та датчиків для виявлення об'єктів і перешкод у навколишньому середовищі. Його можна використовувати для виявлення та уникнення перешкод, а також для визначення об'єктів і орієнтирів. Комп'ютерний зір також можна використовувати для відстеження об'єктів (людей, військову техніку). Другий тип — це обробка природної мови (NLP). Цей тип ШІ використовується для інтерпретації та розуміння голосових команд, він дозволяє застосовувати голосові команди з подальшим їх обробленням БПЛА та миттєвим пошуком необхідної відповіді. Третій тип — машинне навчання. Останній тип ШІ є найскладнішим, адже він побудований на алгоритмах самонавчання та самоаналізу ШІ, що само собою становить інтерес серед науковців, адже таке навчання виявляє закономірності в навколишньому середовищі та шукає найраціональніші способи для досягнення поставлених цілей. І, нарешті, четвертий тип—глибоке навчання, яке полягає в аналізі великих обсягів даних і прийняття рішень на основі цих даних. Саме

Міністерство освіти і науки України
Національна Академія наук України
Міністерство науки та вищої освіти Республіки Польща
Національний університет "Львівська політехніка"
Інститут прикладних проблем механіки і математики ім. Я. С. Підстригача НАН України
Одеський національний політехнічний університет
Університет Бєльсько-Бяла (Польща)

ЗАХИСТ ІНФОРМАЦІЇ І БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ

МАТЕРІАЛИ
IX Міжнародної
науково-технічної конференції

25–26 травня, 2023

Львів
Видавництво Львівської політехніки
2023

INFORMATION PROTECTION AND INFORMATION SYSTEMS SECURITY

MATERIALS
of IXth International Scientific and Technical Conference

May 25–26, 2023

ДОСЛІДЖЕННЯ МЕТОДІВ ОЦІНКИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Володимир ДЖУЛІЙ[a], Віктор ЧЕШУН[a], Сергій МОСТОВИЙ[a] and Євген МАЙОР[a]

[a]*Хмельницький національний університет, вул. Інститутська,11, Хмельницький, Україна*

Анотація. *На даний час не існує універсального вирішення задачі виявлення шкідливого програмного забезпечення в умовах обмежених часових та обчислювальних ресурсів, а також відсутності відкритих вихідних текстів. Для підвищення ефективності вирішення даної задачі пропонується підхід, що полягає в автоматизації оцінки інформаційної безпеки та базується на моделі безпечного функціонування програмного забезпечення.*

Ключові слова. *Кібератака, шкідливе програмне забезпечення, статичний аналіз, динамічний аналіз, інтерактивне тестування.*

Вступ

Число кібератак на найважливіші галузі інфраструктури у всьому світі з кожним роком неухильно зростає. Так, за даними компанії Mitnick Security [1], вже в першому кварталі 2022 року було зафіксовано на 23,5% більше кібератак, ніж у останньому кварталі 2021 року.

При реалізації кібератак зловмисники комбінують різні типи шкідливого програмного забезпечення (ШПЗ) – використовують багатофункціональні трояни, завантажують на об'єкти інфраструктури безліч різних за функціоналом видів ШПЗ. Особливої гостроти проблема набуває у зв'язку з об'єктивною необхідністю організації віддаленої роботи та доступу до інформаційних ресурсів, що потребує вирішення задачі захисту «територіально розподіленого периметру».

Виявити потенційні загрози можна на підставі аналізу вихідних текстів та динамічного аналізу поведінки у контрольованому середовищі, але в умовах відсутності вихідних текстів потрібне використання більш технічно складних процедур. На практиці фахівцю із захисту інформації потрібно швидко і ефективно визначити можливість використання програмного забезпечення (ПЗ) без порушення інформаційної безпеки, коли час та обчислювальні ресурси обмежені, а повна автоматизація процесу неможлива у зв'язку з високою складністю предметної області. На даний час не існує оптимального вирішення задачі виявлення ШПЗ. Використання всього спектру наявних засобів дозволяє максимально наблизитися до вирішення задачі виявлення ШПЗ, проте вимагає кваліфікації, докладання значних зусиль і часових витрат.

Дослідження методів оцінки інформаційної безпеки програмного забезпечення

Одним із найпоширеніших методів реалізації кібератак є використання ШПЗ. На початку 2022 року кількість кібератак, реалізованих цим методом, незначно поступилася лише кібератакам методами соціальної інженерії. Частка кібератак з використанням ШПЗ, за даними компанії Mitnick Security, склала 73% для фізичних осіб та 62% для юридичних осіб [1].

Класифікація і основні властивості методів

Проведене дослідження дозволило виділити три ключові методи та підходи до оцінки інформаційної безпеки ПЗ без вихідних текстів.

1. Статичний аналіз бінарного коду програм. Перевагою статичного аналізу, на відміну від методів динамічного аналізу, є повне покриття коду [2,3]. Виконуючи бінарний аналіз [3] власного ПЗ, можна виконати перевірку сторонніх бібліотек, які були використані при розробці. Статичний аналіз може бути успішно задіяний при дослідженні стороннього ПЗ.
2. Динамічний аналіз із використанням «пісочниці». Це метод представляє два підходи до виявлення підозрілої поведінки у віртуальному ізольованому середовищі [4].
3. Інтерактивне тестування [3] безпеки додатків є методикою аналізу безпеки ПЗ, переважно спрямоване на дослідження поведінки веб-додатків під час їх роботи. Деякі рішення можуть не тільки активно відстежувати вразливості (SQL-ін'єкції), але й перевіряти їх, демонструючи їхню придатність для використання зловмисниками. Для аналізу безпеки часто використовуються методи форензики (методи, пов'язані з розслідуванням вже існуючих комп'ютерних інцидентів).

Перший підхід полягає у віртуалізації єдиного загального середовища виконання ПЗ, це середовище потім використовується при аналізі всіх зразків ПЗ. Такий підхід не позбавлений недоліків: можливий пропуск зразків ШПЗ, функціонування яких залежить від певного набору ресурсів або параметрів конфігурації.

У другому випадку віртуалізується декілька середовищ, кожне з яких імітує певну платформу

та конфігурацію обчислювальної системи. Досліджуване ПЗ запускається в кожному такому середовищі, що знижує помилки другого роду, проте при цьому збільшує помилки першого роду. Недоліком є обсяг використовуваних ресурсів – він значно більший, ніж у разі першого підходу. Оптимізований підхід – запуск зразка ПЗ виконується в такому віртуальному середовищі, яке повністю відповідає цільовій системі. У такому разі «пісочниця» повинна «на льоту» визначати цільове середовище вузла мережі та запускати необхідну віртуальну машину.

При оцінці безпеки ПЗ без вихідних текстів, найкращий результат досягається при їх сукупному статичного та динамічного методів, оскільки статичний аналіз дозволяє отримати повне покриття коду та усунути ключовий недолік, властивий динамічному аналізу.

Оцінка ресурсів, необхідних для реалізації досліджених методів

Оцінка ресурсів необхідних для практичного використання досліджених методів значно варіюється залежно від типу системи, тому можлива лише наближена оцінка, побудована у взаємному порівнянні методів. Для отримання порівняльної оцінки використано шестибальну шкалу. Крім використання ресурсів даними методами розглядаються такі параметри: покриття коду; швидкість аналізу; можливість виявлення складних уразливостей та проблем безпеки; обчислювальні ресурси. Отримані результати представлені у таблиці 1.

Дослідження показали, що застосування засобів віртуалізації при динамічному аналізі повністю виправдане, хоч і потребує досить великого обсягу ресурсів. Також важливо відзначити, що найбільш повне покриття коду досягається саме при статичному аналізі.

Таблиця 1

Порівняння параметрів методів аналізу безпеки без вихідних текстів

	Статичний аналіз	Динамічний аналіз	Інтерактивне тестування
Покриття коду	5	3	3
Швидкість аналізу	4	4	3
Виявлення складних вразливостей	3	3	2
Обчислювальні ресурси	3	3	4

Висновки

Дослідження науково-технічних джерел дозволило дійти висновку, що найбільш популярною є техніка динамічного аналізу ПЗ, при чому особливо ефективною вона стає в поєднанні з механізмами віртуалізації, оскільки такий симбіоз надає можливість дослідити поведінку зразка ПЗ у середовищі, наближеному за своїми характеристиками до цільового. Для підвищення ефективності виявлення ШПЗ досягнення поєднанням статичного і динамічного аналізу, оскільки статичний аналіз забезпечує максимальне покриття коду. Однак, в умовах відсутності вихідних текстів ПЗ це зробити важче, ніж за їх наявності, оскільки використання дизассемблерів дозволяє лише приблизно відновити високорівневий код програмного забезпечення.

Список літератури

[1] Common Hacking Techniques in 2022 and Predictions for 2023. *Cyber Security Articles & News*. Posted by Mitnick Security on Sep 14, 2022. – URL: https://www.mitnicksecurity.com/blog/common-hacking-techniques-2022

[2] IoTSIT: A Static Instrumentation Tool for IoT Devices / C. Chen, Z. Jing, J. Ma, B. Cui, H. Xu, Q. Zou. *IEEE Access*. 2020. №.8. P. 92153-92161.

[3] Kargen U. Scalable Dynami: Analysis of Binary Code. Linköping Studies in Science and Technology Dissertations. 2019. 86 p.

[4] Le H. V., Ngo Q. D. V-Sandbox for Dynamic Analysis IoT Botnet. IEEE Access. 2020. № 8. P. 145768-145786.

ISSN 2219-9365

https://doi.org/10.31891/2219-9365-2023-76-6
УДК 004.056:621.397.3:004.942

МАЙОР Євген
Хмельницький національний університет
https://orcid.org/0009-0004-1867-6241
e-mail: gorix2019@gmail.com
ДЖУЛІЙ Володимир
Хмельницький національний університет
https://orcid.org/0000-0003-1878-4301
e-mail: dzhuliivm@khmnu.edu.ua
ЧЕШУН Віктор
Хмельницький національний університет
https://orcid.org/0000-0002-3935-2068
e-mail: cheshunvn@khmnu.edu.ua
ПЕТЛЯК Наталія
Хмельницький національний університет
https://orcid.org/0000-0001-5971-4428
e-mail: npetlyak@khmnu.edu.ua

АЛГОРИТМИ ПРОГНОЗУВАННЯ ВРАЗЛИВОСТЕЙ ТА ЗАГРОЗ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ НА ОСНОВІ ТЕМАТИЧНИХ ІНТЕРНЕТ-РЕСУРСІВ

Розглянуті особливості функціонування форумів тематичних інтернет-ресурсів дозволяють здійснювати прогнозування виникнення вразливостей та загроз безпеки конфіденційних даних. Для вирішення даної задачі передбачається проведення аналізу інтернет-повідомлень, створюваних учасниками форумів тематичних інтернет-ресурсів, відповідно до представлених алгоритмів. Алгоритм прогнозування вразливостей та загроз безпеки інформації відрізняється можливістю виявлення вразливостей та загроз на ранніх етапах їх практичної реалізації, ґрунтується на проведенні аналізу потоку повідомлень форумів тематичних інтернет-ресурсів, що дозволяє спеціалістам з інформаційної безпеки приймати адекватні та своєчасні заходи щодо захисту конфіденційних даних.

Алгоритм фільтрації потоку повідомлень та статистичного аналізу передбачає фільтрацію тематичних повідомлень, що не відносяться до заданої предметної області, яка задана відповідною онтологією. Запропоновані алгоритми дозволяють прогнозувати вразливості та загрози, вживати адекватних заходів щодо захисту інформації.

Ключові слова: інформаційна безпека, алгоритм прогнозування вразливостей, алгоритм фільтрації повідомлень , тематичні інтернет-ресурси.

MAIOR Yevhen, DZHULIY Volodymyr, CHESHUN Viktor, PETLIAK Nataliia
Khmelnytskyi National University

ALGORITHMS FOR PREDICTING INFORMATION SECURITY VULNERABILITIES AND THREATS BASED ON THEMATIC INTERNET RESOURCES

The considered features of the functioning of forums of thematic Internet resources allow to forecast the occurrence of vulnerabilities and threats to the security of confidential data. To solve this problem, the analysis of Internet messages created by participants of forums of thematic Internet resources is expected, according to the presented algorithms.

The algorithm for predicting vulnerabilities and threats to information security is distinguished by the possibility of detecting vulnerabilities and threats at the early stages of their practical implementation, is based on the analysis of the flow of messages of forums of thematic Internet resources, which allows information security specialists to take adequate and timely measures to protect confidential data. The result of the algorithm's work is reports on identified vulnerabilities and threats to the information security of confidential data. The reports may also include information reflecting the results of text message analysis, based on which a conclusion about the occurrence of vulnerabilities and threats was obtained.

The message flow filtering and statistical analysis algorithm involves filtering thematic messages that do not belong to the given subject area, which is specified by the corresponding ontology. The result of the algorithm is the determination of statistical indicators characterizing the flow of thematic messages during the period of data flow analysis. The algorithm also calculates the number of text messages that have passed the data flow filtering stage and determines the average rating of text message authors. The results of the algorithm can be used to build a system of logical fuzzy inference for forecasting the events of the subject area for which the analysis is conducted.

The proposed algorithms allow predicting vulnerabilities and threats, taking adequate measures to protect information. The obtained results testify to the effectiveness of the proposed algorithms for forecasting vulnerabilities and threats, and also confirm the correctness of the information and analytical system.

Keywords: information security, vulnerability prediction algorithm, message filtering algorithm, thematic Internet resources.

Постановка проблеми у загальному вигляді
та її зв'язок із важливими науковими чи практичними завданнями

Актуальними та пріоритетними на сучасному етапі є задачі аналізу і класифікації існуючих механізмів реалізації атак та загроз інформаційної безпеки, які можуть призвести до отримання несанкціонованого доступу до конфіденційної інформації та порушення функціонування інформаційних систем. Постає задача визначення заходів протидії атакам та загрозам, усунення вразливостей, оцінки можливості задання шкоди, підготовки нормативно-правової бази, механізмів захисту та критеріїв безпеки.

Проблеми інформаційної безпеки розвитку суспільства у більшості сфер діяльності виходять на перший план. Це пов'язано зі значним зростанням кількості реалізованих проектів інформатизації, більшість з яких спрямовано на побудову єдиного телекомунікаційного та інформаційного простору з метою оптимізації процесів обробки різноманітної інформації великих об'ємів, наприклад, для забезпечення оперативного доступу до інформації, надійного зберігання даних для користувачів інформаційного обміну

Важливість даних проблем пов'язана з наступними основними факторами [1-3]: зростанням різноманітності та кількості засобів комп'ютерної техніки та сфер людської діяльності їх застосування; високим рівнем довіри до інформаційно-пошукових систем обробки та управління даними; зростанням числа користувачів інформаційного простору взаємодії; накопиченням великих об'ємів різнотипної інформації; інтенсивним обміном потоком даних в мережі між користувачами; використання широкого спектра механізмів доступу до конфіденційних ресурсів, інформаційних процесів; промисловим шпигунством та конкурентною боротьбою у сфері інформаційних послуг суспільства; недостатньою кількістю фахівців високої кваліфікації в сфері інформаційної безпеки; ринковими відношеннями в галузі розробки програмного забезпечення, обслуговування, розповсюдження, виробництва обчислювальної комп'ютерної техніки для реалізації інформаційної безпеки; різноманіттям атак, загроз і різнотипних каналів отримання несанкціонованого доступу до конфіденційних ресурсів та диференціацією негативних наслідків.

Більшість існуючих моделей безпеки інформації, на сучасному етапі, ґрунтуються на забезпеченні конфіденційності, доступності, цілісності задіяної інформації [9]. Вразливості мережевих інформаційних систем, як правило, є наслідком внесених в систему помилок. Помилки, що є причиною формування вразливостей, в свою чергу, поділяються на помилки реалізації та помилки адміністрування [2,6].

Постановка задачі

На сьогодні не існує єдиного підходу до вирішення проблеми захищенності інформаційно-пошукових систем стосовно предметних областей [1,2]: забезпечення надійного захисту інформаційних ресурсів потребує реалізації технічних та організаційних заходів в комплексі, що супроводжуються розробкою відповідної документації, а розробниками програмно-апаратних засобів захисту інформації пропонуються відповідні компоненти на вирішення конкретних задач.

Таким чином, є необхідність вирішення наступних задач для забезпечення інформаційної безпеки [4,7,8]: формування основ для опису процесів реалізації та виникнення атак, загроз, вразливостей інформаційної безпеки системи в умовах невизначеності та непередбачуваності їх прояву; розробка відповідних засобів забезпечення захисту конфіденційної інформації на основі проведеного дослідження та класифікації вразливостей і загроз; визначення загальних підходів до створення інформаційних систем забезпечення захисту конфіденційних даних, механізмів управління захистом на різних рівнях діяльності суспільства.

Більшість сучасних програмно-апаратних систем виявлення комп'ютерних загроз та атак працюють із використанням підходів сигнатурного аналізу та фіксації інтернет-мережевих аномалій. Дані підходи мають недоліки, пов'язані із використанням потужних обчислювальних ресурсів на їх реалізацію, а також мають низьку ефективність при виявленні нових комп'ютерних загроз [8].

Основними джерелами надходження знань про вразливості та атаки інформаційної безпеки є бази даних та знань, створювані державними та комерційними структурами. Наповнення інформаційних баз даних здійснюється із залученням дослідних авторитетних центрів експертним шляхом. Разом з тим, інформація, що міститься в базах даних та знань вразливостей та загроз не є повною. Актуальною залишається задача виявлення доступних інформаційних ресурсів про комп'ютерні загрози, віруси, вразливості, а також можливість доступу до результатів досліджень компаній з виявлення загроз інформаційної безпеки. Одним із джерел надходження інформації про вразливості та загрози інформаційної безпеки є інтернет-ресурси (інформаційні соціальні ресурси, анонімні тощо, які відносяться до сфкри інформаційної безпеки), що обумовлено популярністю спеціалізованих інтернет-ресурсів у зацікавлених відповідними предметними областями. Події, що відбуваються в відповідних предметних областях, є предметом для обговорення учасників дискусійних тематичних інтернет-майданчиків. Даний фактор дозволяє прогнозувати виявлення вразливостей, атак, загроз безпеки інформації, ґрунтуючись на проведенні аналізу потоку повідомлень тематичних інтернет-ресурсів. Як один із підходів вирішення задачі розглянуто можливість використання інформаційних систем нечіткого логічного виводу, вхідними даними яких є результати проведеного аналізу інформації тематичних інтернет-ресурсів. Фахівець безпеки інформації

зможе оцінити ступінь інформаційної небезпеки ресурсів на основі отриманих результатів прогнозування виникнення вразливості, атаки, загрози, оцінити коректність моделі загроз безпеці інформації та задіяти протидію щодо нейтралізації вразливостей.

В результаті аналізу виявлено невирішенні питаня стосовно автоматизації інформаційних процесів прогнозування вразливостей та загроз безпеки інформації. Актуальною постає задача проектування та розробки методу і системи прогнозування, виявлення вразливостей, загроз безпеки інформації.

Вирішення поставлених задач спрямоване на підвищення якості прийнятих рішень у процесі виявлення та протидії шкідливій інформації; сортування інформаційна об'єктв впливу для оператора за пріоритетом; задання вхідних даних налаштування системи виявлення та протидії поширенню шкідливої інформації в мережах.

Основна частина

На теперішній час в мережі Інтернет функціонує велика кількість інтернет-майданчиків та форумів (спеціалізованих інформаційних ресурсів), які використовуються учасниками мережі для обговорення механізмів та способів несанкціонованого доступу до конфіденційних даних, а також забезпечення безпеки інформації. Частина зареєстрованих користувачів цікавляться відомостями про захист та безпеку інформації, інші – способами здійснення атак на інформаційно-комунікаційні системи і мережі. Форуми можуть розглядатися як джерела інформації про вразливості, шкідливе програмне забезпечення, комп'ютерні атаки тощо.

На інтернет-ресурсах переважна більшість тем обговорення присвячено висвітленню наступних питань: програмування з метою реалізації вразливостей та загроз безпеки інформації; програмне забезпечення, що використовується для організації та проведення комп'ютерних атак; шахрайство з використанням сучасних інформаційних технологій; поширення та створення шкідливого програмного забезпечення; забезпечення сеансу анонімності при здійсненні протиправних дій із застосуванням сучасних інформаційних технологій; переведення в готівку викрадених коштів, протиправні операції з банківськими картками; захист інформації. Перераховані теми відповідають актуальним загрозам безпеки конфіденційним даним [6,8,12], що надає можливість розглядати тематичні інтернет-ресурси як джерела повідомлень для проведення аналізу та виявлення вразливостей і загроз. Події, що відбуваються у конкретній предметній області, знаходять свій відбиток на відповідних дискусійних інтернет-майданчиках. Серед тематичних учасників інтернет-ресурсів присутні учасники, які володіють відомостями про вразливості та загрози безпеці інформації, а також потенційні зловмисники, зацікавлені в подоланні механізмів та засобів захисту конфіденційних даних.

Зазначені фактори надають можливість прогнозувати вразливості та загрози інформаційної безпеки даних, ґрунтуючись на проведеному аналізі повідомлень тематичних інтернет-ресурсів, використовуючи, при цьому, закономірності, характерні для процесу обговорення вразливостей та загроз. В загальному вигляді процес аналізу тематичних інтернет-ресурсів та їх інформаційного наповнення наведено на рис. 1.

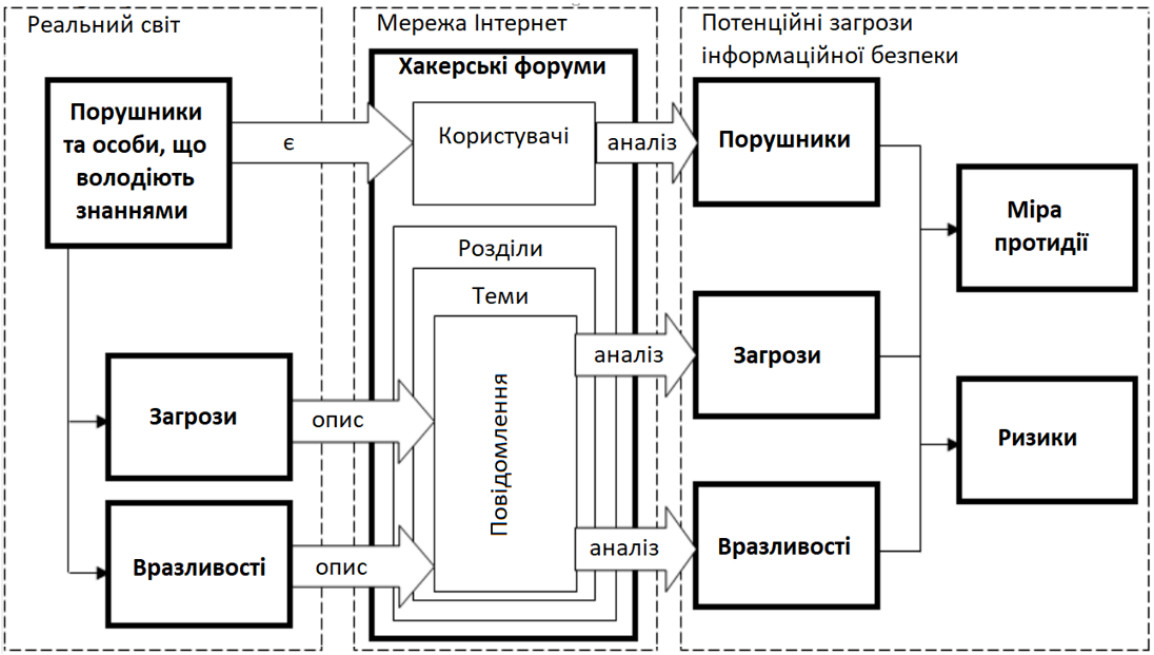


Рис.1. Інформаційне наповнення тематичних інтернет-ресурсів

Для повідомлень інтернет-форуму, доступна інформація, про автора, час його створення, приналежність до відповідного форуму та до теми форуму, кількості повідомлень по темі форуму, рейтинг автора. Наведена структура повідомлень дозволяє проводити статистичний та семантичний аналіз інформації форуму. В результаті проведення семантичного аналізу інформації форумів можливо здійснення фільтрації тих даних, які не мають відношення до заданої предметної області вразливостей та загроз безпеці інформації. На наступному кроці проведення аналізу виключаються дані, що не містять інформації про інформаційну безпеки, відповідно, досліджується інформація, що відноситься до вразливостей і загроз.

На теперішній час, для ефективного опису предметної галузі застосовується онтологія. При використанні даного підходу для опису предметної галузі застосовують її представлення у вигляді сукупності понять, враховуючи організацію та існуючі властивості, зв'язки між ними. Онтологічні механізми та методи дозволяють обчислювати відстань (близкість) повідомлень форумів до термінів предметної області, заданої онтологією. Повідомлення, що мають нульове значення коефіцієнта близькості (відстані) до термінів онтології, не мають відношення до предметної області, що аналізується [12].

Для функціонування тематичних інтернет-ресурсів характерна закономірність, яка полягає в наступному: з появою вразливості чи загрози безпеці інформації користувач форуму, якому відомо про загрозу, створює нову тему на інтернет форумі та залишає інформацію. Інші користувачі інтернет форуму залишають у новій створеній темі повідомлення, в яких спростовують чи доповнюють попередні повідомлення. Таким чином, в залежності від важливості інформації, що обговорюється на тематичному інтернет-ресурсі, проводиться оцінка внутрішнього рейтингу користувачів повідомлень. Високій значущості теми інтернет форуму відповідний високий рейтинг користувачів повідомлень, також закономірне збільшення частоти появи інформації у темі інтернет форуму, де обговорюється важливі повідомлення, особливо на початковій стадіяї проведення дискусії. Зазначені закономірності можуть бути задані та описані у вигляді відповідних правил нечітких продукцій, що застосовуються в інформаційних системах логічного нечіткого виводу.

Для прогнозування вразливостей та загроз безпеці конфіденційних даних можуть використовуватись результати проведеного аналізу повідомлень тем інтернет-ресурсів. Для вирішення даної задачі необхідно провести статистичний аналіз потоку даних інтернет-форуму та застосувати, при цьому, системи логічного нечіткого виводу. Учасники тематичних інтернет-форумів можуть створювати повідомлення, які не відносяться до предметної області, що аналізується. Для виключення їх з числа аналізованих доцільно застосовувати методи семантичного аналізу. Тобто, вхідними даними в системі нечіткого виводу можуть бути використанні статистичні параметри, що характеризують інформаційний процес обговорення вразливостей та загроз інформаційної безпеки. Нечіткі правила системи нечіткого виводу описують закономірності зміни потоку інформації інтернет-ресурсів, правила розміщенні в базі нечітких продукцій. Обґрунтованість використання нечітких моделей в системі протидії пов'язана зі значним ступенем присутньої невизначеності в інформації, що підлягає аналізу, складності предметної області та неповноти інформації інтернет-форумів [12,14].

Грунтуючись на результатах прогнозування, отриманих при виникненні раніше невідомих вразливостей та загроз інформаційної безпеки, спеціаліст, який здійснює захист інформації підприємства, може оцінити ступінь небезпеки атак та вжити необхідних заходів щодо усунення можливих загроз та вразливостей, переглянути відповідно ситуації моделі загроз інформаційної безпеки системи протидії.

Розглянуті особливості функціонування форумів тематичних інтернет-ресурсів дозволяють системі протидії здійснювати прогнозування виникнення вразливостей та загроз безпеки конфіденційних даних. Для вирішення цього завдання необхідне проведення аналізу інтернет повідомлень, створюваних учасниками форумів тематичних інтернет-ресурсів, що може бути здійснено відповідно до алгоритму прогнозування (рис. 2). Вхідними параметрами запропонованого алгоритму є список форумів тематичних інтернет-ресурсів, онтологія вразливостей та загроз безпеки інформації, система логічного нечіткого виводу.

Алгоритм передбачає виконання наступних кроків:

1. Пошук нових форумів тематичних інтернет-ресурсів та додавання виявлених до наявного списку форумів.

2. Пошук нових термінів предметної області у наявних тематичних інтернет-ресурсів вразливостей та загроз безпеки конфіденційних даних, додавання нововиявлених термінів в онтологію.

3. Збір потоку повідомлень тематичних інтернет-ресурсів.

4. Проведення семантичної фільтрації потоку повідомлень тематичних інтернет-ресурсів із використанням онтологічних методів.

5. Додавання інформації, що пройшла етап семантичної фільтрації інтернет-ресурсів, до бази даних прецедентів.

6. Статистичний аналіз потоку повідомлень, що зберігаються в базі даних прецендентів.

7. Логічний нечіткий вивід про виникнення вразливостей та загроз безпеки конфіденційних даних.

8. Підготовка відповідного звіту про виявлену вразливість чи загрозу безпеки інформації.

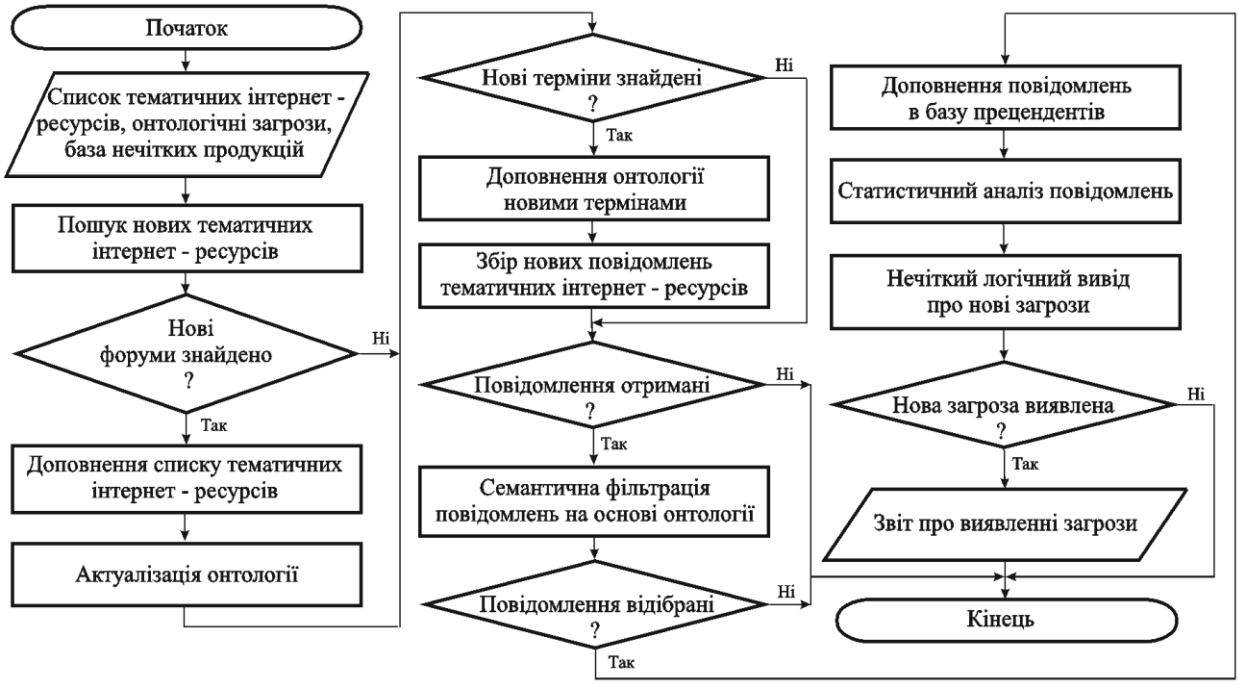


Рис.2. Алгоритм прогнозування вразливостей та загроз інформаційної безпеки

Результатом роботи алгоритму прогнозування вразливостей та загроз інформаційної безпеки є звіти про виявлених вразливостей, загроз інформаційної безпеки конфіденційних даних. До звітів можуть також включатися відомості, що відображають отриманні результати аналізу текстових повідомлень, на підставі яких здійснено висновок про виникнення вразливостей та загроз. Такими відомостями в період проведення аналізу можуть бути: частота створення участниками на форумах тематичних інтернет-ресурсів повідомлень, що відносяться до предметної області вразливостей та загроз інформаційної безпеки даних; частотна характеристика термінів вразливостей та загроз інформаційної безпеки даних, присутніх у повідомленнях інтернет-ресурсів; середній рейтинг користувачів повідомлень, що відносяться до предметної галузі вразливостей та загроз інформаційної безпеки даних; список присутніх у повідомленнях користувачів термінів онтології вразливостей та загроз інформаційної безпеки даних, що дозволяє класифікувати прогнозовані вразливості та загрози; добірка текстів тематичних інтернет-ресурсів, що містять терміни вразливостей та загроз інформаційної безпеки даних, створені на форумах.

Алгоритм прогнозування вразливостей та загроз безпеки інформації відрізняється можливістю виявлення вразливостей та загроз на ранніх етапах, їх практичної реалізації, ґрунтується на проведення аналізу потоку повідомлень форумів тематичних інтернет-ресурсів, що в даній ситуації дозволяє спеціалістам з інформаційної безпеки приймати адекватні та своєчасні заходи щодо захисту конфіденційних даних організації.

На підставі описаних вище особливостей функціонування тематичних інтернет-ресурсів та методів семантичної фільтрації текстових повідомлень послідовність проведення аналізу створюваних учасниками форуму повідомлень в період проведення аналізу може бути представлена алгоритмом фільтрації потоку тематичних повідомлень та статистичного аналізу інформаційної безпеки (рис. 3).

Запропонований алгоритм фільтрації потоку повідомлень та статистичного аналізу передбачає фільтрацію тематичних повідомлень, що не відносяться до заданої предметної області, яка задана відповідною онтологією, а також підрахунок кількості текстових повідомлень, що пройшли етап фільтрації потоку даних, та визначення середнього рейтингу авторів текстових повідомлень.

Вхідними параметрами алгоритму фільтрації потоку повідомлень та статистичного аналізу інформаційної безпеки є: D_τ – множина текстових повідомлень тематичних інтернет-ресурсів, створених в період проведення аналізу потоку даних; O – онтологія предметної області вразливостей та загроз інформаційної безпеки конфіденційних даних.

Основні кроки алгоритму проведення аналізу потоку текстових повідомлень наступні:

1. Обнулення значень K_τ – кількості тематичних повідомлень про вразливості та загрози інформаційної безпеки конфіденційних даних та A_τ – середнього рейтингу авторів тематичних повідомлень створених у період часу проведення аналізу τ.

2. Обчислення для кожного текстового повідомлення коефіцієнта k_{Ont} – близькості до термінів предметної області O заданої онтології.

3. Додавання тематичних повідомлень множини D_τ, для яких виконується нерівність $k_{Ont}>0$, до бази даних прецедентів для їх подальшого використання для формування відповідних звітів про прогнозування вразливостей та загроз інформаційної безпеки конфіденційних даних.

4. Обчислення K_τ – кількості повідомлень множини D_τ, для яких виконується нерівність $k_{Ont}>0$.

5. Обчислення A_τ – середнього рейтингу авторів тематичних повідомлень множини D_τ , для яких $k_{Ont}>0$.

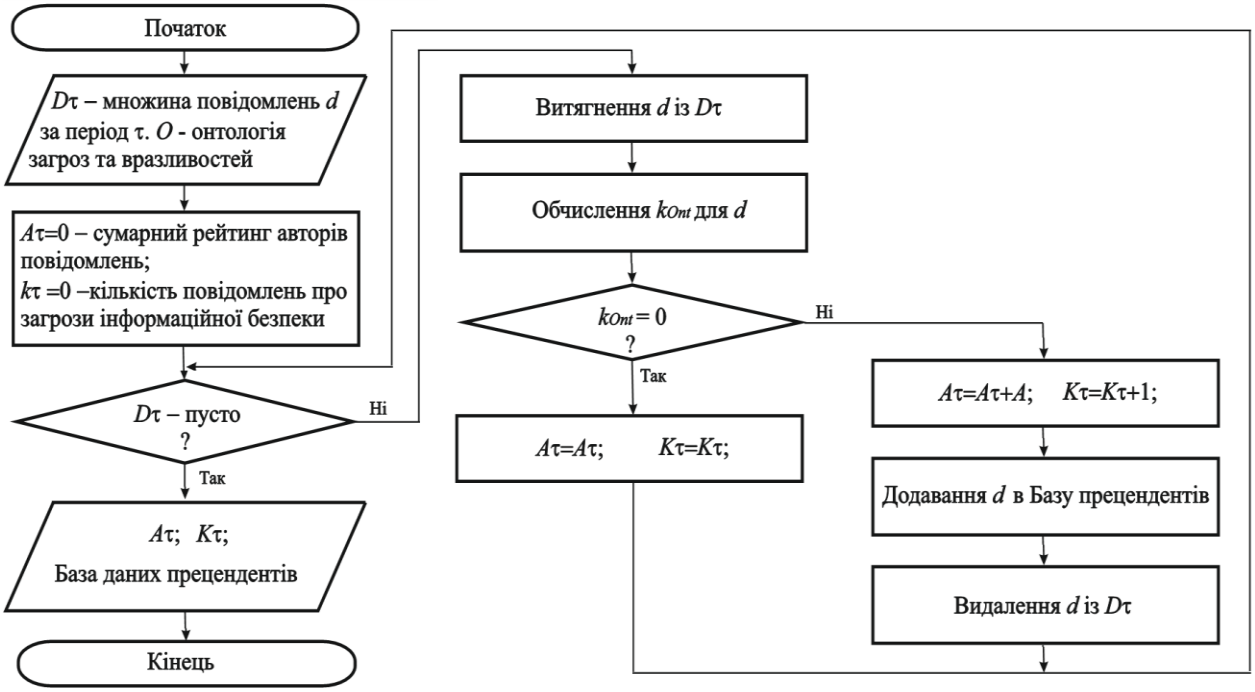


Рис.3. Алгоритм фільтрації та статистичного аналізу потоку тематичних повідомлень

Результатом роботи алгоритму є визначення статистичних показників, що характеризують потік тематичних повідомлень в період проведення аналізу потоку даних: K_τ – кількість текстових повідомлень, що містять терміни вразливостей та загроз інформаційної безпеки з онтології конфіденційним даним; A_τ – середній рейтинг авторів тематичних повідомлень, що містять терміни вразливостей та загроз інформаційної безпеки з онтології конфіденційним даним; поповнення бази даних прецедентів текстовими повідомленнями, що містять терміни вразливостей та загроз інформаційної безпеки з онтології конфіденційним даним. Алгоритм дозволяє обчислювати статистичні параметри, здійснювати семантичну фільтрацію текстових повідомлень. Результати роботи алгоритму можуть бути використанні для побудови системи логічного нечіткого виводу для прогнозування подій предметної області, для якої проводиться аналіз. Отриманні результати застосування алгоритму аналізу потоку текстових повідомлень можуть бути використані як значення вхідних параметрів у системі логічного нечіткого виводу та при формуванні звітів прогнозування вразливостей та загроз інформаційній безпеці організації.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Алгоритм прогнозування вразливостей та загроз безпеки інформації, заснований на логічному нечіткому виводі, семантичному та статистичному аналізі, відрізняється від аналогів можливістю виявлення вразливостей та загроз до їх безпосередньої реалізації. Він дозволяє гнучко описувати закономірності процесу наповнення тематичних форумів інтернет-ресурсів новими текстовими повідомленнями, що в результаті сприяє покращенню якості прогнозування загроз. Для вирішення задачі прогнозування вразливостей та загроз безпеки інформації запропоновано алгоритм фільтрації потоку тематичних повідомлень та статистичного аналізу інформаційної безпеки, заснований на семантичному та статистичному аналізі і відрізняється від аналогів можливістю обчислювати статистичні параметри, здійснювати семантичну фільтрацію текстових повідомлень, для прогнозування подій системи нечіткого логічного виводу.

Запропоновані алгоритми дозволяють прогнозувати вразливості та загрози, вживати адекватних заходів щодо захисту інформації. Отримані результати свідчать про ефективність запропонованих алгоритмів прогнозування вразливостей та загроз, а також підтверджують коректність роботи інформаційно-аналітичної системи та можливості застосування на практиці.

Література

1. Ленков С.В. Модель безпеки поширення забороненої інформації в інформаційно-телекомунікаційних мережах / С.В. Ленков, В.М. Джулій, В.С. Орленко, О.В. Сєлюков, А.В. Атаманюк. // Збірник наукових праць Військового інституту КНУ ім. Тараса Шевченка. – К.: ВІКНУ, 2020. – Вип. №68. – С. 53-64.

2. Ленков С.В. Інформаційно-аналітична системи прогнозування вразливостей та загроз інформаційної безпеки / С.В. Ленков, В.М. Джулій, О.В. Мірошніченко, В.О. Браун, С.І. Прохорський. // Збірник наукових праць Військового інституту КНУ ім. Тараса Шевченка.– К.: ВІКНУ, 2023. – Вип. №79. – С. 114-127.

3. Модель потоку текстових повідомлень тематичних інтернет-ресурсів системи прогнозування інформаційної безпеки / В. Джулій, Н. Петляк, Ю. Хмельницький, О. Пахар. // Вісник Хмельницького національного університету. Технічні науки. – 2022. – № 5. – С. 294-300.

4. Lienkov S., Podlipaiev V., Tolok I., Lisitsky I., .Lytvynenko N., Kuznichenko S. The Information and Analytical Using of Non-Structured Information Resources CEUR Workshop Proceedingsthis link is disabled, 2021, 3126, pp. 81–87.

5. Соціальні мережі – реальні загрози віртуального світу. [Електронний ресурс]. – Режим доступу : http://ogo.ua/ articles/view/011–02–23/26490.htm.

6. Ленков С.В. Методы и средства защиты информации. В 2-х томах /С.В. Ленков, Д.А. Перегудов, В.А. Хорошко – К: Арий, 2008. – 464с

7. Остапов С. Е. Технології захисту інформації: навчальний посібник / С.Е. Остапов, С.П. Євсеєв, О.Г. Король. – Харків : Вид-во ХНЕУ, 2016. – 476 с.

8. Аналіз існуючих методів та алгоритмів виявлення атак в бездротових мережах передачі даних / С.В. Лєнков, В.М. Джулій, Н.М. Берназ, С.О. Божук. // Збірник наукових праць Військового інституту КНУ ім. Тараса Шевченка. – К.: ВІКНУ. – 2017. – Вип. № 56. – С.124-132

9. Інформаційно-ознакова модель шкідливої інформації в соціальних мережах / І.В. Муляр, В.М. Джулій, В.М. Пічура, О.О Зацепіна. // Вимірювальна та обчислювальна техніка в технологічних процесах № 3 (2022). – С.73–78.

10. Модель потоку текстових повідомлень тематичних інтернет-ресурсів системи прогнозування інформаційної безпеки / В.М. Джулій, Ю.В. Хмельницький, Н.С. Петляк, О.В. Пахар. // Вісник Хмельницького національного університету. Технічні науки. 2022. № 5. С. 294-300с.

11. Контроль додатків інтернет-трафіка комп'ютерних мереж методами машинного навчання. / Джулій, В.М., Кльоц Ю.П., Муляр І.В., Жилевич М.Л., Джулій А.В. // Вісник Хмельницького національного університету. Технічні науки. 2021. № 5. С. 22-26.

12. Метод класифікації додатків трафіка комп'ютерних мереж на основі машинного навчання в умовах невизначеності / В.М. Джулій, О.В. Мірошніченко, Л.В. Солодєєва // Збірник наукових праць Військового інституту КНУ ім. Тараса Шевченка. – К.: ВІКНУ, 2022. – Вип. №74. – С. 73-82.

13. Математичні методи дослідження операцій : підручник / Є. А. Лавров, Л. П. Перхун, В. В. Шендрик – Суми: Сумський державний університет, 2017. – 212 с.

14. Гончар С. Ф. Оцінювання ризиків кібербезпеки інформаційних систем об'єктів критичної інфраструктури : монографія. / С. Ф. Гончар. – Київ, 2019. – 175 с.

15. Organizational Network Analysis as a Tool for Leadership Assessment in Software Development Team. / L.Yemchuk, O. Zhylinska; A. Chornyi; V. Dzhuliy // – Institute of Electrical and Electronics Engineers (30 September 2020); INSPEC Accession Number: 20008165; DOI: 10.1109/ACIT49673.2020.

References

1. Lenkov S.V. Model bezpeky poshyrennia zaboronenoi informatsii v informatsiino-telekomunikatsiinykh merezhakh / S.V. Lenkov, V.M. Dzhulii, V.S. Orlenko, O.V. Sieliukov, A.V. Atamaniuk // Zbirnyk naukovykh prats Viiskovoho instytutu KNU im/ Tarasa Shevchenka. – K.: VIKNU, 2020. – №68. – pp. 53-64.

2. Lenkov S.V. Informatsiino-analitychna systemy prohnozuvannia vrazlyvostei ta zahroz informatsiinoi bezpeky / S.V. Lenkov, V.M. Dzhulii, O.V. Miroshnichenko, V.O. Braun, S.I. Prokhorskyi // Zbirnyk naukovykh prats Viiskovoho instytutu KNU im/ Tarasa Shevchenka. – K.: VIKNU, 2023. – №79. – pp. 114-127.

3. Model potoku tekstovykh povidomlen tematychnykh internet-resursiv systemy prohnozuvannia informatsiinoi bezpeky / V. Dzhulii, N. Petliak, Yu. Khmelnytskyi, O. Pakhar // Herald of Khmelnytskyi National University. Technical sciences. – 2022. – № 5. – pp. 294-300.

4. Lienkov, S., Podlipaiev, V., Tolok, I., Lisitsky I., .Lytvynenko, N., Kuznichenko, S. The Information and Analytical Using of Non-Structured Information Resources CEUR Workshop Proceedingsthis link is disabled, 2021, 3126, стр. 81–87.

5. Cotsialni merezhi – realni zahrozy virtualnoho svitu. [Elektronnyi resurs]. – Rezhym dostupu : http://ogo.ua/ articles/view/011– 02–23/26490.htm

6. Metodyy sredstva zashchytы ynformatsyy. V 2-kh tomakh / S.V. Lenkov, D.A. Perehudov, V.A. Khoroshko – K: Aryi, 2008. –464s.

7. Tekhnolohii zakhystu informatsii: navchalnyi posibnyk / S.E. Ostapov, S.P. Yevseiev, O.H. Korol–Kharkiv : Vyd-vo KhNEU, 2016. – 476 s.

8. AnalIz Isnuyuchih metodiv ta algoritmiv viyavlennya atak v bezdrotovih merezhah peredachI danih / S.V. Lenkov, V.M. Dzhuliy, N.M. Bernaz, S.O. Bozhuk // Zbirnyk naukovykh prats Viiskovoho instytutu KNU im/ Tarasa Shevchenka. – K.: VIKNU. 2017. – Vip. № 56. – p.124-132

9. Informatsiino-oznakova model shkidlyvoi informatsii v sotsialnykh merezhakh/ I.V. Muliar, V.M. Dzhulii, V. M. Pichura, O.O Zatsepina – Vymiriuvalna ta obchysliuvalna tekhnika v tekhnolohichnykh protsesakh. – № 3 (2022) –S. 73–78.

10. Model potoku tekstovykh povidomlen tematychnykh internet-resursiv systemy prohnozuvannia informatsiinoi bezpeky / V.M. Dzhulii, Yu.V. Khmelnytskyi, N.S. Petliak, O.V. Pakhar // Herald of Khmelnytskyi National University. Technical sciences. 2022. № 5. S. 294-300s.

11. Kontrol dodatkiv internet-trafika kompiuternykh merezh metodamy mashynnoho navchannia. / V.M. Dzhulii, Yu.P. Klots, I.V. Muliar, M.L. Zhylevych, A.V. Dzhulii // Herald of Khmelnytskyi National University. Technical sciences.– Khmelnytskyi. – 2021, – №5. – pp. 22–26.

12. Dzhulii, V.M. (), Metod klasyfikatsii dodatkiv trafika kompiuternykh merezh na osnovi mashynnoho navchannia v umovakh nevyznachenosti / V.M. Dzhulii, O.V. Miroshnichenko, L.V. Solodieieva // Zbirnyk naukovykh prats Viiskovoho instytutu KNU im/ Tarasa Shevchenka. – K.: VIKNU. – 2022. – Vyp. №74. – pp. 73-82.

13. Matematychni metody doslidzhennia operatsii : pidruchnyk / Ye. A. Lavrov, L. P. Perkhun, V. V. Shendryk – Sumy : Sumskyi derzhavnyi universytet? 2017. – 212 p

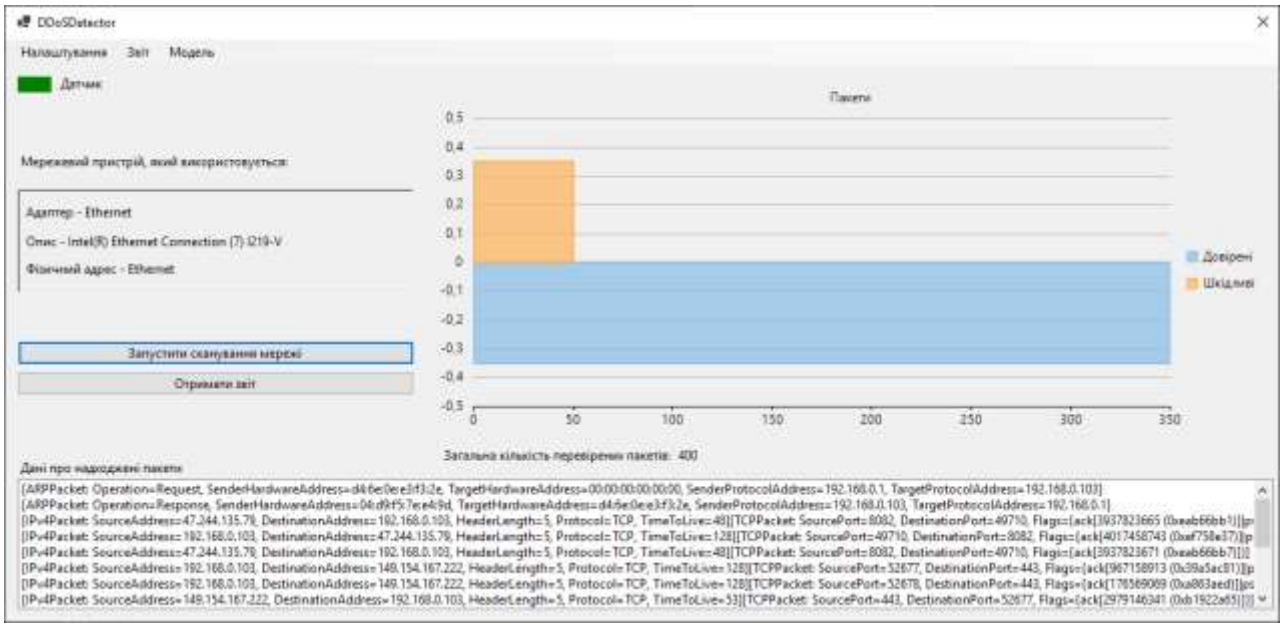
14. Otsiniuvannia ryzykiv kiberbezpeky informatsiinykh system obiektiv krytychnoi infrastruktury : monohrafiia. / S. F. Honchar. – Kyiv, 2019. – 175 s.

15. Organizational Network Analysis as a Tool for Leadership Assessment in Software Development Team. / L.Yemchuk, O. Zhylinska; A. Chornyi; V. Dzhuliy // Institute of Electrical and Electronics Engineers (30 September 2020); INSPEC Accession №: 20008165; DOI: 10.1109/ACIT49673.2020.

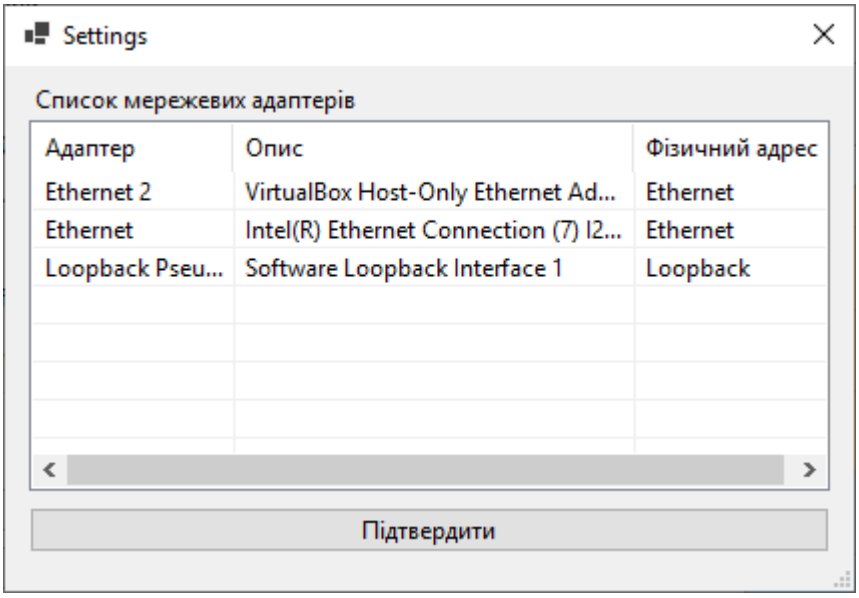
Додаток Б

Вигляд основних компонентів системи аналізу мережевого трафіка з використанням глибоких згорткових нейрнонних мереж

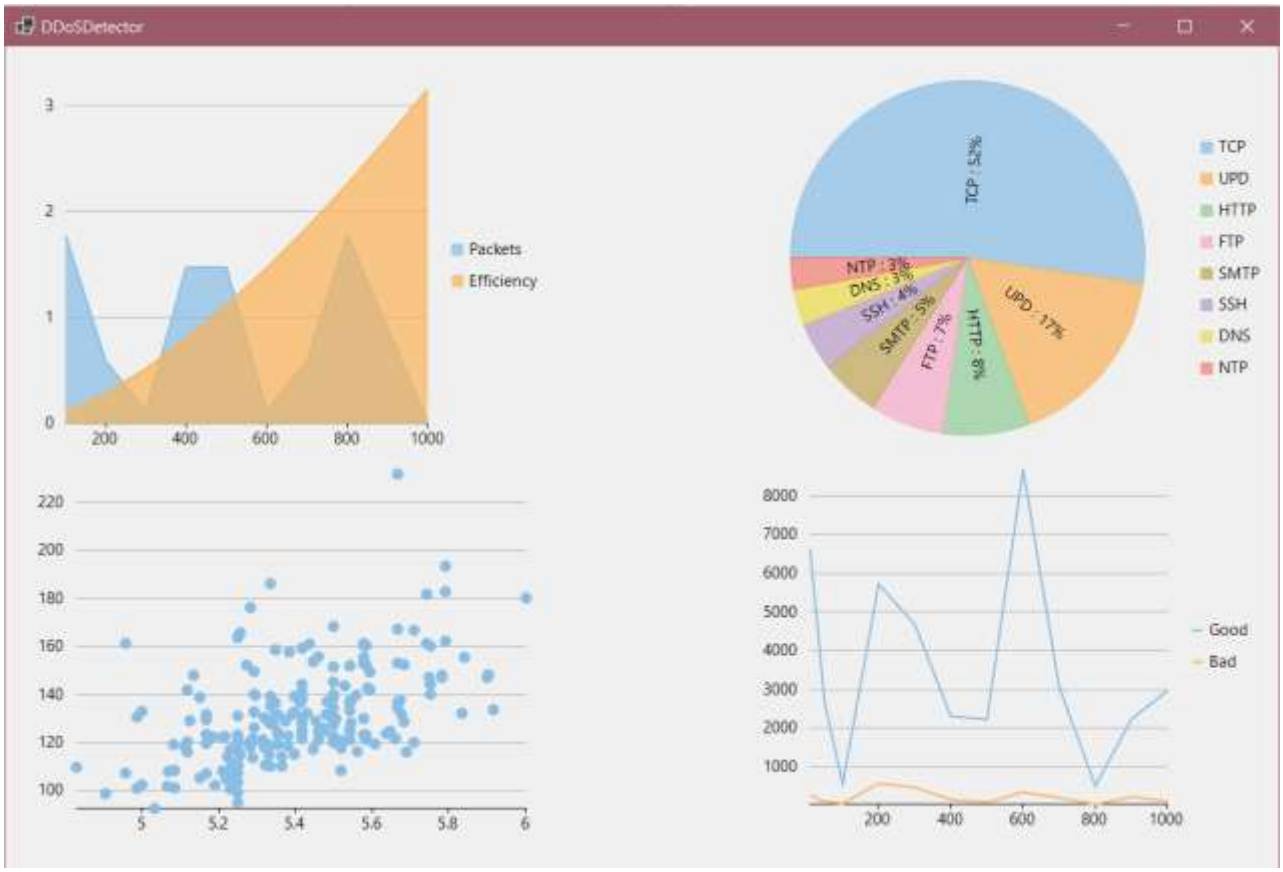
1. Вигляд головного вікна пристрою



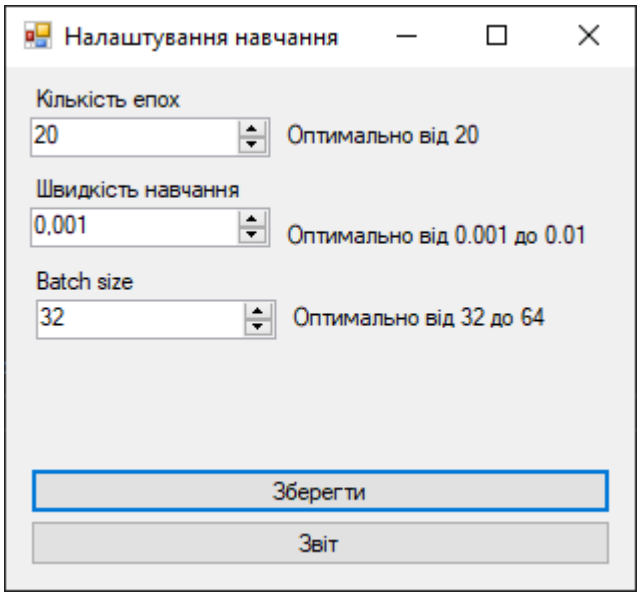
2. Вікно вибору адаптера



3. Вікно звіту



4. Форма тренування моделі



Додаток В

Частини коду реалізації системи

1. Повний вигляд класу NetworkData

```
using Microsoft.ML.Data;

public class NetworkData
{
    [LoadColumn(0)]
    public float Duration { get; set; }
    [LoadColumn(1)]
    public string ProtocolType { get; set; }
    [LoadColumn(2)]
    public string Service { get; set; }
    [LoadColumn(3)]
    public string Flag { get; set; }
    [LoadColumn(4)]
    public float SrcBytes { get; set; }
    [LoadColumn(5)]
    public float DstBytes { get; set; }
    [LoadColumn(6)]
    public int Land { get; set; }
    [LoadColumn(7)]
    public int WrongFragment { get; set; }
    [LoadColumn(8)]
    public int Urgent { get; set; }
    [LoadColumn(9)]
    public int Hot { get; set; }
    [LoadColumn(10)]
    public int NumFailedLogins { get; set; }
```

```
[LoadColumn(11)]
    public int LoggedIn { get; set; }
    [LoadColumn(12)]
    public int LnumCompromised { get; set; }
    [LoadColumn(13)]
    public int LrootShell { get; set; }
    [LoadColumn(14)]
    public int LsuAttempted { get; set; }
    [LoadColumn(15)]
    public int LnumRoot { get; set; }
    [LoadColumn(16)]
    public int LnumFileCreations { get; set; }
    [LoadColumn(17)]
    public int LnumShells { get; set; }
    [LoadColumn(18)]
    public int LnumAccessFiles { get; set; }
    [LoadColumn(19)]
    public int LnumOutboundCmds { get; set; }
    [LoadColumn(20)]
    public int IsHostL { get; set; }
    [LoadColumn(21)]
    public int IsHostK { get; set; }
    [LoadColumn(22)]
    public float Count { get; set; }
    [LoadColumn(23)]
    public float SrvCount { get; set; }
    [LoadColumn(24)]
    public float SerrorRate { get; set; }

    [LoadColumn(25)]
```

```
public float SrvSerrorRate { get; set; }
    [LoadColumn(26)]
    public float RerrorRate { get; set; }
    [LoadColumn(27)]
    public float SrvRerrorRate { get; set; }
    [LoadColumn(28)]
    public float SameSrvRate { get; set; }
    [LoadColumn(29)]
    public float DiffSrvRate { get; set; }
    [LoadColumn(30)]
    public float SrvDiffHostRate { get; set; }
    [LoadColumn(31)]
    public float DstHostCount { get; set; }
    [LoadColumn(32)]
    public float DstHostSrvCount { get; set; }
    [LoadColumn(33)]
    public float DstHostSameSrvRate { get; set; }
    [LoadColumn(34)]
    public float DstHostDiffSrvRate { get; set; }
    [LoadColumn(35)]
    public float DstHostSameSrcPortRate { get; set; }
    [LoadColumn(36)]
    public float DstHostSrvDiffHostRate { get; set; }
    [LoadColumn(37)]
    public float DstHostSerrorRate { get; set; }
    [LoadColumn(38)]
    public float DstHostSrvSerrorRate { get; set; }
    [LoadColumn(39)]
    public float DstHostRerrorRate { get; set; }
```

```
  [LoadColumn(40)]
    public float DstHostSrvRerrorRate { get; set; }
    [LoadColumn(41)]
    public string AttackType { get; set; }
}
```

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Майор Євген Віталійович
ПІБ здобувача вищої освіти

Студента ФІТ, 2 курсу, групи КБм-22-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

15.12.2023
дата

підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 10%**

	Документ		Сумарний збіг по Базі Даних	
ID: 123038 Назва: Метод виявлення шкідливих пакетів та DDoS атак на основі аналізу мережевого трафіку з використанням глибоких згорткових нейронних мереж Додано в БД: 2023-12-13 Автора: Майор Є.В, Керівники: Джулій В.М. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	82661	633	330 (0%)	4 (1%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1016002657

Дата перевірки:
13.12.2023 17:37:37 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
13.12.2023 17:41:00 EET

ID користувача:
100008300

Назва документа: **Майор на плагіат**

Кількість сторінок: **74** Кількість слів: **11581** Кількість символів: **87005** Розмір файлу: **2.16 MB** ID файлу: **1015686323**

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

1.13%
Схожість

Найбільша схожість: **0.52%** з джерелом з Бібліотеки (ID файлу: **1015654797**)

0.79% Джерела з Інтернету	111	Сторінка 76
0.78% Джерела з Бібліотеки	44	Сторінка 76

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	2
Підозріле форматування	21 сторінка

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:
Назва: Методи виявлення шкідливих пакетів та DDoS-атак на основі аналізу мережевого трафіка з використанням глибоких згорткових нейронних мереж

Автор: Майор Євген Віталійович
Спеціальність: 125 – Кібербезпека
Освітня програма: освітньо-професійна
Науковий керівник: Джулій Володимир Миколайович к.т.н, доцент
Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою Unichek складає 98,87%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism v-15.257 складає 99%.

Згідно з правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, авторська робота, обсяг оригінального тексту у відсотках до загального обсягу матеріалу в якій складає 90-100 %, визнається роботою з високою унікальністю тексту і допускається до захисту.

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1. Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 1.13%, з яких 0.52% є збігами з одним джерелом, зумовленими наявністю типових фразеологічних виразів предметної області, а також формулюваннями, які утворюють загальновживані фрази.

2. Інші три збіги є збігами в назвах використаних друкованих видань, розміщених в переліку джерел посилань

Керівник роботи В.М. Джулій

Гарант ОП В.Ю. Тітова

Завідувач кафедри кібербезпеки Ю.П. Кльоц

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітньо-кваліфікаційного рівня «магістр»

Студент Майор Євген Віталійович

Тема: «Метод виявлення шкідливих пакетів та DDoS-атак на основі аналізу мережевого трафіку з використанням глибоких згорткових нейронних мереж»

Галузь знань 12 «*Інформаційні технології*» Спеціальність 125 «*Кібербезпека*» Освітня програма «Кібербезпека»

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «магістр»: кількість сторінок записки 74;

1. Короткий зміст КР та прийнятих рішень: Кваліфікаційна робота присвячена дослідженню питань, пов'язаних з виявленням шкідливих пакетів у мережевому трафіку та DDoS-атакам із подальшим впровадженням системи, яка має можливість додатково навчатися та показувати оцінку ефективності роботи глибокої згорткової мережі. Для досягнення мети проведено дослідження особливостей атак на мережу класифіковано типи пакетів, описано ефективність та доцільність використання нейронних мереж у вирішення данної задачі, сформовано алгоритм та архітектуру система разом із нейронною мережею. Розроблено початкову модель на основі набору якісних вхідних даних, яка демонструє чудові результати і має можливість навчатися. Розроблено архітектуру системи, яка охоплює і вирішує поставлену задачу, враховуючи усі зовнішні фактори.
2. Висновок про відповідність КР завданню: Кваліфікаційна робота у повній мірі відповідає поставленому завданню як в теоретичній так і у практичній частині роботи.
3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі обґрунтовується актуальність теми роботи, її зв'язок з галуззю знань «Інформаційні технології» та спеціальністю «Кібербезпека», формулюється мета та основні завдання кваліфікаційної роботи. У першому розділі проведено аналіз існуючих способів створення атак на мережу, класифіковано шкідливі пакети та розглянуто застосування нейронних мереж у тематиці роботи. У другому розділі детально описано спосіб аналізу мережевого трафіку, необхідність формулювання тестових даних та моделі для роботи, описаний метод оцінки нейронної мережі, та проведений аналіз отриманих раніше результатів. У третьому розділі розроблена інформаційна система, яка влючатиме модуль по роботі із глибокою згортковою нейронною мережею та машинним навчанням, система розроблена відповідно до станлдартів, а архітектура мережі якісно створена для вирішення поставленої проблеми. У четвертому розділі наведені результати роботи системи, оцінка натренованої моделі, складність підбору нового наблору даних для покращення моделі і прописано висновки.
4. Позитивні сторони кваліфікаційної роботи: Кваліфікаційна робота оглядає великий спектр шкідливих пакетів, які бувають у мережі, а також класифікує їх. Також розглянуто та проаналізовано дослідження, які виявляють ефективність нерйонних мереж в сфері виявлення шкідливого трафіку, описана важливіть навчання моделі і додткового машинного навчання. Запропонована архітектура системи, яка має комплексний підхід до вирішення проблеми роботи. Розроблена архітектура глибокої згорткової мережі та архітектура системи, показали гарні результати у виявленні шкідливих пакетів і також особливістю, є додаткове навчання моделі, для оновлення даних і покращення роботи

5. Негативні сторони проекту: Не було достатньо чітко розглянуто методи оцінювання згорткових нейронних мереж, запропонована система має лише описовий характер, не було розроблено рішення під різні системи

6. Оцінка графічного оформлення та пояснювальної записки роботи:

7. Відгук про роботу в цілому: Загалом, робота має високу якість. Усі матеріали чітко структуровані, легко відслідковуються і логічно викладені. Кожен розділ послідовний і логічно сполучений з темою роботи, що робить його зрозумілим, а матеріал відповідає тематиці кваліфікаційної роботи.

8. Інші зауваження:

9. Оцінка дипломної роботи: Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що робота заслуговує оцінки «відмінно/ A (5,0)».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)

Лисенко Сергій Миколайович, д.т.н, професор, кафедра комп'ютерної інженерії та інформаційних систем, Хмельницького національного університету

« 14 » грудня 2023 . (підпис)