Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель загроз ІС на основі ШІ

(тема)

Виконав:

студент II курсу, групи КСМм-23-1

Проценко А.С.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма

Комп’ютерні системи та мережі
(повна назва освітньої програми)

Керівник: доц. Федорченко В.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Проценку Артему Сергійовичу
(прізвище, ім’я, по батькові)

1. Тема роботи Модель загроз ІС на основі ШІ

затверджена наказом по університету від “ 22 ” листопада 2024 р. № 1237 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 20 січня 2025 р.

3. Вхідні дані до роботи
Штучний інтелект
Системи виявлення вторгнень
Інформаційні системи

4. Перелік питань, що потрібно опрацювати у роботі
1. Проблема безпеки ІС
2. Сучасні засоби виявлення вторгнень
3. Використання ШІ у кібербезпеці
4. Огляд систем виявлення вторгнень
5. Розробка моделі загроз
6. Перевірка роботи моделі загроз
7. Прототип моделі загроз

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Лістинги скриптів, схеми, слайд-презентація – 20 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз безпеки ІС	26.11.24-30.11.24	
2	Аналіз використання ШІ для кібербезпеки	02.12.24-05.12.24	
3	Налаштування системи виявлення вторгнень	06.12.24-10.12.24	
4	Розробка ШІ	11.12.24-21.12.24	
5	Тестування моделі загроз	23.12.24-03.01.25	
6	Оформлення матеріалів кваліфікаційної роботи	04.01.25-07.01.25	
7	Подання кваліфікаційної роботи керівникові та	08.01.25-11.01.25	
	її попередній захист		
8	Подання кваліфікаційної роботи на рецензу-	13.01.25-17.01.25	
	вання		

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Федорченко В.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 68 с., 17 рис., 2 дод., 23 джерела.

ШТУЧНИЙ ІНТЕЛЕКТ, НЕЙРОННА МЕРЕЖА, ІНФОРМАЦІЙНА СИСТЕМИ, МОДЕЛЬ ЗАГРОЗИ, ТРАФІК, БЕЗПЕКА.

Метою кваліфікаційної роботи є підвищення безпеки інформаційних даних в комп'ютерних мережах шляхом побудови моделі загроз із використанням ШІ.

Об'єктом дослідження є система виявлення вторгнень до ІС. Саме вона буде взаємодіяти із нейронною мережею для запобігання атак.

Предметом дослідження є модель загроз, що буде побудовано із використанням системи виявлення вторгнень та ШІ.

У ході виконання кваліфікаційної роботи було виконано:

- аналіз можливих рішень для створення ефективнішої моделі загроз;
- вибір системи для інтеграції із ШІ;
- налаштування системи Snort під аналіз мережевого трафіку та побудову записів про підозрілу активність;
- написано програму для навчання нейронної мережі на основі датасету NSL-KDD із 123 записами про нормальні та небезпечні дії у мережі;
- було написано скрипт для інтеграції ШІ у систему Snort для аналізу записів про дії у мережі, виявлення та передбачення вторгнень.

Розроблена модель загроз передбачає постійну роботу зі сканування записів, створених системою Snort та, у разі виникнення підозрілих дій у мережі, швидкого реагування із попередженням користувача про небезпеку.

ABSTRACT

Master’s thesis: 68 pages, 17 figures, 2 appendices, 23 sources.

ARTIFICIAL INTELLIGENCE, NEURAL NETWORK, INFORMATION SYSTEMS, THREAT MODEL, TRAFFIC, SECURITY.

The major goal of this thesis is to improve the security of information data in computer networks by building a threat model using AI.

The object of the study is an intrusion detection system for IS. It will interact with a neural network to prevent attacks.

The subject of the study is a threat model that will be built using an intrusion detection system and AI. During the qualification work, the following was performed:

- analysis of possible solutions for creating a more effective threat model;
- selection of a system for integration with AI; - configuration of the Snort system for analyzing network traffic and building records of suspicious activity;
- a program was written for training a neural network based on the NSL-KDD dataset with 123 records of normal and dangerous actions on the network;
- a script was written for integrating AI into the Snort system for analyzing records of actions on the network, detecting and predicting intrusions.

The developed threat model involves constant scanning of records created by the Snort system and, in the event of suspicious actions on the network, a quick response with a warning to the user about the danger.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІС – інформаційна система

ШІ – штучний інтелект

CNN –згорткові нейронні мережі (англ. Convolutional Neural Network)

CPU – центральний процесор (англ. Central processing unit)

DL – глибинне навчання (англ. Deep Learning)

DNS – система доменних імен (англ. Domain Name System)

FTP – протокол передавання файлів (англ. File Transfer Protocol)

GPU – графічний процесор (англ. Graphics Processing Unit)

IDS / IPS – система виявлення вторгнень (англ. Intrusion Detection System) / система запобігання вторгненням (англ. Intrusion Prevention System)

IoT – інтернет речей (англ. Internet of Things)

IP – інтернет протокол (англ. Internet Protocol)

HTTP – протокол передачі даних (англ. Hypertext Transfer Protocol)

ML – машинне навчання (англ. Machine Learning)

NLP – обробка природної мови (англ. Natural Language Processing)

RAM – пам'ять з довільним доступом (англ. Random Access Memory)

ReLU – зрізаний лінійний вузол (англ. Rectified Linear Unit)

RNN – рекурентна нейронна мережа (англ. Recurrent Neural Networks)

SMB – протокол прикладного рівня (англ. Server Message Block)

SVM – метод опорних векторів (англ. Support Vector Machines)

TLS – захист на транспортному рівні (англ. Transport Layer Security)

TPU – тензорний блок обробки (англ. Tensor Processing Unit)

ВСТУП

Сучасні інформаційні системи (ІС) все більше інтегруються у повсякденне життя та бізнес-процеси, забезпечуючи автоматизацію, обробку та аналіз даних, а також підтримку прийняття рішень у реальному часі. Проте зі зростанням складності та обсягу інформаційних потоків зростають і ризики, пов'язані із захистом інформації та забезпеченням її цілісності, конфіденційності та доступності. Використання штучного інтелекту для побудови моделей загроз у ІС стає ефективним інструментом для ідентифікації, прогнозування та запобігання потенційним загрозам [1].

Тема цієї роботи – розробка моделі загроз для інформаційних систем на основі штучного інтелекту (ШІ) – є актуальною у зв'язку з високими вимогами до інформаційної безпеки та зростанням кількості кібератак. Використання алгоритмів машинного навчання, глибинного навчання та інших підходів ШІ дозволяє не лише виявляти загрози на ранніх етапах, але й адаптуватися до нових, раніше невідомих атак, покращуючи тим самим стійкість інформаційних систем.

Метою цієї роботи є аналіз існуючих підходів до побудови моделей загроз на основі ШІ та розробка власної моделі для підвищення рівня безпеки ІС. У ході дослідження були розглянуті основні методи машинного навчання, проаналізовано існуючі системи виявлення загроз та створено модель аналізу загроз на основі ШІ.

Об’єктом дослідження є система виявлення вторгнень до ІС. Саме вона буде взаємодіяти із нейронною мережею для запобігання атак.

Предметом дослідження є модель загроз, що буде побудовано із використанням системи виявлення вторгнень та ШІ.

1 ПІДХОДИ ДО ОЦІНКИ ЗАГРОЗ В ІС

1.1 Огляд області дослідження

1.1.1 Інформаційні системи у повсякденному житті

Інформаційні системи є невід'ємною частиною сучасних бізнес-процесів, управлінських і соціальних структур, забезпечуючи обробку, зберігання та передачу великих обсягів даних. Проте швидкий розвиток технологій та збільшення складності ІС роблять їх привабливими для кіберзлочинців. Зростаюча кількість кібератак, зокрема DDoS-атаки, фішингові атаки, вірусні програми та атаки на основі вразливостей програмного забезпечення, створює додаткові ризики та ускладнює завдання захисту інформації. Традиційні методи кібербезпеки, такі як брандмауери, антивірусне програмне забезпечення та системи виявлення загроз, хоча й залишаються важливими, не здатні повною мірою відповідати новим викликам.

У відповідь на ці виклики набувають популярності інтелектуальні методи захисту на основі ШІ, які дозволяють створювати адаптивні системи для виявлення та попередження кіберзагроз. Використовуючи алгоритми машинного та глибинного навчання, обробку природної мови та інші інструменти ШІ, можна будувати ефективні моделі загроз, що здатні не лише розпізнавати існуючі види атак, а й прогнозувати нові загрози, забезпечуючи гнучкий та інтегрований підхід до кібербезпеки.

Для захисту інформаційних систем зазвичай розробляється комплекс заходів (рисунок 1.1), спрямованих на забезпечення безпеки даних і стабільної роботи системи. Він включає організаційні аспекти, такі як розробка політик, навчання персоналу та контроль доступу, що регламентують взаємодію з інформаційною інфраструктурою.

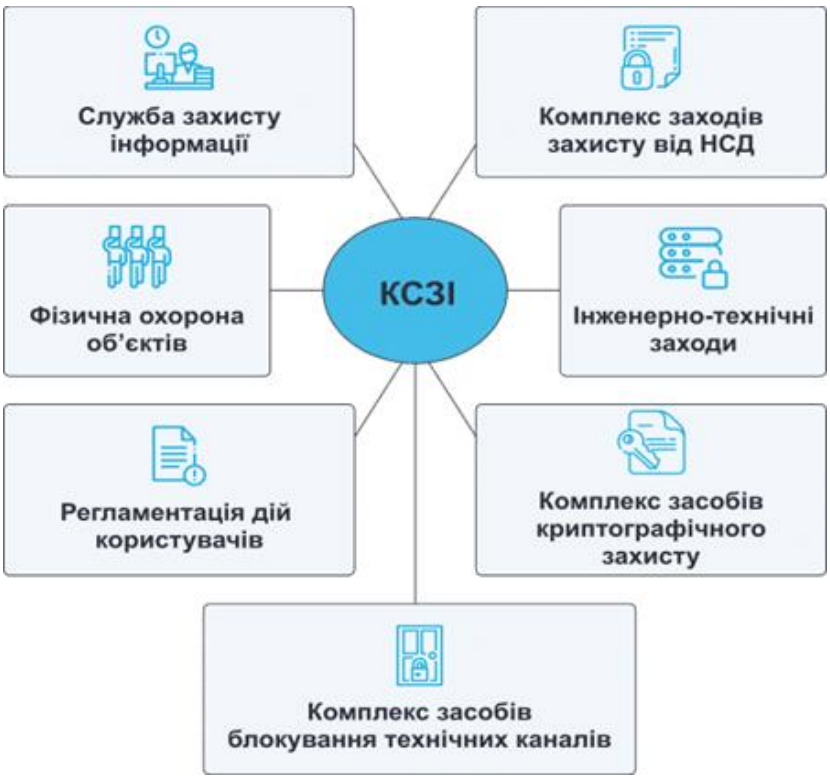

Рисунок 1.1 – Комплексна система захисту інформації

Технічний компонент охоплює використання міжмережевих екранів, систем виявлення загроз, шифрування, антивірусного програмного забезпечення та інших засобів для запобігання несанкціонованому доступу. Програмна частина забезпечується регулярним оновленням програмного забезпечення, закриттям вразливостей та моніторингом трафіку для виявлення аномалій. Фізичний захист спрямований на безпеку обладнання, наприклад, через контроль доступу до серверних приміщень і захист від стихійних лих.

Особливе значення має резервне копіювання, яке дозволяє відновити дані у випадку кібератак або збоїв. Моніторинг та аналіз системи допомагають швидко виявляти та реагувати на потенційні загрози, використовуючи сучасні засоби автоматизації та штучного інтелекту. Важливу роль відіграє швидке реагування на інциденти, ізоляція уражених компонентів і запобігання повторенню атак.

Ефективний захист можливий лише за умови узгодженої роботи всіх

цих компонентів як єдиного механізму, тому для цього будують певні моделі загроз.

1.1.2 Існуючі моделі загроз

Модель загроз – це структурований підхід до ідентифікації, аналізу та опису потенційних загроз, які можуть вплинути на інформаційну систему або інші цифрові активи. Моделі загроз забезпечують систематичний спосіб прогнозування та оцінки ризиків, з якими може зіткнутися система, допомагають зрозуміти можливі шляхи атак, їх наслідки та відповідні методи протидії.

Основним поняттям моделі загроз є будь-яка потенційна подія чи дія, яка може завдати шкоди інформаційній системі, зокрема її конфіденційності, цілісності або доступності. Загрози можуть бути як зовнішніми (наприклад, хакери), так і внутрішніми (недбалість або зловмисні дії працівників). Зазвичай слабким місцем або помилкою в системі, яку можуть використовувати зловмисники для реалізації загрози, може бути недостатній захист від атак або помилки в коді [2].

Процес використання вразливості для отримання доступу до системи або завдання їй шкоди називають атакою. Це реалізована загроза, що призводить до негативних наслідків. Ціллю цього стають ресурси, які мають цінність для організації, наприклад, інформаційні дані, програмне забезпечення, мережеве обладнання тощо. Захист активів є основною метою розробки моделі загроз.

Ціллю таких моделей є заходи або методи, спрямовані на зниження ризику або мінімізацію наслідків реалізації загрози. Це можуть бути як технічні засоби, так і організаційні політики для робітників.

Моделі загроз можуть бути класифіковані за різними підходами (рисунок 1.2), залежно від специфіки загроз та архітектури захищеної системи [3]:

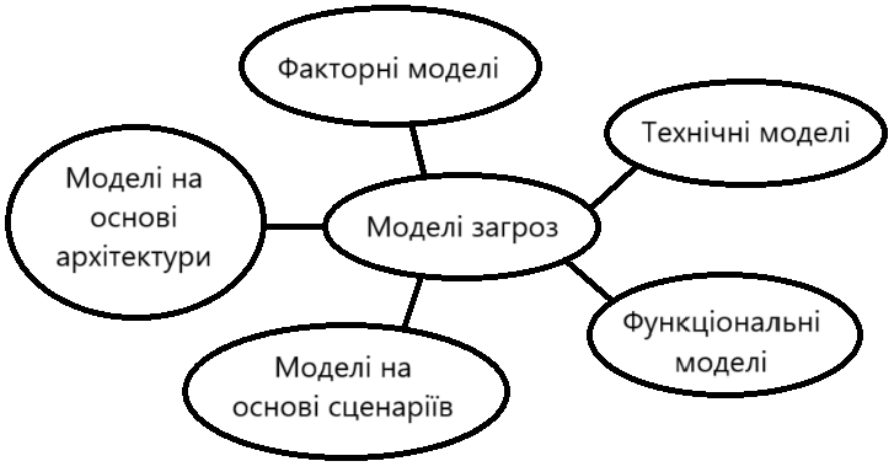

Рисунок 1.2 – Класифікація моделей загроз

- факторні моделі загроз фокусуються на дослідженні загроз, які можуть походити від конкретних типів зловмисників або атакуючих груп. До таких груп можуть належати внутрішні зловмисники (співробітники або підрядники), сторонні хакери, урядові агенції та організовані кіберзлочинні групи. Такі моделі загроз дозволяють ідентифікувати та розробити специфічні стратегії для захисту від різних груп зловмисників, наприклад тих, які не мають легального доступу до системи, зазвичай вдаються до технічних вразливостей або соціальної інженерії для проникнення в систему. Це особливо корисно в організаціях, які зберігають критичну інформацію або працюють у галузях із високими вимогами до безпеки;

- технічні моделі загроз орієнтовані на конкретні вразливості в технологіях, які використовуються в системі, зокрема на програмне забезпечення, мережеву архітектуру, інфраструктуру та окремі технічні процеси. Такі моделі передбачають глибокий аналіз можливих вразливостей, наприклад, у програмному коді, налаштуваннях мережі чи протоколах передачі даних. Якщо система використовує застарілий протокол передачі даних, атакуючі можуть скористатися цим для перехоплення інформації. Застосування технічних моделей загроз важливе в організаціях, які працюють

з критичними або конфіденційними даними та прагнуть знизити ризик атак через технічні уразливості;

- функціональні моделі загроз охоплюють певні функціональні можливості або процеси в системі, що можуть бути мішенню для атак. Наприклад, система електронної пошти є ключовим функціональним компонентом, тому ризик фішингових атак є однією з основних загроз для цього функціоналу. Як приклад, коли система включає важливі для бізнесу процеси, функціональна модель загроз дозволяє зосередитись на специфічних аспектах цих процесів, зокрема на захисті від атаки типу «людина посередині» в фінансових транзакціях або зловживання привілеями користувачів. Застосування функціональних моделей загроз актуальне для організацій, що обробляють великі обсяги чутливої інформації і де будь-яке порушення функціональних можливостей може значно вплинути на роботу системи;

- моделі загроз на основі сценаріїв передбачають аналіз конкретних сценаріїв атак, які можуть бути здійснені на систему. Такий підхід дозволяє створити і проаналізувати різні сценарії розвитку подій під час атаки, визначити можливі шляхи проникнення зловмисників і оцінити наслідки. Як приклад, сценарій компрометації даних користувачів шляхом несанкціонованого доступу або сценарій поширення шкідливого ПЗ через вразливості в мережі. Застосування таких моделей є особливо ефективним для компаній, що прагнуть заздалегідь імітувати можливі інциденти безпеки і підготувати персонал до реагування на них [4];

- моделі на основі архітектури зосереджуються на аналізі загроз, пов’язаних зі специфікою архітектури інформаційної системи. Розподілені системи, хмарні обчислення, інтернет речей мають свої унікальні архітектурні особливості, які можуть стати вразливими місцями для атак. Через спільне використання ресурсів та віддалений доступ хмарні сервіси вразливі до атак на рівні авторизації та конфіденційності. Наприклад, зловмисники можуть скористатися неправильно налаштованими дозволами

до загальнодоступних файлів або баз даних. Застосування моделей загроз на основі архітектури допомагає організаціям глибше розуміти специфіку кіберзагроз для своєї інфраструктури, зокрема якщо вона включає розподілені або хмарні елементи [5].

Такі моделі загроз дозволяють здійснювати більш глибокий та обґрунтований підхід до кіберзахисту, оскільки вони допомагають заздалегідь виявляти можливі сценарії атак, аналізувати слабкі місця системи та оптимізувати стратегії захисту. Такий підхід забезпечує ефективний розподіл ресурсів на кібербезпеку, що є особливо важливим у великих інформаційних системах з великою кількістю активів.

Завдяки використанню моделей загроз можна не лише підвищити рівень захисту ІС, але й знизити кількість помилкових спрацьовувань та непотрібних витрат, направляючи основні зусилля на найбільш критичні аспекти безпеки.

1.2 Актуальність та затребуваність використання ШІ

В умовах глобальної цифровізації, збільшення обсягів даних та необхідності захисту конфіденційної інформації важливість забезпечення кібербезпеки постійно зростає. Здатність ІС до самонавчання, адаптивності та швидкого реагування на загрози стає критично важливою для підтримки стабільної роботи компаній та організацій. Використання ШІ для розробки моделей загроз є одним із найбільш актуальних напрямів у сучасній кібербезпеці. Завдяки технологіям ШІ вдається знижувати ризики, пов'язані з проникненням до ІС сторонніх осіб, запобігаючи потенційним фінансовим втратам та збереженню репутації компаній.

Затребуваність виявлення загроз за допомогою інструментів, що використовують ШІ, обумовлена загальносвітовими трендами в кібербезпеці та інтенсивним розвитком технологій, які стають джерелом нових викликів і ризиків для інформаційної безпеки. Рішення, засновані на ШІ, дають змогу

автоматизувати процеси виявлення та класифікації загроз, оптимізуючи роботу відділів безпеки та мінімізуючи людські помилки. Це дозволяє значно підвищити ефективність роботи ІС, одночасно скорочуючи час реагування на загрози.

Такі моделі набувають все більшої популярності у великих корпораціях, державних установах, банках, страхових компаніях та інших організаціях, які прагнуть захистити свої активи від сучасних кіберзагроз. Їх впровадження також є необхідним у зв’язку з нормативними вимогами щодо забезпечення інформаційної безпеки, що стимулює активне дослідження та розвиток інструментів ШІ у сфері кібербезпеки.

1.3 Переваги перед існуючими методами

1.3.1 Класичні методи пошуку загроз

Існує кілька класичних методів пошуку загроз у ІС. Вони широко використовуються для забезпечення базового рівня кібербезпеки та виявлення потенційних загроз [6].

Сигнатурний аналіз є одним з найстаріших і найпоширеніших методів пошуку загроз. Він базується на порівнянні вхідних даних з відомими шаблонами загроз (сигнатурами), які зберігаються в базах даних, приклад наведений на рисунку 1.3. Основними особливостями сигнатурного аналізу є системи пошуку загроз, що порівнюють файли, процеси чи мережевий трафік з базою відомих сигнатур вірусів, шкідливих програм або атак. Якщо знайдено відповідність, загроза визначається як відома і блокується або поміщується у карантин. Сигнатурний аналіз є швидким і точним для виявлення загроз, якщо вони відповідають вже відомим шаблонам. На жаль він не є ефективним проти нових або модифікованих загроз, що не мають відомих сигнатур. Це потребує постійного оновлення баз даних.

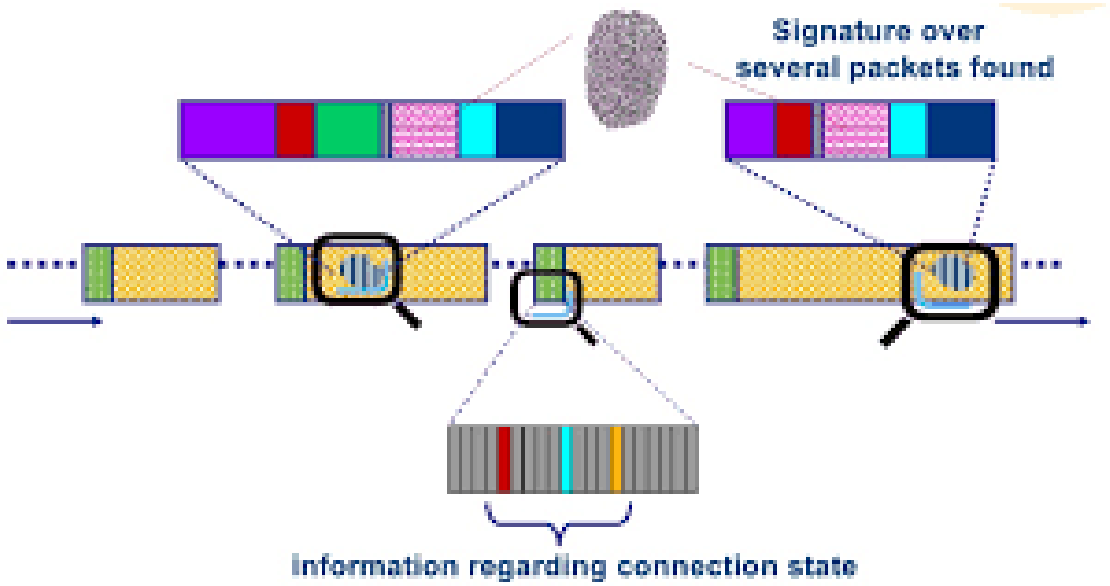

Рисунок 1.3 – Приклад сигнатурного аналізу масиву даних

Контроль доступу та управління правами фокусується на обмеженні доступу користувачів та програм до системних ресурсів, щоб запобігти несанкціонованим діям. Основним аспектом контролю доступу є принцип, за яким кожному користувачеві або програмі присвоюються права доступу відповідно до їхніх ролей і завдань. Система контролює, щоб дії відповідали дозволеним правам. Будь-яка спроба виконати несанкціоновану дію розглядається як потенційна загроза. Це значно знижує ризик несанкціонованих дій всередині системи, а також дозволяє виявляти підозрілу активність відразу після її виникнення. Проте є складність у налаштуванні для великих організацій із багаторівневою структурою прав. До того ж цей метод не захищає від зовнішніх загроз або атак, які використовують слабкі місця в ПЗ.

Сканування вразливостей передбачає перевірку системи на наявність відомих вразливостей, які можуть бути використані зловмисниками для атаки. Основою такого сканування є автоматизовані сканери вразливостей, що перевіряють систему на наявність слабких місць, таких як невиправлені вразливості в ПЗ, неправильні налаштування безпеки або слабкі паролі. Воно

дозволяє заздалегідь виявити відомі слабкі місця та усунути їх до того, як вони будуть використані для атаки, але не захищає від нових або невідомих вразливостей. Сканування може займати тривалий час і потребує регулярного оновлення баз вразливостей.

Аналіз поведінки передбачає спостереження за активністю користувачів та процесів з метою виявлення підозрілих дій. Метод фокусується на пошуку аномалій у поведінці, які можуть свідчити про наявність загрози. Для цього налаштовуються базові параметри нормальної поведінки користувачів, процесів або мережевого трафіку, приклад наведений на рисунку 1.4. Якщо активність відхиляється від норми (наприклад, велика кількість спроб входу, підозріла передача даних), це може розглядатись як ознака потенційної загрози. Такий аналіз дозволяє виявляти нетипову поведінку, яка може свідчити про загрозу, навіть якщо вона не відповідає відомим шаблонам. На жаль аналіз поведінки може давати велику кількість помилкових спрацьовувань, а також потребує ретельного налаштування параметрів для уникнення неправдивих тривог.

Рисунок 1.4 – Приклад нормальної поведінки системи

Аудит журналів – це методичне вивчення журналів подій, що ведуть облік усіх дій в інформаційній системі, таких як спроби входу, доступ до файлів, зміни налаштувань тощо. Системні журнали, або логи, аналізуються

вручну або автоматично на предмет аномальних записів, які можуть свідчити про спроби атак, намагання проникнення або інші несанкціоновані дії. Це дозволяє відстежувати активність в системі та визначати потенційні загрози через аналіз аномальних подій. Журнали забезпечують хронологічну інформацію, яка може бути корисною для розслідування інцидентів. Великим недоліком є те, що аналіз журналів потребує багато часу і часто виконується вручну, що може затримувати виявлення загроз. Також можливе перевантаження інформацією у великих системах, де записується багато подій.

Метод білого та чорного списків передбачає створення списків дозволених (білий список) і заборонених (чорний список) дій або програм. Він широко використовується в брандмауерах, антивірусному програмному забезпеченні та системах управління доступом. У білому списку знаходяться дозволені дії або процеси, які можуть виконуватись у системі. У чорному списку – заборонені дії або програми, які слід блокувати. Якщо активність не відповідає жодному з дозволених пунктів, вона блокується. Це дуже простий у впровадженні метод, особливо ефективний для блокування відомих загроз та зменшення обсягу невідомих загроз, але він неефективний проти нових загроз, що не внесені до чорного списку, і може призводити до затримок в оновленні списків, що знижує оперативність захисту.

Мережевий моніторинг дозволяє аналізувати мережевий трафік для виявлення аномалій або небажаних підключень. Цей метод використовується в брандмауерах та системах виявлення вторгнень. Система аналізує мережевий трафік у реальному часі, перевіряючи його на відповідність правилам безпеки. Виявлення підозрілого трафіку, наприклад, несанкціонованих з'єднань або великої кількості запитів за короткий час, може вказувати на потенційну загрозу. Таке спостереження дозволяє виявляти атаки на ранніх етапах і запобігати підозрілим діям у реальному часі. Моніторинг також може генерувати багато помилкових спрацьовувань і потребує налаштування правил, що може бути складним для великих мереж.

1.3.2 Аналіз програмних продуктів для моделювання загроз

Моделювання загроз рекомендовано проводити на початку циклу розробки системи, коли потенційні проблеми можна виявити та усунути на ранніх етапах, запобігаючи значним витратам на ліквідацію наслідків атак зловмисників. Для цього існує два найвідоміші інструмента, а саме OWASP Threat Dragon та Microsoft Threat Modeling Tool.

OWASP Threat Dragon – це інструмент моделювання, який використовується для створення діаграм моделі загроз у рамках безпечного життєвого циклу розробки. Він дотримується цінностей і принципів маніфесту моделювання загроз (рисунок 1.5). Його можна використовувати для запису можливих загроз і прийняття рішень щодо їх пом'якшення, а також для надання візуальної індикації компонентів моделі загроз і поверхонь загроз. Threat Dragon працює як веб-програма або як настільна програма [7].

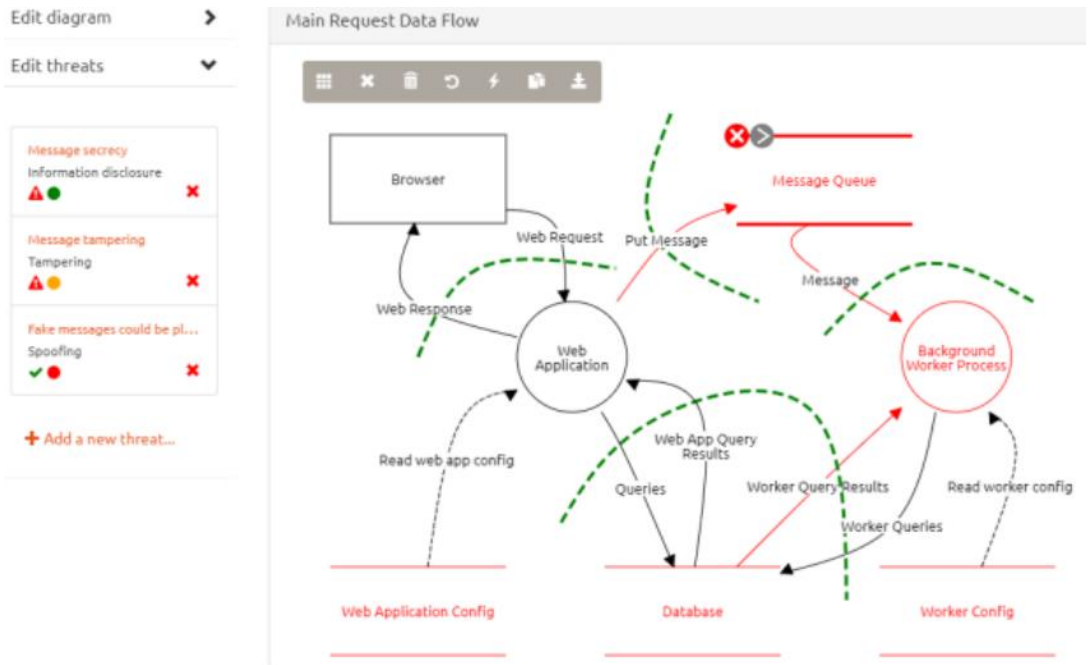

Рисунок 1.5 – Приклад побудованої діаграми загроз за допомогою програми OWASP Threat Dragon

Microsoft Threat Modeling Tool – це інструмент моделювання загроз, який є основним елементом життєвого циклу розробки безпеки Microsoft (SDL). Це дозволяє архітекторам програмного забезпечення виявляти та пом'якшувати потенційні проблеми безпеки на ранній стадії, коли їх відносно легко та економічно ефективно вирішити. В результаті це значно знижує загальну вартість розробки. Крім того, було розроблено інструмент з урахуванням експертів, не пов'язаних із безпекою, що полегшує моделювання загроз для всіх розробників, надаючи чіткі вказівки щодо створення та аналізу моделей загроз (рисунок 1.6) [8].

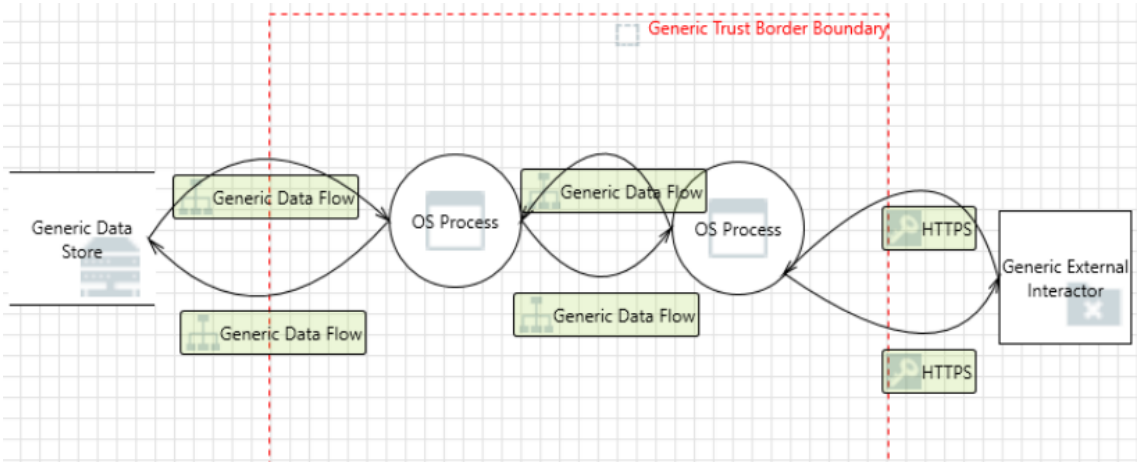

Рисунок 1.6 – Приклад побудованої діаграми загроз за допомогою утиліти Microsoft Threat Modeling Tool

Проаналізувавши дані програми, можна виділити механізм побудови захисту. Він включає в себе моделювання певних загроз для аналізу й подальшої розробки методів боротьби із ними. Це дозволить, при реальній загрозі, мати вже готову систему захисту ІС.

1.3.3 Використання ШІ для пошуку загроз

Головною перевагою використання ШІ у кібербезпеці є автоматизація. Будь який механізм, створений людиною, все одно потребує власноручне

введення параметрів для сканування ІС. ШІ забезпечують автоматичне виявлення загроз, скорочуючи людське втручання та мінімізуючи ймовірність помилок, що значно підвищує ефективність і точність роботи ІС.

Також, на відміну від постійного ручного оновлення систем захисту, нейронні мережі здатні до самонавчання. Тому ШІ можуть адаптуватися до змінних умов та навчатися на нових даних, що дозволяє їм розпізнавати нові, невідомі раніше загрози та швидко реагувати на них.

На відміну від людини, ШІ, завдяки використанню алгоритмів машинного навчання, здатні прогнозувати майбутні загрози, дозволяючи вжити необхідних заходів ще до настання потенційного ураження.

Також, якщо використовувати глибинні нейронні мережі, можна досягти високого рівня точності у визначенні аномальної поведінки та загроз, що забезпечує надійний захист ІС [9].

Однак використання штучного інтелекту для пошуку загроз має певні недоліки. Моделі сильно залежать від якості даних, на яких вони навчаються, тому можуть бути менш ефективними у виявленні нових або рідкісних атак. Через складність нейронних мереж їх важко інтерпретувати, що ускладнює пояснення рішень системи. Також можливі помилкові спрацьовування, які перевантажують операторів. ШІ вимагає значних обчислювальних ресурсів і регулярного оновлення моделей, щоб залишатися актуальним у змінних умовах. Крім того, його можуть обійти зловмисники, використовуючи спеціальні методи модифікації атак. Інтеграція таких рішень у наявні системи безпеки може бути складною, а аналіз мережевого трафіку іноді викликає етичні чи правові питання.

Таким чином, технології ШІ мають як суттєві недоліки, так і значні переваги над традиційними методами кібербезпеки та відіграють важливу роль у створенні комплексних систем захисту інформаційних систем, здатних забезпечувати високий рівень безпеки в умовах постійно зростаючих кіберзагроз.

1.4 Постановка задачі

Задачею даної кваліфікаційної роботи є аналіз існуючих методів, технологій і алгоритмів для подальшого створення прототипу нейронної мережі здатної аналізувати інформацію, що до неї надходить, та попереджати користувача про небезпеку. Під час роботи буде обрано методику, тип ШІ та певні підходи до його реалізації. Також буде описано систему із якою буде працювати нейронна мережа.

2 МЕТОДИ ПОШУКУ ІНФОРМАЦІЙНИХ ЗАГРОЗ

2.1 Огляд технологій, методів і алгоритмів

Сучасні ІС потребують вдосконалених підходів до забезпечення безпеки, особливо з огляду на збільшення обсягів даних і зростання кількості кібератак. Використання технологій ШІ стає ключовим компонентом для побудови моделей загроз, що забезпечують високий рівень захисту. Основні методи і алгоритми, що застосовуються для виявлення, прогнозування та запобігання загрозам в ІС, можна розділити на декілька категорій: методи машинного навчання (ML), глибинного навчання (DL) та обробки природної мови (NLP). У цьому розділі наведено огляд цих технологій і їхнє значення для побудови моделей загроз.

2.1.1 Машинне навчання

Методи машинного навчання є основою для створення адаптивних систем, які здатні виявляти загрози в реальному часі. Серед найбільш поширених алгоритмів машинного навчання для виявлення загроз можна виділити метод опорних векторів, що є алгоритмом класифікації даних, та метод K-середніх, що є алгоритмом кластеризації [10].

Метод опорних векторів (SVM) допомогає розділяти інформацію на категорії на основі попередніх спостережень, приклад наведений на рисунку 2.1. SVM ефективний у виявленні аномалій та виявленні загроз, оскільки здатен класифікувати нові дані як підозрілі або безпечні.

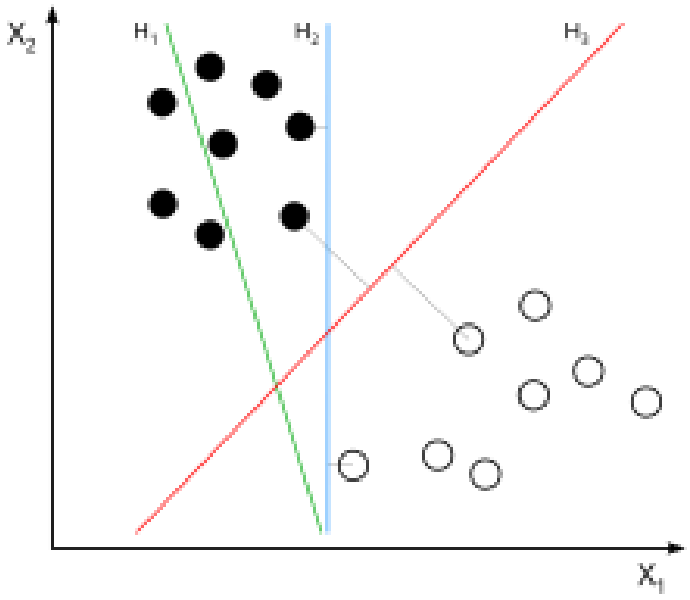

Рисунок 2.1 – Візуальний приклад методу опорних векторів

Щодо методу K-середніх (K-means), цей алгоритм розбиває дані на групи (кластери) на основі схожості між об'єктами. Він використовується для ідентифікації аномалій у поведінці користувачів та у мережевому трафіку (рисунок 2.2).

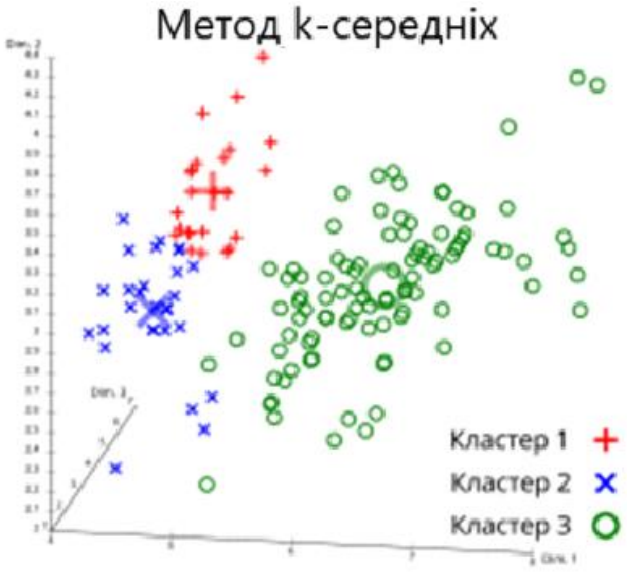

Рисунок 2.2 – Візуальний приклад кластеризації методом K-середніх

Окрім ефективного пошуку аномалій цінуються й інші методи захисту, а саме випадковий ліс та градієнтне посилення. Ці алгоритми спрямовані на виявлення фішингових сайтів, фільтрації спаму та аналізі поведінки користувачів. Випадковий ліс створює найкраще рішення на основі дерева можливих варіантів, але з них двох ефективнішим можна виділити саме градієнтне посилення, так як він є поєднанням слабких алгоритмів для підвищення точності прогнозів, а саме його використання є продуктивним, як на маленьких обсягах даних, так і на великих наборах, що дозволяє підвищити точність класифікації.

2.1.2 Глибинне навчання

Методи глибинного навчання використовуються для побудови більш складних моделей, які здатні розпізнавати складні шаблони і закономірності в даних. Основні архітектури, що застосовуються для виявлення загроз [11]:

- згорткові нейронні мережі (CNN), які зазвичай використовуються для обробки зображень, однак можуть застосовуватись і для аналізу мережевих даних та класифікації трафіку в ІС, схема навчання приведена на рисунку 2.3. Вони дозволяють виявляти відмінності між нормальним та аномальним трафіком;

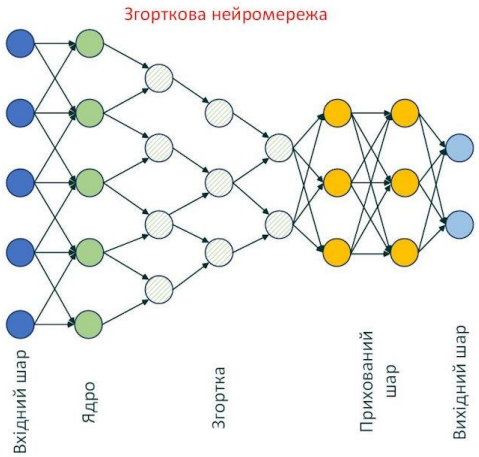

Рисунок 2.3 – Схема навчання згорткової нейронної мережі

- рекурентні нейронні мережі (RNN), які, завдяки можливості обробки послідовних даних, є дуже ефективними для аналізу подій, що залежать від часу, схема навчання приведена на рисунку 2.4. Їх використовують для виявлення аномалій у поведінці користувачів, прогнозування кібератак та виявлення послідовностей подій, які можуть вказувати на потенційну загрозу.

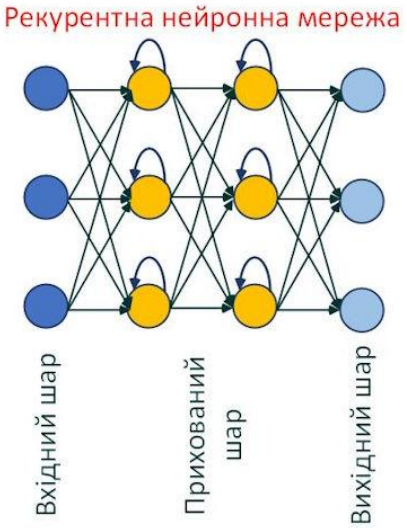

Рисунок 2.4 – Схема навчання рекурентної нейронної мережі

2.1.3 Обробка природної мови (NLP)

Технології NLP застосовуються для автоматизації аналізу текстових даних, таких як журнали безпеки, звіти про інциденти та інформація з відкритих джерел, яка може містити дані про потенційні загрози, архітектура модуля обробки природної мови наведена на рисунку 2.5. Основні методи NLP, що використовуються у створенні моделей загроз, включають в себе токенізацію та стематизацію. Ці методи дозволяють розділяти текст на окремі слова чи частини (токени) та визначати кореневі форми слів. Вони допомагають структурувати інформацію з текстових даних, що спрощує їхній подальший аналіз [12].

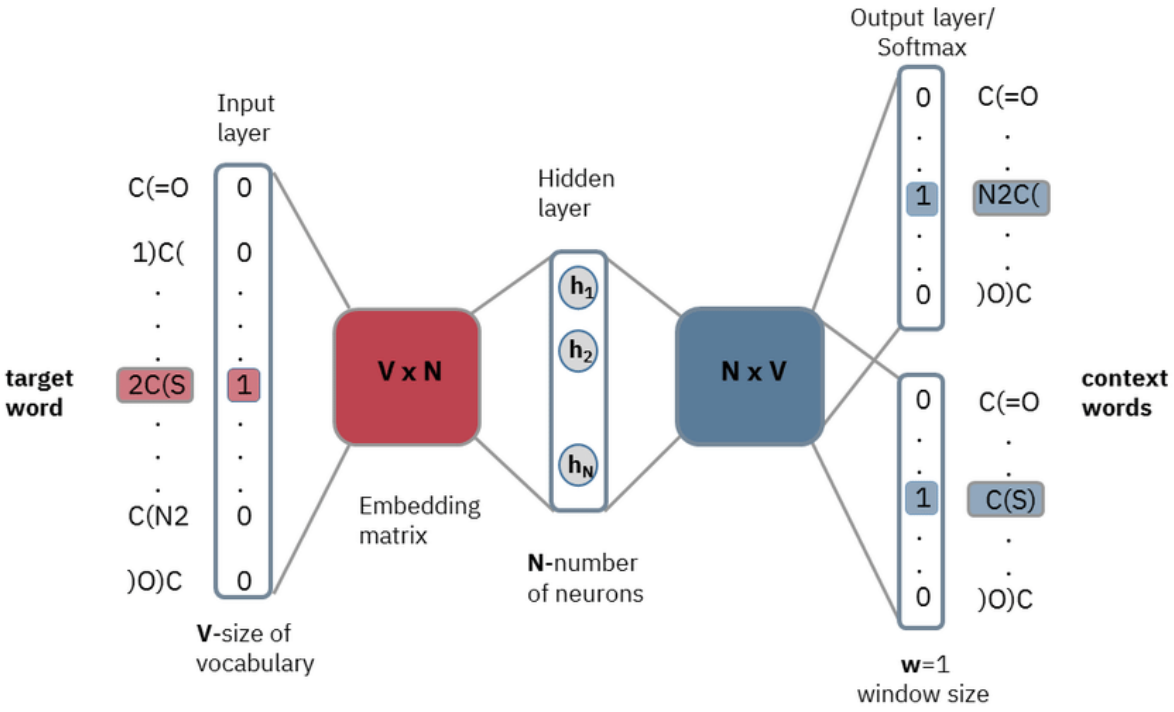

Рисунок 2.5 – Архітектура модуля обробки природної мови

Також існує «Bag of Words» (BoW) – це метод подання тексту у вигляді частотного розподілу слів, який дозволяє створити числове представлення тексту для подальшого аналізу. BoW використовується для класифікації текстів і визначення ключових слів, що можуть вказувати на наявність загроз при механізмах авторизації або роботи із базами даних.

Ще слід загати про вкладання слів або Word Embeddings (Word2Vec, GloVe). Ці методи дозволяють представляти слова у вигляді числових векторів, що враховують їхні семантичні зв'язки. Це полегшує аналіз змісту тексту та дозволяє виявляти контекст загроз, що описуються у звітах і журналах.

Також існує механізм Sentiment Analysis. Він аналізує тональності тексту, що дозволяє виявляти настрій або емоційне забарвлення текстових даних. Цей метод може бути корисним для виявлення негативних відгуків, скарг користувачів або інших текстів, що вказують на можливі проблеми з безпекою.

2.1.4 Інтеграція та комбінування методів

Для побудови надійної моделі загроз часто використовують комбінування різних методів машинного та глибинного навчання, а також NLP. Наприклад, спільне використання методів класифікації дозволяє не лише виявляти вже відомі загрози, але й знаходити нові аномалії у поведінці користувачів або мережевому трафіку [13].

Комбіновані підходи дозволяють моделювати загрози з більшою точністю і ефективністю, створюючи адаптивні, інтелектуальні системи захисту для ІС. Використання алгоритмів ШІ у виявленні загроз є перспективним напрямом розвитку кібербезпеки, оскільки забезпечує високу ефективність захисту та знижує вплив людського фактору на процес виявлення та попередження кібератак.

2.2 Аналіз бібліотек для навчання нейронних мереж

Створення моделі ШІ вимагає використання сучасних інструментів та бібліотек для реалізації алгоритмів машинного та глибинного навчання, а також потужного апаратного забезпечення, здатного обробляти великі обсяги даних.

Серед сучасних бібліотек для розробки моделей на основі ШІ виділяються такі популярні інструменти, як TensorFlow, PyTorch та Keras. Кожний з них має свої особливості та функціональні можливості, що дозволяють застосовувати їх для вирішення різних завдань у сфері кібербезпеки.

2.2.1 TensorFlow

TensorFlow – це одна з провідних бібліотек для машинного та глибинного навчання, яка підтримує різноманітний набір алгоритмів,

включаючи методи класифікації, кластеризації, обробки природної мови, а також алгоритми для аналізу зображень, послідовностей і часових рядів. Завдяки своїй високій продуктивності та можливості розподілених обчислень, TensorFlow є ефективним інструментом для розробки систем виявлення загроз у реальному часі, обробки великих наборів даних і моделювання поведінкових патернів [14].

Ця бібліотека дозволяє розробникам реалізувати різні види нейронних мереж – CNN, RNN, а також трансформерні архітектури, що є критичними для обробки великих обсягів даних і детектування загроз. CNN є корисними для аналізу структурованих даних, таких як графи або матриці взаємозв'язків у мережі, що допомагає виявляти підозрілі зв'язки та аномалії. Натомість, RNN ефективні для обробки часових рядів та послідовностей, що дозволяє аналізувати дії користувача, поведінкові патерни чи потоки даних у реальному часі. Трансформери забезпечують глибоку обробку тексту, що є корисним для NLP, наприклад, при аналізі електронних листів для виявлення фішингу [15].

TensorFlow включає бібліотеку TensorFlow Text, яка надає інструменти для роботи з текстовими даними, що дозволяє ефективно виконувати NLP. Ці інструменти можуть бути використані для аналізу даних із соціальних мереж, електронної пошти та звітів про загрози, допомагаючи ідентифікувати фішингові атаки або ознаки соціальної інженерії. Бібліотека також підтримує трансформери, такі як BERT[16], які можуть використовуватись для побудови моделей класифікації тексту з високою точністю. Це дає можливість системі самостійно аналізувати нові текстові загрози, навіть якщо вони відрізняються від попередніх патернів.

Також TensorFlow має вбудовану підтримку розподілених обчислень, що дозволяє одночасно обробляти великі набори даних на декількох процесорах або GPU. Це особливо важливо для систем, що працюють з великим обсягом даних у режимі реального часу, таких як мережеві лог-файли або потоки даних від IoT-пристроїв. Розподілені обчислення

дозволяють скоротити час навчання моделей та підвищити їх продуктивність, забезпечуючи високу швидкість і точність у виявленні загроз.

Вона дозволяє автоматизувати процес аналізу кіберзагроз, об’єднуючи ML та NLP для виявлення нових вразливостей і кіберзагроз, зокрема шляхом автоматичної класифікації та інтерпретації текстових звітів про загрози. Це допомагає виявляти нові атаки та оперативно реагувати на них.

Дана бібліотека дозволяє інтегрувати моделі з системами моніторингу та реагування на інциденти, що забезпечує автоматизовану реакцію на виявлені загрози. Це значно скорочує час реакції на інциденти та допомагає запобігти розвитку атак, мінімізуючи шкоду для інформаційної системи.

Завдяки високій продуктивності та гнучкості, TensorFlow надає інструменти для побудови масштабованих і надійних моделей виявлення загроз. Його інтеграція з іншими технологіями та інструментами для обробки даних дозволяє створювати адаптивні системи, які навчаються самостійно та не лише виявляють відомі загрози, але й здатні адаптуватися до нових кіберзагроз у режимі реального часу.

2.2.2 PyTorch

PyTorch – це потужна бібліотека для роботи з нейронними мережами, що відзначається зручним і зрозумілим інтерфейсом, який робить її ідеальною для дослідницької роботи, швидкого створення прототипів та розробки складних моделей машинного навчання. Вона особливо підходить для створення моделей, що вимагають складних архітектур та адаптивності до змінних умов кіберсередовища. Здатність працювати з різними типами даних, зокрема з графовими структурами і часовими рядами, дозволяє цій бібліотеці застосовуватися для виявлення нових видів загроз, а гнучкі методи оптимізації роблять її потужним інструментом для задач кібербезпеки [17].

PyTorch надає розробникам можливість змінювати архітектуру моделі під час її роботи, що робить її ідеальною для досліджень і тестування нових

ідей. Це особливо важливо у сфері виявлення загроз, де нові типи атак можуть потребувати модифікацій існуючих архітектур. Графові нейронні мережі дозволяють аналізувати складні взаємодії між вузлами у мережах, що допомагає виявляти аномалії або підозрілі з'єднання у мережевому трафіку.

PyTorch підтримує розподілені обчислення та прискорення на GPU, що дозволяє проводити обробку великих наборів даних у реальному часі. Це особливо корисно для задач, де потрібно обробляти великий обсяг мережевого трафіку або логи для виявлення підозрілих дій. Завдяки бібліотеці TorchServe, PyTorch спрощує розгортання моделей у виробничому середовищі, дозволяючи швидко реалізовувати результати експериментів у масштабованих системах моніторингу та захисту.

Ця бібліотека добре інтегрується з численними інструментами для обробки та візуалізації даних, такими як NumPy, Pandas, Scikit-learn, та Matplotlib. Це полегшує процес попередньої обробки даних, аналізу результатів та створення звітів про виявлені загрози. Наприклад, PyTorch можна поєднати з NLP-бібліотекою Hugging Face Transformers для розробки моделей обробки тексту, що може бути використано для аналізу повідомлень або звітів про кіберзагрози.

TorchVision, додаткова бібліотека PyTorch, спрощує роботу з зображеннями та полегшує розробку моделей для обробки візуальної інформації, такої як знімки з камер спостереження чи графічні дані мережевого трафіку. Наприклад, використання моделей для класифікації зображень або аналізу графів може бути корисним для виявлення певних патернів у складних структурах даних.

PyTorch дозволяє створювати моделі для виявлення аномалій у мережевому трафіку за допомогою алгоритмів класифікації, таких як автоенкодери, які здатні знаходити нові загрози за відхиленням від нормальної поведінки.

Використання RNN дозволяє моделювати поведінку користувачів і виявляти підозрілі дії, такі як спроби несанкціонованого доступу до ресурсів

або зловживання привілейованими правами. Це особливо корисно для захисту від внутрішніх загроз.

Завдяки методам навчання з підкріпленням PyTorch може бути використана для створення адаптивних моделей, що навчаються у реальному часі та автоматично вдосконалюють свої параметри, реагуючи на нові типи загроз. Такі системи навчаються самостійно та є особливо корисними в умовах швидко змінюваного кіберсередовища.

PyTorch продовжує розширювати свої можливості, зберігаючи високу гнучкість та інтегруючись з новими інструментами та архітектурами, що дозволяє створювати високопродуктивні моделі для виявлення кіберзагроз.

2.2.3 Keras

Keras – це багаторівнева бібліотека для роботи з нейронними мережами, яка використовує TensorFlow як основний бекенд, а її інтуїтивний і простий у використанні інтерфейс робить процес розробки моделей швидким і зручним. Це особливо корисно для побудови прототипів моделей кібербезпеки, де потрібні висока швидкість експериментів і можливість тестування різних архітектур. Keras добре підходить для задач, пов’язаних із базовою класифікацією, аналізом аномалій, обробкою природної мови та кластеризацією даних.

Ця бібліотека надає можливість швидко і просто створювати нейронні мережі, що робить її ідеальним вибором для дослідників та інженерів без великого досвіду в програмуванні. Інтуїтивний інтерфейс дозволяє розробляти як прості, так і складні моделі лише з кількома рядками коду, що прискорює процес створення прототипів та тестування різних архітектур для виявлення загроз. Це особливо важливо у середовищах, де швидкість розробки та гнучкість архітектури є ключовими факторами [18].

Keras має вбудовані засоби для попередньої обробки даних, такі як нормалізація, масштабування та токенізація, що полегшує підготовку даних

для моделювання. Це зручно для роботи з різнорідними наборами даних, наприклад, текстовими даними у завданнях обробки природної мови (NLP) або зображеннями для аналізу мережевих схем і візуалізації трафіку. Інструменти для роботи з текстовими даними у Keras дозволяють з легкістю створювати моделі для аналізу електронної пошти або соціальних мереж з метою виявлення фішингу чи інших кіберзагроз.

Дана бібліотека забезпечує гнучкість у налаштуванні гіперпараметрів, таких як швидкість навчання, розмір партії та кількість епох, що дає можливість швидко знаходити оптимальні значення для конкретної моделі.

Завдяки інтеграції з TensorFlow Extended, Keras може використовуватися для розробки моделей кібербезпеки, які легко розгортаються у виробничих середовищах. Це дозволяє будувати повний конвеєр машинного навчання від підготовки даних до обслуговування моделі. Крім того, завдяки інтеграції з бібліотеками, такими як Scikit-learn для класичних методів ML та Matplotlib для візуалізації, Keras може слугувати базою для більш комплексних досліджень.

Завдяки функціям NLP, Keras може бути використаний для створення моделей, що автоматично класифікують повідомлення як «фішингові» або «безпечні» на основі їх вмісту. Це спрощує виявлення спроб соціальної інженерії в електронних листах та інших текстових повідомленнях.

Keras продовжує залишатися надійним і простим у використанні інструментом для розробки моделей кібербезпеки, дозволяючи швидко адаптуватися до змінюваного ландшафту загроз і забезпечувати гнучкість та ефективність у розробці рішень для захисту інформаційних систем.

Проаналізувавши дані бібліотеки, для створення ШІ була обрана TensorFlow у поєднанні із Keras, бо їх команди дозволяють дуже гнучкими методами налаштувати нейронну мережу, що буде працювати із записами про активність мережі. Також під час навчання можна легко інтегрувати будь-який датасет.

2.3 Апаратне забезпечення для реалізації ШІ

Розробка та тестування моделей на основі ШІ потребує потужних обчислювальних ресурсів, особливо коли мова йде про великі обсяги даних або складні глибинні нейронні мережі. Основні вимоги до апаратного забезпечення включають наявність графічних процесорів (GPU) та центральних процесорів (CPU) з високою продуктивністю, достатнього обсягу оперативної пам’яті та сховища даних [19].

GPU від NVIDIA, серії Tesla, A100 або V100, широко використовуються для обчислень у ШІ завдяки їх здатності паралельно обробляти дані, що дозволяє значно прискорити навчання моделей. Вони підходять для обробки великих наборів даних, необхідних для виявлення і класифікації загроз. Також, TensorFlow і PyTorch мають спеціальні механізми оптимізації для роботи з GPU, що робить їх особливо ефективними для побудови великих моделей.

CPU, наприклад сучасні багатоядерні процесори Intel Xeon або AMD EPYC, також використовуються для обчислень у ШІ, особливо для початкового аналізу даних та тренування моделей невеликого масштабу. Вони є незамінними для завдань, що не потребують масивних обчислень, але вимагають стабільності та надійності.

Оперативна пам'ять (RAM) потрібна для обробки великих обсягів даних та навчання моделей ШІ у дуже значних обсягах. Рекомендовано використовувати системи з не менш ніж 32 ГБ оперативної пам'яті, а для більш складних обчислень – від 64 ГБ і вище, особливо якщо планується аналіз великих текстових даних або журналів.

Також обов’язково потрібно велике сховище даних. Оскільки моделі для кібербезпеки часто потребують доступу до великих обсягів даних, таких як логи, записи мережевого трафіку та інші журнали, використання SSD-накопичувачів є доцільним для забезпечення швидкого доступу до даних.

Для проектів із високими вимогами до обчислювальних ресурсів також можна використовувати хмарні платформи, такі як Google Cloud Platform, Amazon Web Services та Microsoft Azure, які надають можливість використовувати віртуальні машини з потужними GPU або TPU, доступ до великих обсягів пам'яті та спеціалізоване ПЗ для роботи із ШІ.

Оптимальне поєднання програмних бібліотек та відповідного апаратного забезпечення дозволяє побудувати адаптивну та ефективну модель загроз для ІС. Вибір конкретних інструментів залежить від типу завдань, обсягу даних та фінансових можливостей проекту. Бібліотеки, такі як TensorFlow та PyTorch, у поєднанні з GPU або TPU, забезпечують високу продуктивність і дозволяють створювати складні моделі, здатні ефективно виявляти та попереджувати кібератаки, а використання хмарних платформ спрощує масштабування ресурсів.

2.4 Системи виявлення вторгнень

2.4.1 Snort

Snort – це відкрита та потужна система для виявлення та запобігання вторгнень у комп'ютерних мережах, яка широко використовується для моніторингу трафіку та аналізу шкідливої активності. Вона працює за принципом аналізу вхідних і вихідних даних у реальному часі, порівнюючи їх із набором визначених правил або сигнатур атак. Завдяки цьому Snort здатен виявляти та сигналізувати про підозрілу активність, як від спроби зловмисників проникнути в систему, так і виконуючи постійне сканування портів [20].

Основною перевагою Snort є його гнучкість. Система може працювати в кількох режимах: як простий сніффер, що відображає поточний мережевий трафік, як журнал, який записує дані для подальшого аналізу, або як система виявлення вторгнень, яка аналізує мережеву активність за визначеними

правилами. Якщо інтегрувати Snort з іншими компонентами мережевої інфраструктури, він може виконувати роль системи запобігання вторгнень, блокуючи небажаний трафік у реальному часі.

Одним із ключових аспектів Snort є використання правил. Кожне правило визначає шаблон або умову, за якою система ідентифікує загрозу. Це можуть бути певні IP-адреси, порти, протоколи чи специфічні шаблони вмісту даних. Правила можна налаштовувати відповідно до потреб користувача, а також використовувати готові набори, які регулярно оновлюються спільнотою Snort для врахування нових типів атак.

Snort підтримує широкий спектр протоколів і може аналізувати їх на предмет аномалій або порушень. Система фіксує кожен випадок підозрілої активності у вигляді журналу чи сповіщення, що містить детальну інформацію про подію: IP-адреси джерела й призначення, використовувані порти, тип атаки, час події тощо. Ці дані можна інтегрувати з іншими інструментами, такими як SIEM-системи, для глибшого аналізу.

Snort є безкоштовним інструментом, тому він доступний як для індивідуальних користувачів, так і для великих організацій. Завдяки великій спільноті користувачів і розробників Snort постійно вдосконалюється, отримуючи нові функції та актуалізовані бази правил. Хоча він може бути складним у налаштуванні та потребувати значних ресурсів для обробки великого обсягу трафіку, його універсальність і ефективність роблять Snort одним із найпопулярніших рішень у сфері кібербезпеки.

2.4.2 Suricata

Suricata – це сучасна багатофункціональна система для виявлення та запобігання вторгнень (IDS/IPS), яка також може виконувати роль мережевого аналізатора трафіку. Вона була створена для забезпечення високої продуктивності й масштабованості, а також для підтримки сучасних мережевих стандартів і протоколів. Розроблена як проєкт з відкритим кодом,

Suricata отримала широке визнання завдяки своїй гнучкості, швидкості роботи та здатності аналізувати великий обсяг даних у реальному часі [21].

Особливістю Suricata є її багатопотокова архітектура. Це означає, що система може ефективно використовувати ресурси багатоядерних процесорів для аналізу великого потоку даних, розподіляючи навантаження між потоками. Завдяки цьому вона здатна забезпечувати високу швидкість обробки навіть у великих мережах із насиченим трафіком. Suricata підтримує сучасні апаратні технології, такі як Intel QuickAssist і GPU-прискорення, що ще більше підвищує її продуктивність.

Однією з головних переваг Suricata є її здатність працювати з багатьма типами даних одночасно. Вона не лише аналізує заголовки пакетів, але й виконує глибокий аналіз вмісту, що дозволяє виявляти загрози на рівні прикладних протоколів. Suricata підтримує такі популярні протоколи, як HTTP, DNS, TLS, FTP і SMB, і може виділяти з них ключову інформацію, що є корисною для детекції складних атак.

Suricata використовує правила, які сумісні зі Snort, що дозволяє легко інтегрувати існуючі набори сигнатур. Однак вона йде далі, пропонуючи розширені можливості аналізу за допомогою скриптів на основі мови Lua. Це дає змогу створювати більш гнучкі та адаптивні сценарії для виявлення складних загроз, таких як багатокрокові атаки або аномалії у поведінці мережевих пристроїв.

Ще однією ключовою особливістю Suricata є підтримка потокового аналізу. Вона здатна відстежувати окремі з'єднання, збираючи всі дані в контексті потоку. Це особливо важливо для виявлення атак, які можуть маскуватися під легітимний трафік або виконуватися поступово.

Suricata також інтегрується з багатьма інструментами кібербезпеки та мережевого моніторингу. Вона може працювати як джерело даних для SIEM-систем, таких як Elasticsearch і Splunk, або використовуватися разом із платформами для візуалізації, наприклад, Kibana. Крім того, вона забезпечує

можливість зберігання захопленого трафіку у форматі pcap для подальшого аналізу.

Suricata активно підтримується спільнотою Open Information Security Foundation, яка регулярно випускає оновлення та покращення. Завдяки цьому система завжди залишається актуальною та готовою до протидії новітнім загрозам. У підсумку, Suricata є потужним інструментом для забезпечення безпеки сучасних мереж, пропонуючи не лише високоефективний аналіз, а й гнучкість для адаптації до індивідуальних потреб користувача чи організації.

Після аналізу даних систем було зроблено висновок, що у даній роботі буде використана система Snort, бо вона є простою у порівнянні із Suricata, що дозволить зробити проект швидше та якісніше, бо буде витрачено менше часу на вивчення програмного синтаксису. Також Snort зможе краще взаємодіяти із нейронною мережею для побудови моделі захисту ІС, так як він зможе використовуватись як журнал активності, створюючи потрібні для аналізу логи.

3 ДОСЛІДЖЕННЯ РОБОТИ СИСТЕМИ SNORT ІЗ ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ

3.1 Технічні характеристики середовища для розробки моделі загроз

Виходячи з розглянутої раніше інформації, було вирішено обрати багатошарову персептронну нейронну мережу. Цей тип ШІ належить до класу мереж глибокого навчання та використовується для завдань класифікації, регресії та інших завдань машинного навчання, саме чому й підходить під виконання поставленої задачі.

Для побудови даної моделі був обраний ноутбук із GPU компанії NVIDIA, а саме Geforce RTX 2050 із графічною пам’яттю 4 гігабайти та CPU компанії Intel, а саме Core i5-12450H із частотою 2 Ггц. Щодо операційної пам’яті було зроблено висновок, що 16 гігабайт для створення прототипу буде достатньо, але для створення повноцінної мережі для захисту буде потрібно більший обсяг. У якості носія інформації був використаний SSD накопичувач компанії KINGSTON із форм-фактором M.2 ємністю в 1 терабайт. У якості операційної системи була обрана Windows 10 pro.

Програмне забезпечення, що використовувалось під час роботи над проектом, містила в собі середовище Visual Studio Code із встановленим заздалегідь пакетом Python та бібліотеками TensorFlow та Keras. Саме цей набір був обраний тому що дозволив створити ШІ найшвидшими та відносно легшими методами на відміну від інших.

Для того, щоб нейронній мережі було із чим працювати, була встановлена система для виявлення та запобігання вторгнень Snort. Так як вона була розроблена під дистрибутиви Linux знадобилося додатково встановити NPcap, що є новою версією програмного засобу WinPcap, для встановлення бібліотеки із спеціалізованими пакетами Windows. Snort був обраний, так як він дозволяє гнучкими методами налаштувати правила для

сканування трафіку та створити зручні логи для зчитування нейронною мережею для їх сумісної роботи.

3.2 Методика побудови захисту

У цьому проекті для захисту інформаційної системи буде використано підхід, що поєднує традиційні методи виявлення вторгнень (IDS) із сучасними технологіями штучного інтелекту. Основною метою є створення моделі, здатної аналізувати мережевий трафік і розпізнавати потенційні загрози в режимі реального часу. Методика реалізації базується на навчанні нейронної мережі на історичних даних атак, інтеграції цієї моделі з популярною системою IDS Snort і забезпеченні автоматичного моніторингу активності у мережі.

Для даного проекту було обрано датасет NSL-KDD, який є одним із стандартів у сфері кібербезпеки. Він включає опис різних типів мережевих атак і нормальної активності. Ці дані містять численні особливості мережевих з'єднань, такі як тип протоколу, кількість байтів, прапори TCP тощо. Усі дані структуровано, що дозволяє їх використовувати для навчання моделі.

Дані потрібно попередньо обробляти: числові значення нормалізувати для забезпечення рівномірного масштабу, а категорії ознак, такі як тип протоколу, треба перетворювати в числовий формат за допомогою техніки «one-hot encoding». Це дозволить зробити всі ознаки придатними для обробки нейронною мережею. Крім того, дані будуть розділені на тренувальні й тестові підмножини для оцінки продуктивності моделі.

Для класифікації трафіку було обрано багатошаровий персептрон. Це тип штучної нейронної мережі, який добре підходить для задач класифікації з багатьма вхідними ознаками. Модель включала кілька прихованих шарів із функціями активації ReLU для обробки нелінійних залежностей між ознаками. Вихідний шар має сигмоідальну активацію, що забезпечує зручний

формат для бінарної класифікації на те атака це чи нормальний трафік. Для уникнення перенавчання будуть застосовані Dropout-шари, які випадково «відключають» частину нейронів під час тренування.

Навчання моделі здійснюватиметься на основі функції втрат binary_crossentropy, яка є стандартом для бінарної класифікації. У процесі оптимізації буде використаний алгоритм Adam, що забезпечує швидке та стабільне порівняння схожих ознак. Після завершення тренування модель буде оцінюватися на тестових даних, де буде обчислена точність, повнота, AUC (також відома як крива похибок або ROC-крива) та інші метрики, що демонструють її здатність ефективно розпізнавати атаки.

Готову модель буде інтегровано в систему Snort, яка виступає основною платформою для аналізу мережевого трафіку. Snort записує підозрілі події у файл журналу alerts.txt. Для автоматизації буде реалізовано скрипт на Python, який постійно зчитує нові записи з цього журналу, витягує необхідні ознаки та передає їх до нейронної мережі для аналізу. Якщо модель виявляє атаку, система видаватиме попередження, яке може бути доповнене додатковими діями, наприклад, сповіщенням адміністратора.

Цей підхід дозволить поєднати переваги системи Snort для обробки реального трафіку з потужністю штучного інтелекту для більш точного виявлення загроз. Система зможе працювати автономно, аналізуючи нові події в режимі реального часу, і при цьому залишатися достатньо гнучкою для адаптації до нових типів атак.

3.3 Розробка моделі загроз

Модель загроз, що розробляється може бути використана у будь-яких комп’ютерних системах Windows. Вона здатна визначати актуальні загрози на транспортному та мережевому рівні.

Розробка моделі загроз із використанням Snort та нейронної мережі складалася з кількох взаємопов'язаних етапів, які забезпечують інтеграцію аналізу трафіку та сучасних методів штучного інтелекту.

Спочатку була проведена підготовка і налаштування системи Snort. Це включало завантаження та встановлення програмного забезпечення на платформу Windows 10. Так як система Snort була розроблена під дистрибутиви Linux, знадобилося встановити бібліотеку NPcap, що надає Snort можливість працювати із пакетами Windows. Після цього було виконано налаштування базової конфігурації. У конфігураційному файлі snort.conf були задані ключові параметри, такі як адреси локальної мережі, шляхи до журналів та правил (рисунок 4.1).

```
100
101     # Path to your rules files (this can be a relative path)
102     # Note for Windows users:  You are advised to make this an absolute path,
103     # such as:  c:\snort\rules
104     var RULE_PATH c:\Snort\rules
105     var SO_RULE_PATH ../so_rules
106     var PREPROC_RULE_PATH c:\Snort\preproc_rules
107
108     # If you are using reputation preprocessor set these
109     # Currently there is a bug with relative paths, they are relative to where snort is
110     # not relative to snort.conf like the above variables
111     # This is completely inconsistent with how other vars work, BUG 89986
112     # Set the absolute path appropriately
113     var WHITE_LIST_PATH c:\Snort\rules
114     var BLACK_LIST_PATH c:\Snort\rules
115
```

Рисунок 3.1 – Шляхи до файлів із правилами у snort.conf

Крім того, були додані індивідуальні правила (рисунок 4.2), які дозволяли фільтрувати певний тип трафіку або виявляти підозрілі дії, наприклад сканування портів чи спроби підключення до заборонених ресурсів.

```
#-------------
# LOCAL RULES
#-------------

alert tcp any any -> any any (msg:"High traffic detected"; dsize:>1500; sid:1000002; rev:1;)

alert icmp any any -> any any (msg:"ICMP Ping detected"; itype:8; sid:1000003; rev:1;)

alert tcp any any -> 192.168.1.10 80 (msg:"Access to forbidden site"; sid:1000004; rev:1;)

alert tcp any any -> any 23 (msg:"Telnet access detected"; sid:1000005; rev:1;)

```

Рисунок 3.2 – Індивідуальні правила фільтрації трафіку

Після налаштування Snort потрібно було створити нейронну мережу для аналізу загроз. Для побудови моделі були використані бібліотеки TensorFlow та Keras, яка, у результаті, отримувала вхідні дані у вигляді мережевих характеристик, таких як IP-адреси, порти чи протоколи. Скрипт для навчання нейронної мережі можна побачити у лістингу Б.1.

Архітектура мережі включала 3 шари: 2 вхідні шари із функціями активації ReLU та Dropout для запобігання перенавчанню, а також вихідний шар із сигмоідальною активацією для класифікації, чи є трафік загрозливим (рисунок 4.3).

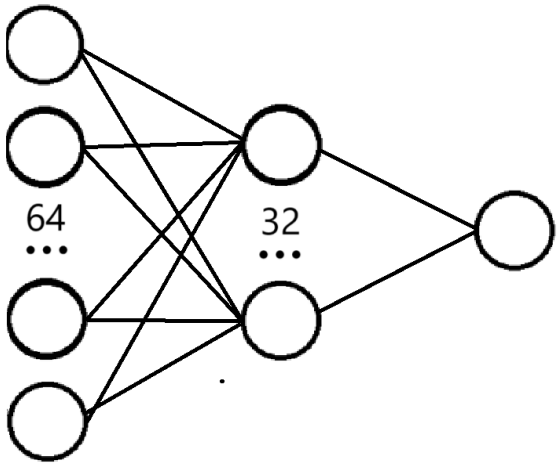

Рисунок 3.3 – Схема шарів розробленого ШІ

Навчання проводилося на датасеті NSL-KDD, який містив 123 записи про нормальні та небезпечні дії у мережі (рисунок 4.4). Попередня обробка даних передбачала масштабування числових значень, кодування категоріальних ознак і розділення вибірки на навчальну та тестову. Збережену модель було експортовано у файл attack_detection_model.h5 для подальшого використання.

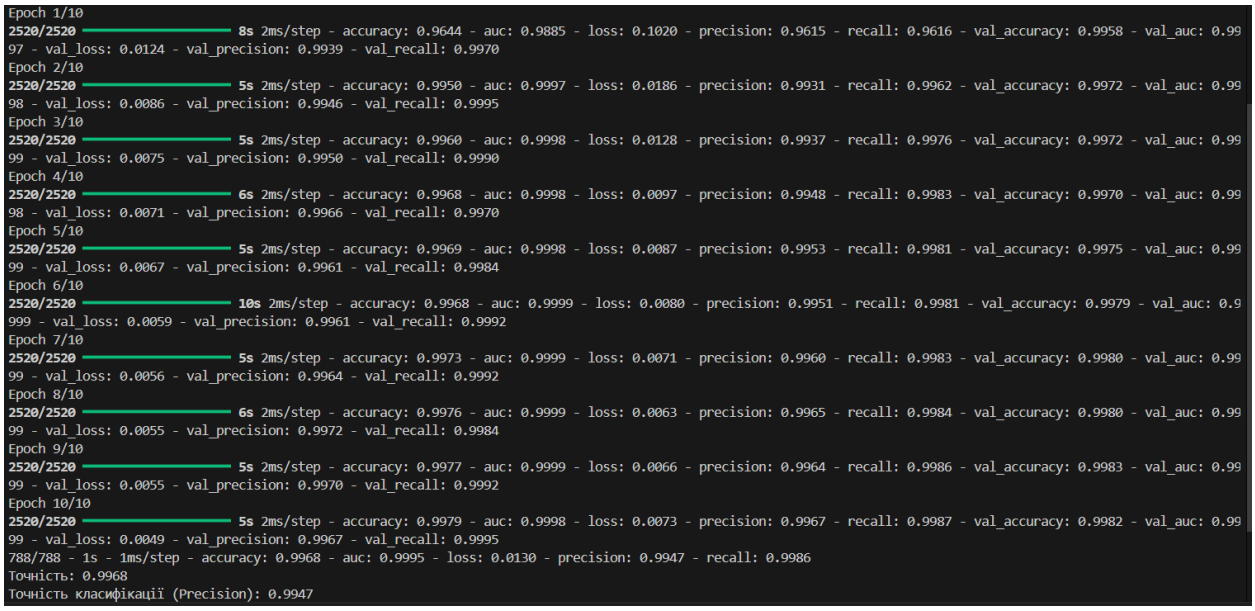

Рисунок 3.4 – Візуалізація процесу навчання нейронної мережі

Інтеграцію Snort із нейронною мережею було організовано через аналіз журналів Snort у реальному часі. Для цього розроблено скрипт на Python, який читає файл alerts.txt (рисунок 4.5), обробляє нові записи та витягує ключові ознаки. Скрипт для аналізу записів про загрози можна побачити у лістингу Б.2.

```
[**] [1:1000001:0] ET MALWARE Possible Malware Activity - Suspicious DNS Query [**]
[Classification: Trojans and Backdoors] [Priority: 2]
01/11-14:20:03.123456 192.168.1.105 -> 8.8.8.8
TCP TTL:64 TOS:0x0 ID:12345 IpLen:20 DgmLen:48 DF
Proto:UDP SrcPort:53 DstPort:53 Len:28

[**] [1:1000001:1] ICMP ping request [**]
[Classification: Attempted Information Leak] [Priority: 3]
12/30-14:23:45.456789 192.168.1.5 -> 192.168.1.10
ICMP TTL:64 TOS:0x0 ID:56789 IpLen:20 DgmLen:56
Type:8  Code:0  ID:100  Seq:1  ECHO
```

Рисунок 3.5 – Записи у alerts.txt

Ці ознаки перетворюються у формат, придатний для аналізу нейронною мережею. Завантажена модель аналізує кожен запис і передбачає, чи є він загрозою. У разі виявлення атаки система виводить попередження, що дозволяє швидко реагувати на загрози.

```
Зчитано 11 нових рядків.
IP: None -> None, Ports: None -> None
Витягнуті ознаки: [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step
Передбачення для рядка: 0.9988047480583191
!!! Виявлено атаку: [**] [1:1000001:0] ET MALWARE Possible Malware Activity - Suspicious DNS Query [**]
IP: None -> None, Ports: None -> None
Витягнуті ознаки: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
Передбачення для рядка: 0.5829297304153442
!!! Виявлено атаку: [Classification: Trojans and Backdoors] [Priority: 2]
IP: 192.168.1.105 -> 8.8.8.8, Ports: None -> None
Витягнуті ознаки: [192.0, 168.0, 8.0, 8.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
Передбачення для рядка: 1.0
!!! Виявлено атаку: 01/11-14:20:03.123456 192.168.1.105 -> 8.8.8.8
IP: None -> None, Ports: None -> None
Витягнуті ознаки: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Передбачення для рядка: 0.5829297304153442
!!! Виявлено атаку: TCP TTL:64 TOS:0x0 ID:12345 IpLen:20 DgmLen:48 DF
IP: None -> None, Ports: 53 -> 53
Витягнуті ознаки: [0, 0, 53.0, 53.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
```

Рисунок 3.6 – Результат сканування записів з alerts.txt

На завершальному етапі модель проходила тестування на реальних даних. Досліджувалася її здатність виявляти як відомі, так і нові загрози. Приклади готових записів про шкідливі атаки, так само як й дата сет NSL-KDD, були взяті з порталу Kaggle [23]. За результатами тестування

вдосконалювалася як модель нейронної мережі, так і правила Snort для зменшення помилкових спрацьовувань.

Таким чином, модель загроз була побудована як інтегроване рішення, що поєднує можливості традиційного аналізу трафіку Snort і сучасних технологій штучного інтелекту для покращення точності виявлення та швидкості реагування на загрози.

Завдяки великій кількості ознак при навчанні, ця модель загроз працює водночас на транспортному та мережевому рівні, що дозволяє охопити великий спектр використання ІС.

Основним недоліком є налаштування метрик зчитування власноруч, тобто розробнику потрібно обирати які саме метрики буде зчитувати ШІ та вписувати їх у скрипт для аналізу трафіку.

З можливостей покращення можна виділити більш дружній інтерфейс, бо у цій версії все виводиться у консолях та текстових файлах. Також дана система не має механізму самонавчання, але можливість додати це присутня.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено модель захисту для виявлення та попередження вторгнень.

У результаті було виконано наступні дії:

- було проаналізовано можливі рішення для створення ефективнішої моделі загроз;
- було обрано систему для інтеграції із ШІ;
- було налаштовано систему Snort під аналіз мережевого трафіку та побудову записів про підозрілу активність;
- було написано програму для навчання нейронної мережі на основі датасету NSL-KDD із 123 записами про нормальні та небезпечні дії у мережі;
- було написано скрипт для інтеграції ШІ у систему Snort для аналізу записів про дії у мережі, виявлення та передбачення вторгнень.

Розроблена модель загроз передбачає постійну роботу зі сканування записів, створених системою Snort та, у разі виникнення підозрілих дій у мережі, швидкого реагування із попередженням користувача про небезпеку.

Модель не є ідеальною, тому може бути покращена. Із можливих покращень можна виділити:

- покращення дружнього інтерфейсу, замість консолей та текстових файлів;
- додавання самонавчання до моделі ШІ.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Проценко А.С., Федорченко В.М., «Using artificial intelligence to analyze security threats in information systems». Тези доповідей сьомої міжнародної науково-технічної конференції COMPUTER AND INFORMATIONAL SYSTEMS AND TECHNOLOGIES – Харків: ХНУРЕ; Рига: ISMA University; Київ: НАУ; Київ: Інститут реєстрації інформації НАНУ; Львів: національний університет «Львівська Політехніка»; Баку: Національний університет оборони, вересень 26-27, с. 43-45, 2024. URL: http://csitic.com/images/data/CSITIC.all_2024ENG.pdf – 01.11.2024.

2. F. De Rosa, N. Maunero, P. Prinetto, F. Talentino and M. Trussoni, «ThreMA: Ontology-Based Automated Threat Modeling for ICT Infrastructures,» vol. 10, pp. 116514-116526, 2022, URL: https://ieeexplore.ieee.org/ stamp/stamp.jsp?tp=&arnumber=9936611&isnumber=9668973 – 01.11.2024.

3. KONEV, Anton, et al. A survey on threat-modeling techniques: protected objects and classification of threats. *Symmetry*, 2022, vol 14.3, pp. 549. URL: https://www.mdpi.com/2073-8994/14/3/549 – 01.11.2024.

4. P. Lavanya, H. Anila Glory and V. S. Shankar Sriram, «Mitigating Insider Threat: A Neural Network Approach for Enhanced Security,» vol. 12, pp. 73752-73768, 2024, URL: https://ieeexplore.ieee.org/stamp/ stamp. jsp?tp=&arnumber=10538122 – 01.11.2024.

5. A. Batool and Y. -C. Byun, «An Ensemble Architecture Based on Deep Learning Model for Click Fraud Detection in Pay-Per-Click Advertisement Campaign,» vol. 10, pp. 113410-113426, 2022, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9908561 – 01.11.2024.

6. M. M. Asiri, H. Alfraihi, Y. Said, K. M. Othman, A. S. Salama and R. Marzouk, «Securing Consumer Electronics Devices: A Blockchain-Based Access Management Approach Enhanced by Deep Learning Threat Modeling for IoT Ecosystems,» vol. 12, pp. 110671-110680, 2024, URL:

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10632117 – 01.11.2024.

7. OWASP Threat Dragon. URL: https://owasp.org/www-project-threat-dragon/ – 01.11.2024.

8. Threat Modeling. URL: https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling – 01.11.2024.

9. W. Choi, S. Pandey and J. Kim, «Detecting Cybersecurity Threats for Industrial Control Systems Using Machine Learning,» vol. 12, pp. 153550-153563, 2024, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10714344 – 01.11.2024.

10. S. V. Mahadevkar, B. Khemani, S. Patil, K. Kotecha, D. R. Vora, A. Abraham, «A Review on Machine Learning Styles in Computer Vision – Techniques and Future Directions,» vol. 10, pp. 107293 - 107329, 2022, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9903420 – 01.11.2024.

11. H. Ali *et al*., «A Survey on Attacks and Their Countermeasures in Deep Learning: Applications in Deep Neural Networks, Federated, Transfer, and Deep Reinforcement Learning,» vol. 11, pp. 120095-120130, 2023, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10288459 – 01.11.2024.

12. M. Izadi and M. N. Ahmadabadi, «On the Evaluation of NLP-based Models for Software Engineering,» *2022 IEEE/ACM 1st International Workshop on Natural Language-Based Software Engineering*, pp. 48-50, 2022 URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9808680 – 01.11.2024.

13. G. Ioannou, P. Louvieris and N. Clewley, «A Markov Multi-Phase Transferable Belief Model for Cyber Situational Awareness,» vol. 7, pp. 39305-39320, 2019, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8636494 – 01.11.2024.

14. T. J. Sheng *et al*., «An Internet of Things Based Smart Waste Management System Using LoRa and Tensorflow Deep Learning Model,» vol. 8,

pp. 148793-148811, 2020, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9165744 – 01.11.2024.

15. S. Kumbale, S. Singh, G. Poornalatha and S. Singh, «BREE-HD: A Transformer-Based Model to Identify Threats on Twitter,» vol. 11, pp. 67180 - 67190, 2023, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10168907 – 01.11.2024.

16. M. A. Ferrag *et al.*, «Revolutionizing Cyber Threat Detection With Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices,» vol. 12, pp. 23733-23750, 2024, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10423646 – 01.11.2024.

17. Adam Paszke et al., «PyTorch: An Imperative Style, High-Performance Deep Learning Library» / 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, pp. 1-12, 2019, URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf – 01.11.2024.

18. M. Si, J. Tarnoczi, and M. Wiens, «Development of Predictive Emissions Monitoring System Using Open-Source Machine Learning Library – Keras: A Case Study on a Cogeneration Unit,» vol. 7, pp. 113463-113475, 2019, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8771122 – 01.11.2024.

19. T. Speith, J. Speith, S. Becker, Y. Zou, A. Biega and C. Paar, «Explainability as a Requirement for Hardware: Introducing Explainable Hardware (XHW),» *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, Reykjavik, Iceland, pp. 354-362, 2024, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10628479 – 01.11.2024.

20. Snort. URL: https://www.snort.org – 04.01.2025.

21. Suricata. URL: https://suricata.io – 04.01.2025.

22. Kaggle. URL: https://www.kaggle.com/datasets – 04.01.2025.

23. Методичні вказівки до організації виконання та захисту кваліфікаційних робіт другого (магістерського) рівня вищої освіти, URL: https://drive.google.com/drive/folders/1Ezfs1zDAH73pRANGzHhWWqvwf-Jl0kZm – 16.01.2025.