ДВНЗ «ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»

Факультет комп'ютерно-інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету)

Кафедра прикладної математики і інформатики

(повна назва кафедри)

«До захисту допущено»
Завідувач кафедри прикладної математики і інформатики
_____ Наталія МАСЛОВА
(підпис) (ім'я та прізвище)
"___"_____ 2023 р.

Випускна кваліфікаційна робота

магістра

(освітньо-кваліфікаційний рівень)

на тему «Дослідження та застосування технології штучного інтелекту для забезпечення захисту індустріальної мережі»

Виконав студент 2 курсу, групи ІПЗм–22

(шифр групи)

спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Садовніченко Максим Віталійович _____

(Прізвище, Ім'я та По-батькові) (підпис)

Керівник професор каф. ПМІ, д. ф.-м. н. Володимир МАСОЛ _____

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Рецензент доцент каф. АТ, к.т.н. Анна ВОРОПАЄВА _____

(посада науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Рецензент доцент каф. ЕТ, к.т.н. Віталій ШАМАЄВ _____

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Засвідчую, що у цій випускній кваліфікаційній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

Луцьк – 2023 р.

ДВНЗ "Донецький національний технічний університет"

Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики і інформатики
Освітній ступінь магістр
Напрям підготовки (спеціальність) 121 Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Завідувача кафедри ПМІ
Наталія МАСЛОВА
"___" _____2023 року

З А В Д А Н Н Я

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Садовніченко Максим Віталійович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) «Дослідження та застосування технології штучного інтелекту для забезпечення захисту індустріальної мережі»
керівник проекту (роботи) Масол В. І. професор, доктор фіз.-мат. наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом від 22 вересня року № 478
2. Строк подання студентом проекту (роботи) 5 грудня 2023 року
3. Вихідні дані до проекту (роботи)
Технічна документація та матеріали з переддипломної практики

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз сучасного стану індустріальної мережі
2) Аналіз сучаснх мережевих загроз
3) Аналіз технології штучного інтелекту
4) Виявлення DDoS за допомогою нейронної мережі

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

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Назарова І.А.		

7. Дата видачі завдання 2 жовтня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз сучасного стану індустріальної мережі	02.10.23 – 16.10.23	
2	Аналіз сучаснх мережевих загроз	16.10.23 – 30.10.23	
3	Аналіз технології штучного інтелекту	30.10.23 – 13.11.23	
4	Виявлення DDoS за допомогою нейронної мережі	13.11.23 – 04.12.23	
5	Оформлення пояснювальної записки.	04.12.23 – 10.12.23	
6	Представлення дипломної роботи науковому керівнику	11.12.23	
7	Захист дипломної роботи	26.12.23	

Студент **Максим САДОВНІЧЕНКО**
(підпис) (прізвище та ініціали)

Керівник роботи _____ **Володимир МАСОЛ**
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Садовніченко Максим Віталійович. Дослідження та застосування технології штучного інтелекту для забезпечення захисту індустріальної мережі / Випускна кваліфікаційна робота на здобуття освітнього ступеня «магістр» зі спеціальності 121 Інженерія програмного забезпечення – ДВНЗ «ДонНТУ», Луцьк, 2023.

Пояснювальна записка: 83 стор., 25 рис., 1 табл., 23 посилання.

Дипломна робота присвячена дослідженню та застосуванню технології штучного інтелекту для підвищення рівня захисту індустріальної мережі. Робота включає в себе аналіз сучасних методів захисту індустріальних мереж, виявлення їхніх переваг та недоліків. Можливості використання системи виявлення вторгнення на основі LSTM RNN моделі для виявлення аномалій в мережевому трафіку.

Ключові слова: Industry 4.0, штучний інтелект, нейроні мережі, LSTM, RNN, Python, TensorFlow.

ABSTRACT

Sadovnichenko Maksym Vitaliiovych. Research and application of artificial intelligence technology to ensure industrial network protection / Final qualification work for the degree of Master's Degree in specialty 121 Software Engineering - SHEI "DonNTU", Lutsk, 2023.

Explanatory note: 83 pages, 25 figures, 1 tables, 23 references.

The thesis is devoted to the research and application of artificial intelligence technology to improve the level of protection of the industrial network. The work includes an analysis of modern methods of protecting industrial networks, identifying their advantages and disadvantages. Possibilities of using an intrusion detection system based on the LSTM RNN model to detect anomalies in network traffic.

Keywords: Industry 4.0, artificial intelligence, neural networks, LSTM, RNN, Python, TensorFlow.

ЗМІСТ

Перелік умовних позначень, скорочень і термінів 8
Вступ 9
Розділ 1 Аналіз сучасного стану індустріальної мережі 11
1.1 Industry 4.0 11
1.2 SCADA 13
1.3 Industrial Internet of Things 15
1.4 Великі дані та аналітика даних 18
1.5 Висновки за розділом 20
Розділ 2 Аналіз сучасних мережевих загроз 22
2.1 Мережевий стек інтернет протоколу 22
2.2 Класифікація мережевих загроз 25
2.3 Проблема безпеки в мережі Industry 4.0 26
2.4 DDoS атаки та їх вплив на доступність системи 29
2.5 Заходи забезпечення безпеки мереж 30
2.6 Висновки за розділом 32
Розділ 3 Аналіз технології штучного інтелекту 34
3.1 Теорія машинного навчання 34
3.2 Штучні нейронні мережі 37
3.2.1 Глибині нейронні мережі 39
3.2.2 CNN 41
3.2.3 RNN 45
3.3 LSTM модель 47
3.4 Навчальні мережі 51
3.5 Використання нейронних мереж для виявлення загроз 54
3.6 Висновки за розділом 56
Розділ 4 Виявлення DDoS атаки за допомогою нейронної мережі 58
4.1 Підготовка моделі 58
4.2 Підготовка і обробка даних 59

4.3 Аналіз пакетів 61
4.4 Результати навчання 64
4.5 Висновки за розділом 66
Висновки 68
Список використаних джерел 70
Додаток А Зауваження нормоконтролера 73
Додаток Б Лістинг навчальної моделі 74
Додаток В Лістинг програми виявлення загрози 79

Перелік умовних позначень, скорочень і термінів

CAD — Computer-aided design

CNN — Convolutional neural network

DDoS — Denial-of-service attack

DNN — Deep neural network

HMI — Human Machine Interface

IIoT — Industrial Internet of Things

IP — Internet protocol

LSTM — Long short-term memory

ML — Machine learning

MTU — Master Terminal Units

QoS — Quality of service

RNN — Recurrent neural network

RTU — Remote Terminal Unit

SCADA — Supervisory Control and Data Acquisition

SGD — Stochastic gradient descent

ДМЗ — Демілітаризована зона

ШНМ — Штучна нейрона мережа

ВСТУП

Актуальність роботи. Полягає в тому, що індустріальні мережі, що використовуються в промисловості, стають об'єктом зростаючого інтересу для кіберзлочинців та кіберзагроз. З розвитком технологій та інтернету речей (IoT), ці мережі стають все більш підключеними та вразливими перед кібератаками.

Індустріальні системи контролю стають об'єктом спеціалізованих атак, що може призвести до серйозних наслідків, включаючи простірні та техногенні аварії, порушення виробничих процесів та втрату конфіденційної інформації.

Використання технології штучного інтелекту для захисту індустріальних мереж стає необхідністю, оскільки вона дозволяє виявляти та відповідати на кіберзагрози в режимі реального часу. Автоматизовані системи на основі ШІ можуть аналізувати величезні обсяги даних, виявляти аномалії в мережі, ідентифікувати невідомі атаки та приймати швидкі та точні рішення для захисту індустріальних об'єктів.

Метою даної дипломної роботи є дослідження та розробка інноваційного підходу до забезпечення кібербезпеки в індустріальних мережах через використання технології штучного інтелекту. Робота спрямована на вивчення сучасних методів захисту індустріальних систем, аналіз їхньої ефективності та визначення найбільш перспективних підходів. На основі отриманих знань розробляється та впроваджується система ШІ, спрямована на виявлення та відповідь на потенційні кіберзагрози в реальному часі.

Об’єктом дослідження в даній роботі є індустріальна мережа, зокрема системи автоматизації та управління, яку метою дослідження є застосування технології штучного інтелекту для підвищення рівня захисту від кіберзагроз.

Методи дослідження. Під час реалізації цього проекту проведено теоретичні дослідження, використовуючи методи комп'ютерного моделювання.

Практична цінність роботи заключається в можливості розширення програмного рішення з подальшим впровадженням у індустріальній сфері.

Структура та обсяг роботи. Робота містить 83 сторінки, складається з вступу та 4 розділів, висновків до розділів, 25 рисунків, 1 таблиць, 3 додатків, списку використаних джерел, які містять 23 найменувань.

РОЗДІЛ 1
АНАЛІЗ СУЧАСНОГО СТАНУ ІНДУСТРІАЛЬНОЇ МЕРЕЖІ

1.1 Industry 4.0

Індустрія 4.0, або як також відома Четверта промислова революція, є концептуальним підходом до оновлення та трансформації сучасних виробничих систем. Цей термін виник у відповідь на розвиток нових технологій та інформаційних систем, які впливають на способи та управління виробництвом.

Industrial 4.0 описується як бізнес модель майбутнього, яка буде побудована для охоплення механічного обладнання, систем зберігання та мережі на основі інформації. Одною із головних інструменті у сучасному індустріальному середовищі є системи управління та нагляду над процесами.

Завдяки взаємопов'язаному промисловому обладнанню та розширеним обчисленням Industry 4.0 дає змогу зацікавленими сторонам оптимізувати об'єкти та підвищити продуктивність. На рисунку 1.1 можна побачити засади індустрії 4.0 [1].

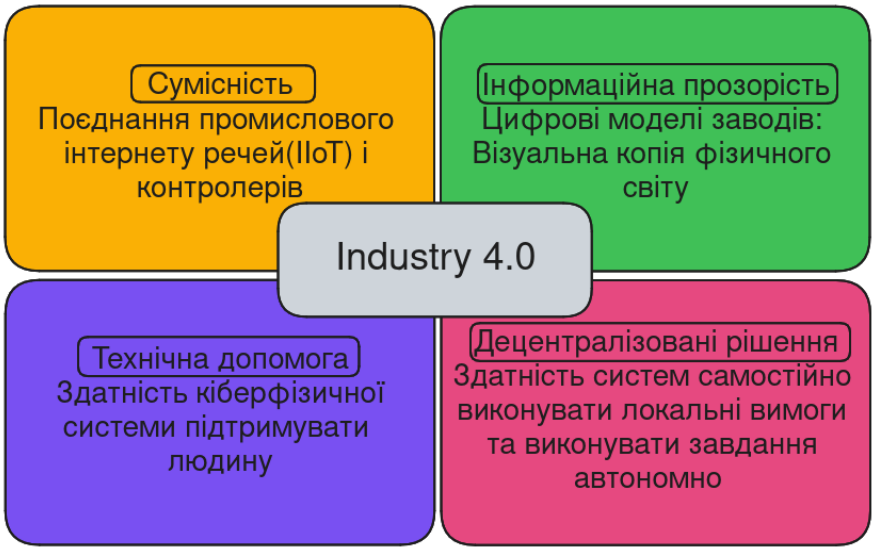

Рисунок 1.1 — Принцип проектування Industry 4.0

Давайте розглянемо кожен із принципів проектування Industry 4.0 окремо. Принцип сумісності є одним з ключових концепції. Цей принцип передбачає максимальну взаємодію інтелектуальних систем, технологій та процесів в області виробництва, забезпечуючи гнучкість швидкість та ефективність.

SCADA систем і IIoT, які будуть в подальшому розглянуті.

Принцип інформаційної прозорості передбачає створення цифрових моделей, які надають повний огляд всіх етапів виробничого процесу, а також можливість отримання детальної інформації про стан обладнання продукції та інших аспектів в реальному часі. Вони дозволяють підвищити ефективність виробництва, зменшити час простою обладнання та поліпшити управління всім життєвими циклом виробничого процесу.

Технічна допомога в контексті Industry 4.0 включає в себе використання передових технології для підтримки та оптимізації виробничих процесів, а також вирішення проблем та удосконалення ефективності. До таких аспектів відносяться: системи моніторингу та діагностики, аналітика даних, штучний інтелект та машинне навчання тощо.

Децентралізація рішень вказують на тенденцію розділення функцій та прийняття рішень на різних рівнях виробничого процесу, а не централізоване управління з одного пункту. Цей підхід базується на використанні розподілених систем, технологій блокчейн, децентралізованих мереж, інтернету речей (IoT) та інших інноваційних технології.

Ці принципи спільно створюють фундамент для трансформації сучасних виробничих процесів, роблячи їх більш гнучкими, продуктивними та підготовленими до викликів майбутнього. Застосування цих принципів дозволяє підприємствам вигравати в умовах змін і швидко реагувати на нові вимоги ринку, сприяючи створенню конкурентних та інноваційних рішень для керування виробничим середовищем.

1.2 SCADA

Індустріальна системи керування (ICS) представляє собою сукупність технологій та процесів, спрямованих на автоматизацію та управління різноманітними індустріальними процесами. До цих систем відносять така технологія як SCADA, Вона пропонує користувачам можливість керувати процесами і збирати інформацію з одного або кількох віддалених об’єктів. SCADA усуває необхідність присутності людини на віддалених об’єктах, де відбуваються стандартні операції.

Програми SCADA складаються з двох частин, а саме процесу керування і взаємозв’язку пристроїв, які формують інтерфейс із процесом, що розглядається, за допомогою датчиків і вихідних сигналів керування. Цей взаємозв’язок дозволяє користувачеві вимірювати та контролювати явні елементи виробничого процесу за допомогою SCADA.

Інтернет надає нам відкритий доступ до системи, до якої мають вільний доступ усі, що дозволяє швидко передавати інформацію всім зацікавленим сторонам в організації. Таким чином звичайні SCADA системи будуються на базі протоколів Інтернету. Це дозволяє виробникам дозволяє розробляти з підтримкою відкритих інтернет протоколі, які використовуються для з’єднання різних елементів мережі, в тому рахунку і IoT пристроїв. На рисунку 1.2 можна розглянути повну структуру SCADA-систем.

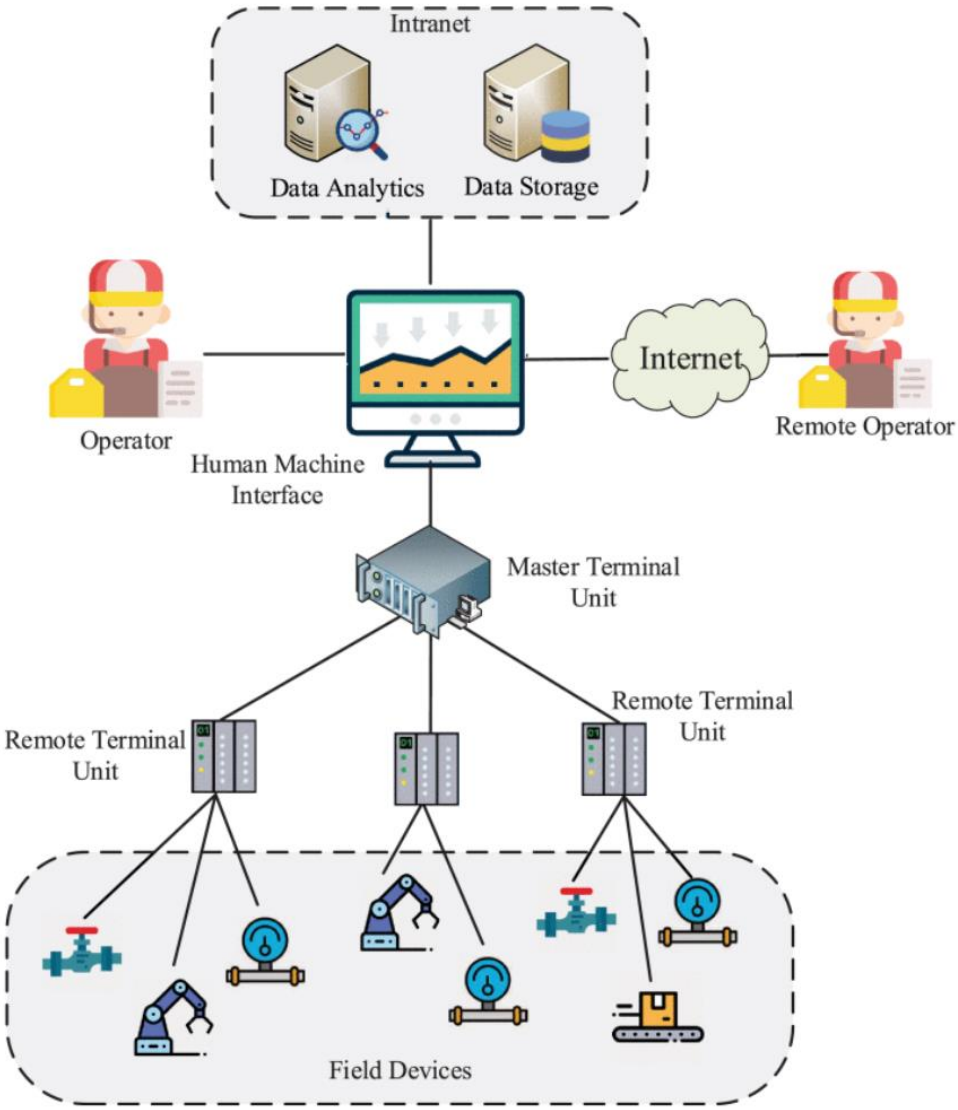

Рисунок 1.2 — Архітектура SCADA - системи

З рисунку можна побачити що SCADA-система складається з наступних компонентів:

- оператор – це особа яка відповідає за моніторинг системи, відправляти повідомлення, і контролювати систему, оператор може мати як прямий доступ та і віддалений доступ через мережу інтернет до системи;
- HMI — людино-машиний інтерфейс, який дозволяє взаємодіяти оператору із SCADA-системою;
- сервери — вони відповідають за збір даних з польових пристроїв і їх подальший аналіз, зазвичай вони розташовані в середині підприємства;

- головний термінальний блок (MTU) — відповідає за збір даних від RTU, і подальший їхню передачу до HMI, також він відправляє сигнали керування до віддалених терміналів, він також забезпечує високорівневу логіку керування системою;
- віддалений термінал (RTU) — на даний пристрій надсилаються команди з головного блоку, які в подальшому RTU вже направляє сигнали керування до промислових пристроїв;
- промислові пристрої — вони розподіленні по всьому виробничому об’єкту які керують промисловими процесами і датчиками які збирають всю необхідну інформацію [2].

Системи SCADA мають велике значення в управлінні процесами. Їх можна використовувати для широкого спектру застосувань, як малих, як клімати-контроль так і великих як моніторинг атомної станції чи системи масового транспорту.

Також данні системи надзвичайно економічні, менше персоналу потрібно для керування обладнанням у віддалених місцях і вирішення проблем технічного обслуговування. [3]

1.3 Industrial Internet of Things

IoT, або Інтернет речей, це концепція мережі фізичних об’єктів, які з’єднанні і взаємодіють між собою за допомогою інтернет протоколів. Тобто в концепції закладане наступна ідея, що звичайні об’єкти, перетворюють на інтернет об’єкти. Ключовою ідеєю IoT полягає в розгортанні мільярдів розумних об’єктів, здатних сприймати навколишнє середовище, передавати й обробляти отримані дані, а потім забезпечувати зворотній зв’язок із середовищем.

Застосування IoT у промисловій сфері обіцяє кращу гнучкість і ефективність, що дає створювати повністю зв’язану інфраструктуру для підтримки інноваційних послуг. Очікується, що IoT з’єднає нетрадиційні об’єкти з Інтернетом, підвищить стабільність і безпеку галузей і суспільства, а також забезпечить ефективну взаємодію між фізичним світом і його цифровим аналогом, тобто тим, що зазвичай називають кіберфізичною системою (CPS).

Підключення пристроїв IIoT змінюється залежно від того яка комбінація магістральної та локальної архітектури є корисною в певній ситуацій, а також прийнятна комбінація бездротових та/або дротових технології. Ключова мета полягає в тому, щоб уникнути ізольованих систем, заснованих на власних рішеннях і забезпечити обмін даними та взаємодію між цими закритими підсистемами і майбутніми додатками у галузі і між ними.

IIoT в рамка виробництва вимагає протоколів зв’язку здатних підтримувати ефективне, своєчасне та повсюдне агрегування та доступність інформації. Нижні рівні стеку повинні адекватно відповідати вимогам масштабованості та гнучкості. Верхні рівні повинні дозволяти так званим «розумним» пристроям транспортувати дані, не обмежуючися цікавою інформацією, але також забезпечуючи обізнаність користувачів, для яких вони призначені, і всю семантичну правила, які також повинні бути правильно зрозумілі в місцях призначення. Можна виділити три макрорівні, тобто рівень сприйняття, мережевий, рівень і прикладний рівень.[4] На рисунку 1.3 можна розглянути рівні протоколів в IIoT.

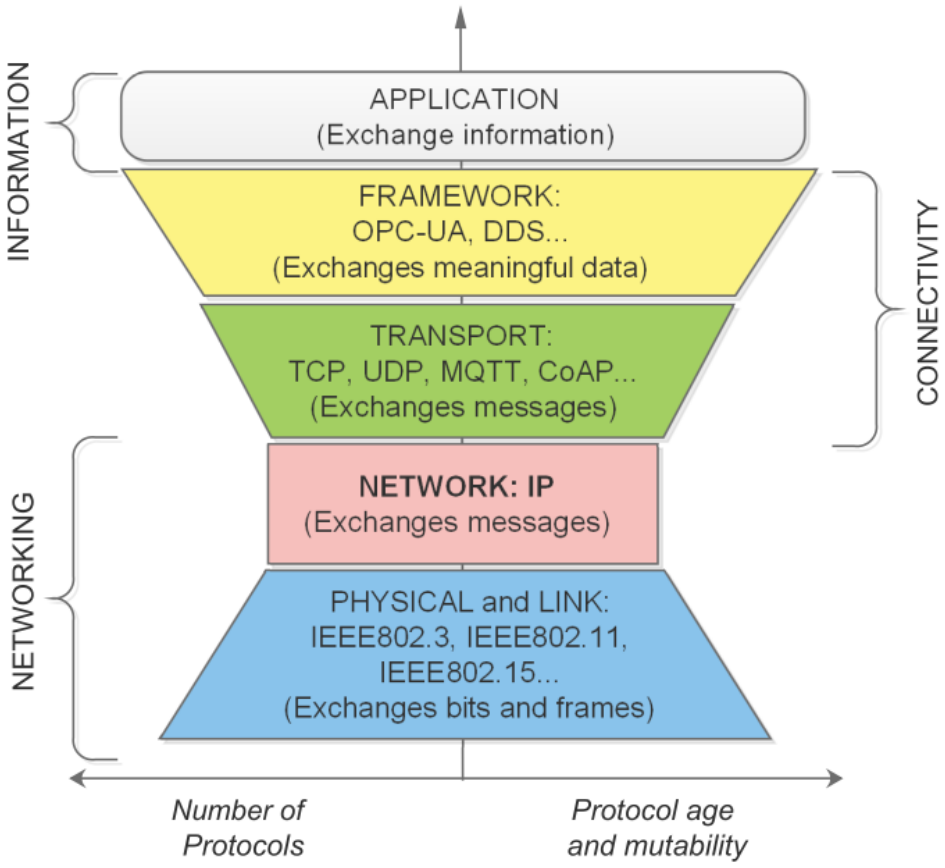

Рисунок 1.3 — Стек протоколів IIoT

Потрібно також зауважити, що парадигма IIoT, не призначена для заміни традиційних програм автоматизації, а спрямована на розширення знань про фізичну систему. Тому впровадження IIoT в промислову мережу є досить складним завданням. Програмам часто пред'являють різноманітні та суворі вимоги до якості обслуговування(QoS) щодо конфігурацій, стійкості, надійності, затримки, енергоефективності, тощо. Як наслідок IIoT не пов'язаний із застосуванням керування на польовому рівні, де має бути забезпечений обмежений час реакції.

Однак слід додати, що концепція інтернет речей активно розвивається, особливо в індустріальному напрямку, і знаходяться рішення щодо подальшого його провадження на виробництві. Тому потрібно завжди мати на увазі що кількість «розумних» пристроїв буде збільшуватися, як і трафік між ними.[5]

1.4 Великі дані та аналітика даних

Індустрія 4.0 і кіберфізичні системи дозволяють генерувати великі обсяги промислових великих даних з рідних джерел, таких як датчики машини, персонал, клієнти, тощо. Big Data можна використовувати для створення основ даних для систем підтримки прийняття рішень, які можуть допомогти компаніям оптимізувати свою продуктивність шляхом збору та аналізу даних протягом усього життєвого циклу продукту. Що дозволяє виробничому процесу бути більше прозорим та контролюватися більш ефективним способом.

Обробка промислових великих даних передбачає форматування даних, зменшення розмірності, ідентифікацію прихованих шаблонів, оцінку продуктивності та прогнозування тощо. Основною метою використання великих даних у промисловості є досягнення безвідмовності та ефективність витрати для виконання бажаного рівня продуктивності і якості.

Останнім часом активно починає використовуватися архітектура OPC-UA на підприємствах. Яка може використовуватися для інтерфейсів різних IT-інструментів у компанії, а також збору та передачі даних із різноманітних інструментів IoT, а також даних із мобільних пристроїв людей-операторів.

OPC-UA (Уніфікована архітектура) — це стандарт, який забезпечує незалежну від платформи сервіс-орієнтовану архітектуру для безпечного та надійного обміну даними між системами промислової автоматизації та корпоративними системами. Архітектура OPC-UA забезпечує розширювану модель даних, яка визначає схему даних, що дозволяє навіть системам, які не знайомі з моделлю даних, отримувати інформацію з інших систем. На рисунку 1.4 показано, як OPC-UA може забезпечувати зв’язок на всіх рівнях виробничих підприємств, від ресурсного до фабричного, за допомогою сервіс-орієнтованої архітектури.

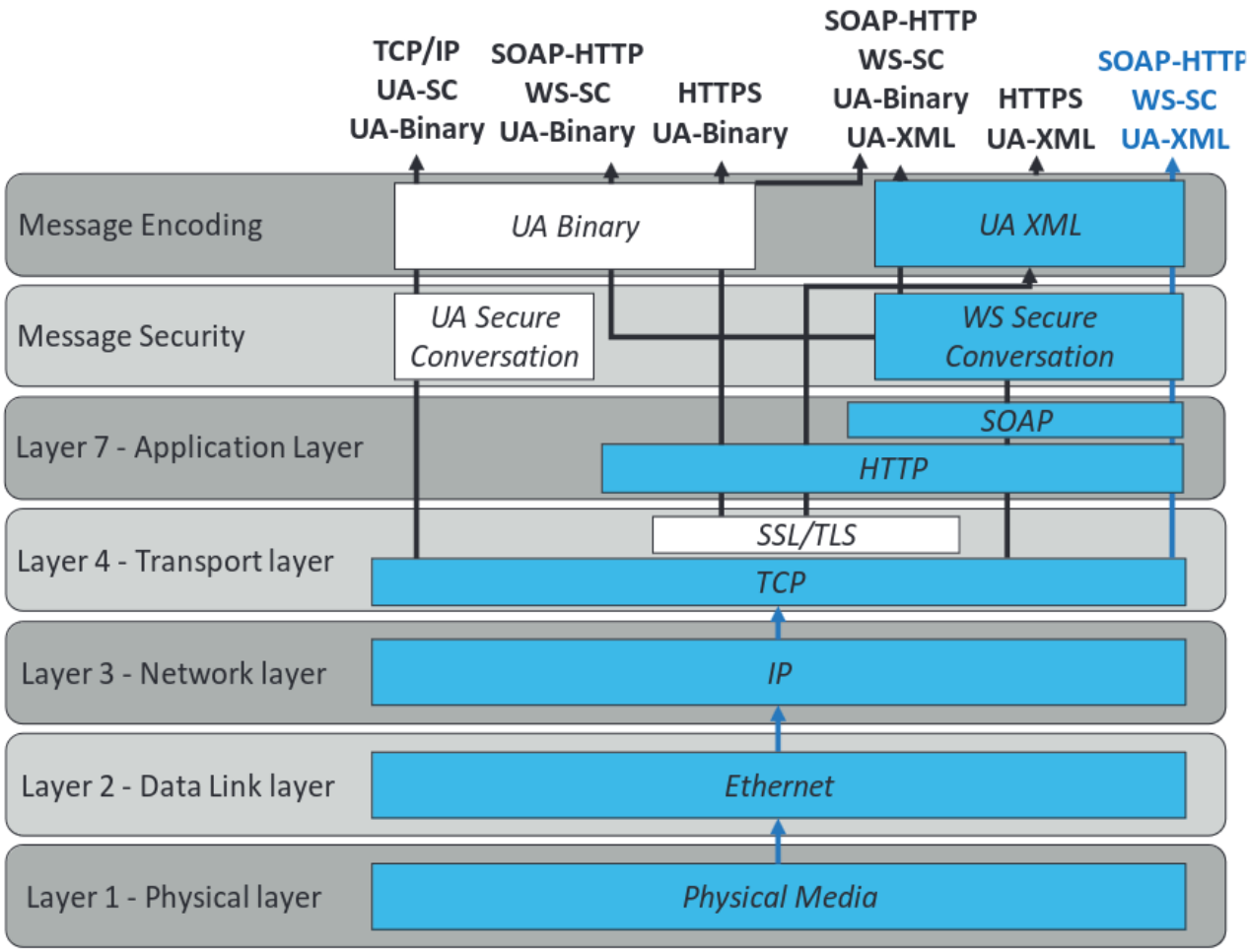

Рисунок 1.4 — Зв’язок OPC-UA з OSI

На рисунку 1.4 також можна побачити різні рівні моделі OSI, включаючи фізичний рівень, рівень каналу даних, мережевий рівень, транспортний рівень і прикладний рівень, а також різні Інтернет-протоколи, які можна використовувати на кожному рівні, наприклад Ethernet, IP, TCP, SSL/TLS, SOAP-HTTP, WS-SC, UA-Binary, UA-XML і HTTPS.

Архітектура OPC-UA забезпечує незалежну від платформи сервіс-орієнтовану архітектуру для безпечного та надійного обміну даними між системами промислової автоматизації та корпоративними системами, а також підтримує різноманітні інтернет-протоколи, такі як SOAP-HTTP, WS-SC, HTTPS та UA- XML.[6]

Стандарт OPC-UA має потенціал для підтримки інтеграції обладнання різних виробників і архітектури для реалізації бачення Industry 4.0. А впровадження парадигми IoT у виробництво генеруватиме промислові великі дані.

1.5 Висновки за розділом

В першому розділі випускної роботи був проведений аналіз сучасного стану індустріальної мережі. Зосередились на ключових аспектах сучасної промисловості, таких як Industry 4.0, Industrial Internet of Things (IIoT), Supervisory Control and Data Acquisition (SCADA), та Big Data. Ці технології та концепції визначають майбутній етап розвитку індустріального виробництва, забезпечуючи підвищену автоматизацію, ефективність та конкурентоспроможність.

Індустрія 4.0, як концепція, вводить інтеграцію сучасних цифрових технології у виробниці процеси, створюючи так званні «розумні підприємства» здатні до самонавчання та оптимізації. Індустріальні IoT розширюють можливість збору та обміну даними між пристроями в реальному часі, роблячи процеси виробництва більш гнучкими та ефективними. SCADA системи відіграють ключову роль у візуалізації та керування промисловими процесами.

Однак важливо зазначити , що сучасні технології також стають об’єктами збільшення кількості кіберзагроз. Забезпечення надійного захисту виробничих мереж стає невід’ємною частиною впровадження цих інновацій. У контексті дослідження можна зробити висновок, що всі технології активно використовують мережеву модель IP.

IP визначає стандартний протокольний стек для забезпечення зв’язку та обміну даними в мережах Industry 4.0. Його поширеність свідчить про

універсальність та ефективність у вирішенні завдань комунікацій між обладнанням та системою.

Отже, висновуючи, можна стверджувати, що розвиток індустріальних мереж, заснованих на технологіях Industry 4.0, IIoT, SCADA та використання мережевої моделі інтернет протоколу, вимагає уважного управління кібербезпекою для забезпечення стійкості, конфіденційності та доступності виробничих систем.

РОЗДІЛ 2
АНАЛІЗ СУЧАСНИХ МЕРЕЖЕВИХ ЗАГРОЗ

2.1 Мережевий стек інтернет протоколу

Інтернет-протокол(IP) — це метод за допомогою якого дані надсилаються з одного пристрою на інших або в Інтернет мережу. В протоколі кожен пристрій відомий як хост, який має унікальний ідентифікатор, тобто IP-адресу.

Також IP визначає протокол за яким дані переміщуються між хостами, одиницею, яка фактично переміщується, це є IP-пакет. IP-пакет складається з двох частин: заголовку і поля даних. Вона може містити інформацію необхідну для маршрутизації пакет, це такі дані як: ідентифікатор, розмір, протокол, адреса відправника і отримувача, протокол тощо. Це все відбувається на етапі інкапсуляції, коли на кожному рівні OSI відбувається додавання заголовків з технічною інформацією для подальшої передачі пакету, на рисунку 2.1 зображено як відбувається інкапсуляція пакету.

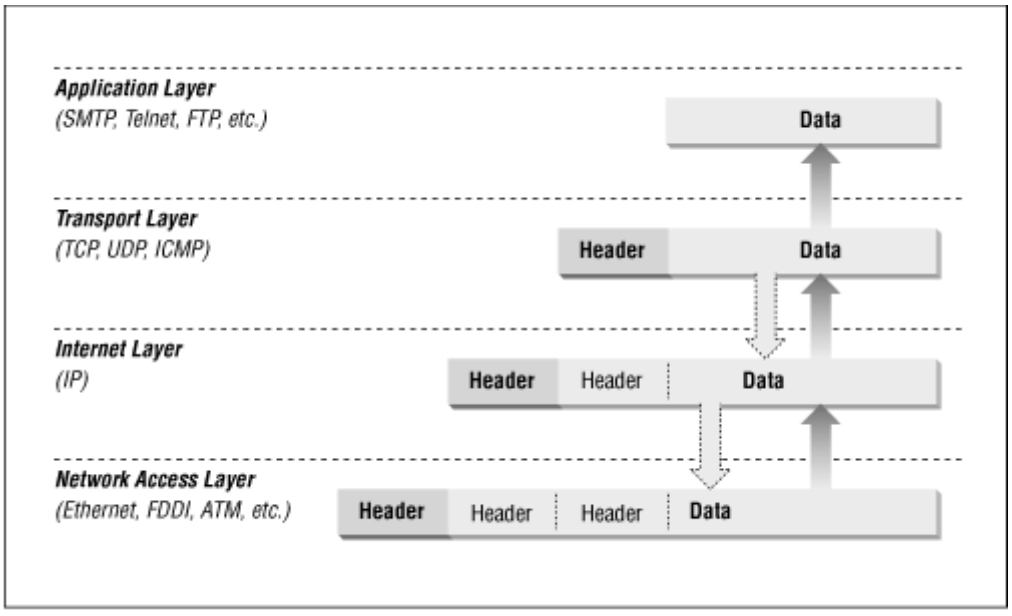

Рисунок 2.1 — Інкапсуляція пакету

Industry 4.0 використовує IP як основний протокол для забезпечення зв’язку та обміну даними між пристроями та системами в індустрії. IP є стандартом для мережевого з’єднання та передачі даних в інтернеті. Використання його в індустрії 4.0 дозволяє створювати інтегровані та ефективні виробничі системи.

Для з’єднання пристроїв за допомогою кабелів використовують частину фізичну протоколу, а саме Ethernet/IP. Це дозволяє спілкуватися пристроями один з одним через провідну мережу Ethernet, забезпечуючи обмін даними в реальному часі та контролювати промислові процеси.

Однією з ключових переваг протоколу Ethernet/IP є його гнучкість. Його можна використовувати з широким спектром пристроїв і систем, до яких входять програмовані логічні контролери(ПЛК), людино-машині інтерфейси(HMI), сервери і датчики. Це робить його популярними вибором для промислової автоматизації, особливо у великих установка.

Ще однією перевагою протоколу Ethernet/IP є його швидкість. Завдяки швидкості передачі даних до 1 Гбіт/с він забезпечує зв’язок між пристроями в реальному часі, забезпечуючи швидку та надійну передачу даних. Це особливо важливо для промислових застосувань, де навіть невеликі затримки в передачі даних можуть мати значний вплив на роботу.

Окрім гнучкості та швидкості, протокол Ethernet/IP також забезпечує надійні механізми виявлення та виправлення помилок. Він використовує перевірку циклічної надлишковості (CRC), щоб переконатися, що дані передаються правильно та без помилок. Це робить його надійним протоколом для промислових застосувань, де точність даних має вирішальне значення.

Ще однією ключовою перевагою протоколу Ethernet/IP є його сумісність з іншими протоколами зв’язку. Багато промислових пристроїв і систем підтримують протокол Ethernet/IP, що полегшує інтеграцію з існуючими установками. Крім того, багато програмних програм підтримують протокол

Ethernet/IP, що полегшує розробку спеціальних програм, які можуть обмінюватися даними з промисловими пристроями.

Загалом, протокол Ethernet/IP є потужним і універсальним протоколом зв'язку, який покращує промисловий зв'язок. Його гнучкість, швидкість, надійність і сумісність роблять його популярним вибором для промислової автоматизації. [7]

В контексті використання IIoT стає питання в використанні UDP/IP і TCP/IP. Вони мають наступні відмінності між собою і використовують при визначених сценаріях які зображені в таблиці 2.1.[8]

Таблиця 2.1 — Різниця між транспортними протоколами

Характеристика	UDP/IP	TCP/IP
Затримка	Низька	Висока
Багатоадресна розсилка	Підтримується	Не підтримується
Управління потоком	Не підтримується	Підтримується
Виправлення помилок	Не підтримується	Підтримується
Надійність	Ненадійний	Надійний
Випадки використання	Програми IoT які працюють в реальному часі з низькою пропускною здатністю	Програми IoT з високою пропускною здатністю та QoS

Загалом використання IP протоколу в Industry 4.0 є ключовим фактором у створенні сучасних, високоефективних систем, які перетворюють сферу промисловості. Цей протокол дозволяє впроваджувати розподілені мережі, що мають здатність взаємодіяти інтегровано, створюючи величезний потенціал для удосконалення виробничих процесів. Інтеграція цих мереж сприяє не лише збору великої кількості даних з різних джерел, таких як датчики, обладнання та системи моніторингу, але й їхній обробці в режимі реального часу.

2.2 Класифікація мережевих загроз

Мережева загроза — це потенційна можливість або намір використати вразливості в мережах та інформаційних системах для завдання шкоди, незаконного отримання доступу чи викрадення даних.

Активна атака на комп'ютерні мережі є важливим аспектом забезпечення кібербезпеки, і її можна розділити на три основні типи: переривання, модифікація і підробка. Кожен з цих видів атак викликає певні загрози, які можуть серйозно підірвати безпеку системи.

Переривання – це атаки, які спрямовані на порушення доступності системи. Серед них особливо визначаються атаки типу DoS (Denial of Service) та DDoS (Distributed Denial of Service), які мають на меті переповнення ресурсів системи або мережі, тим самим призводячи до тимчасового або постійного відмову в обслуговуванні. Крім того, SQL-ін’єкції також відносяться до цього типу атак, оскільки вони спрямовані на неправомірне внесення змін у базу даних через введення шкідливого SQL-коду.

Атаки модифікації, зокрема загроза "людина-посередині", націлені на зміну або перехоплення інформації під час її передачі по мережі. Цей вид атаки може призвести до витоку конфіденційних даних або навіть зміни вмісту передаваних повідомлень.

Підробка включає атаки, спрямовані на створення фальшивих ідентичностей чи даних для обману системи безпеки. Повторні атаки в цьому контексті можуть включати низку спроб використання вже вкрадених або здобутих даних для нанесення подальших ударів. Метод маскування використовується для приховування справжнього характеру атаки, роблячи її важкозрозумілою для систем безпеки.

На рисунку 2.2 подано загальну класифікацію існуючих атак, яка слугує основою для розуміння різних методів, які можуть бути використані для захисту від них. Вивчення та усвідомлення цих ризиків дозволяє розробникам і

адміністраторам мережі ефективно вдосконалювати системи безпеки та вчасно реагувати на потенційні загрози.

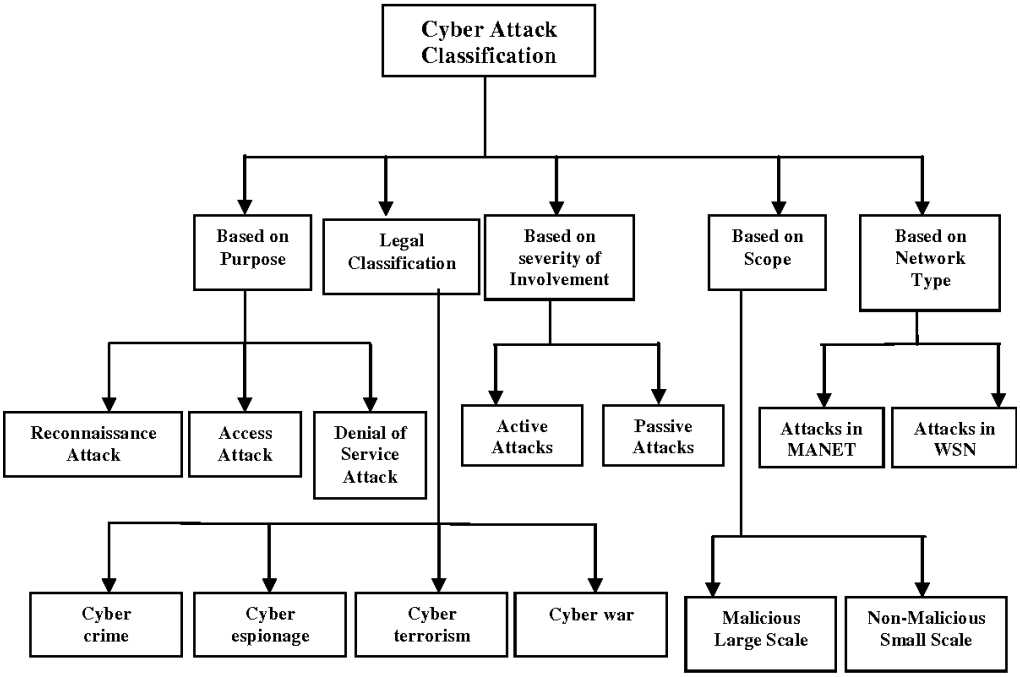

Рисунок 2.2 — Класифікація атак

2.3 Проблема безпеки в мережі Industry 4.0

Оскільки однією з ключових характеристик Industry 4.0 є використання інтелектуального виробничого обладнання, датчиків і процесів нового покоління, вектори атак, відомі кіберзловмисникам, необхідно розглядати в контексті середовища Індустрії 4.0.

Промислове обладнання без належних заходів безпеки є не єдиною проблемою в цьому контексті. Існують також інші платформи та інфраструктура , яка також може бути атаковані зловмисником. Наприклад, комп'ютери які використовуються для розробки моделей автоматизованого проектування (CAD), можуть бути атаковані зловмисним програмним забезпеченням.

Оскільки попит на вищу продуктивність, більшу потужність і продуктивність постійно зростає, питання масштабованості стає ключовим

моментом для розгляду. Розширення потужностей обробної промисловості зазвичай передбачає додавання виробничого обладнання, а згодом і розширення інформаційної архітектури виробництва.

Поширене розділення будівель є лише одним із багатьох прикладів факторів, які суттєво визначатимуть характер розширення мережі. Фактори, що визначають характер розширення мережі, як правило, включають великомасштабне взаємозв'язок, високу пропускну здатність, низьку затримку та, звичайно, надійну безпеку.

Окрім того, що заходи безпеки промислової системи розглядаються як другорядний пріоритет, той самий аргумент можна застосувати до безпеки промислової системи. Зокрема, що стосується високого рівня автоматизації системи, необхідного для досягнення вищих рівнів продуктивності та ефективності, існує успадкований ризик того, що безпека системи не однаково враховується при проектуванні таких промислових систем. [9]

На рисунку 2.3 наведена схема типової архітектури промислової мережі, яка включає такі компоненти як:

- корпоративна мережа — до даної частини відносяться звичайні хости в мережі, такі як персональні комп'ютери, сервери, периферія тощо;
- ДМЗ — в мережах також існує ДМЗ яка має функцію обмеження важливих хостів в мережі, такі як сервери, файлові сховища;
- технологічна мережа — в ній находяться HMI і сервери які обробляються інформацію з виробництва;
- мережа керування — розташовані мережеві термінали;
- польова зона — промислові пристрої і датчики.

Також потрібно зауважити, що мережі можуть бути розділені між собою за допомогою брандмауера, який повинен виконувати роль контролю трафіку, що проходить через нього, і блокування небажаного трафіку.

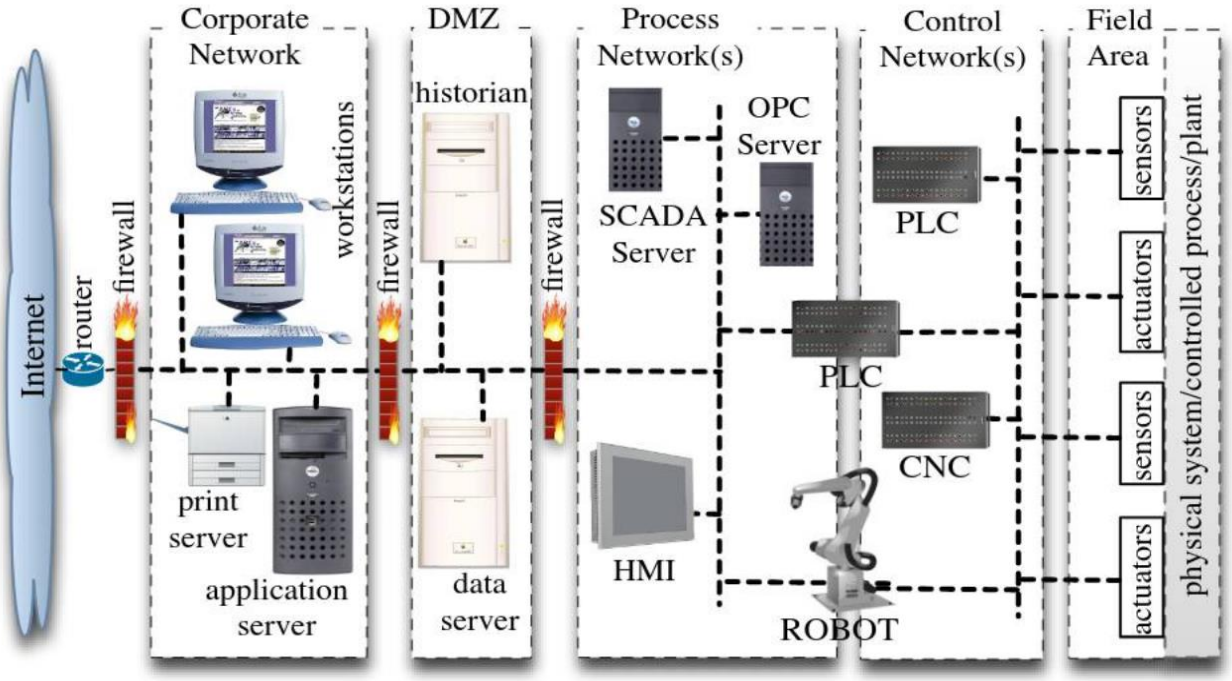

Рисунок 2.3 — Архітектура промислової мережі

У подальшому дослідженні буде розглядатися вплив DDoS (розподілені атаки на обслуговування) на індустріальні мережі. Цей вид кібератак представляє серйозну загрозу для систем Industry 4.0, оскільки вони є необхідним елементом сучасних виробничих процесів. DDoS атаки можуть призводити до тимчасового припинення функціонування виробничих систем, порушуючи нормальний режим роботи та викликаючи значні фінансові збитки.

У цьому контексті також важливо враховувати еволюцію DDoS атак та їхніх методів, оскільки кіберзлочинці постійно вдосконалюють свої стратегії. Дослідження цього питання дозволить розробити ефективні та сучасні заходи протидії DDoS атакам, щоб забезпечити неперервну та безпечну роботу індустріальних мереж у сучасному виробничому середовищі.

2.4 DDoS атаки та їх вплив на доступність системи

Розподілена атака на відмову в обслуговуванні (DDoS) намагається зробити онлайн-службу або сервер недоступним, перевантажуючи їх величезним потоком через інтернет мережу, створеного з багатьох джерел.

Атака на відмову в обслуговуванні (DoS), під час якої один пристрій який має зв’язок з пристроєм для атаки, заповнюючи пристрій пакетами. Але DDoS використовується багато комп’ютерів які підключені в мережу інтернет, і часто вони мають ознаку розподіленості по всьому світу, такі мережі з пристроїв називається ботнет.

Масштабна об’ємна DDoS-атака може генерувати трафік, який вимірюється десятками гігабіт на секунду. Зазвичай мережа не може впоратися з такою атакою. На рисунку 2.4 зображена схема роботи DDoS-атаки.

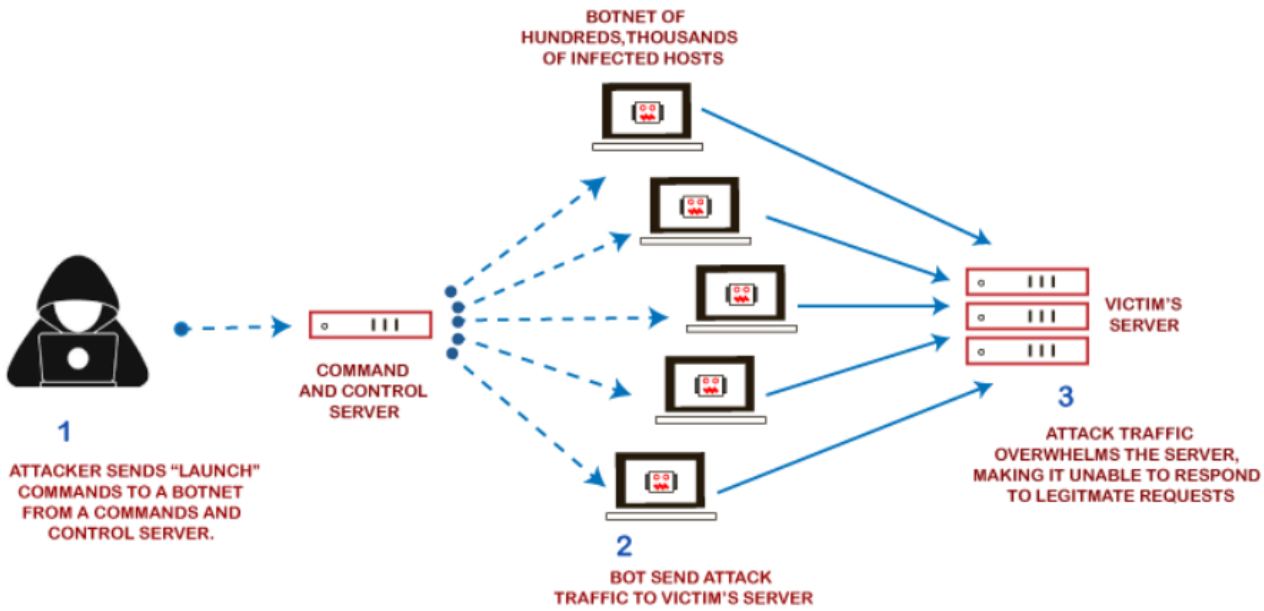

Рисунок 2.4 — Принцип роботи DDoS-атаки

Розподілені атаки типу «відмова в обслуговуванні» можна розділити на три категорії:

1. Об’ємні атаки — вони використовують величезну кількість підробленого трафіку, щоб перевантажити об’єкт. Він включає такі пакети як TCP, UDP, ICMP та інші підроблені пакети.

2. Атаки на протокол — DDoS-атаки на рівні протоколу або мережі, надсилають велику кількість пакетів до цільової мережевої інфраструктури та інструментів керування інфраструктурою. Це включає SYN-флуд, Ping of Death, фрагментовані пакетні атаки, Smurf DDoS тощо. Цей тим атаки використовує наявні ресурси сервера та інші ресурси такі як брандмауери та балансувальники навантаження.
3. Атаки рівня додатків — заповнення додатків зловмисно створеними запитами призводить до атак на прикладному рівні. Тобто відбувається атаки вже на цільовий застосунок.[10]

Можна зазначити що DDoS-атаки є серйозною загрозою для інформаційної безпеки. DDoS-атаки визначаються масштабністю і високою складністю, оскільки вони спрямовані на перевантаження ресурсів мережі чи сервера. Застосування ефективних захисних стратегій та планування в разі DDoS атак є критичними елементами для збереження стабільності мережевого середовища та забезпечення безпеки в онлайн просторі.

2.5 Заходи забезпечення безпеки мереж

Розповсюдженість атак DoS і DDoS у сфері кібербезпеки зростає з кожним днем, і це визначається не лише їхньою ефективністю, а й легкістю їхнього впровадження. Однією з основних причин такого поширення є доступність інструментів та програм для здійснення атак. Зловмисники можуть використовувати навіть обмежені ресурси для створення великих масштабів атак, що призводять до великих збоїв у роботі веб-сайтів та мережевих інфраструктур.

Зростання кількості пристроїв, які підключені до Інтернету, також відіграє ключову роль у поширенні таких атак. Розгалуження Інтернету різними пристроями, починаючи від комп'ютерів та смартфонів до побутових речей та промислових систем, створює величезний атакувальний потенціал для

кіберзлочинців. Популяризація концепції "речей в Інтернеті" (IoT) надає зловмисникам додаткові можливості віддаленого вторгнення в пристрої, які можуть стати частиною ботнетів і брати участь у масштабних DDoS-атаках.

Зловмисники також використовують розподілені системи для координації своїх атак, зробивши їх більш важкими для виявлення та блокування. Спрощена можливість взяття під контроль та координації пристроїв, які входять до складу ботнетів, робить цю стратегію особливо привабливою для зловмисників.

Загалом, поширеність атак DoS і DDoS визначається не лише технічними аспектами, а й суспільними тенденціями, що допомагають кіберзлочинцям використовувати максимум можливостей для здійснення широкомасштабних і вразливих атак.

Останнім часом технології штучного інтелекту отримують широке визнання і розповсюдження. З їхньою допомогою можна виявляти та ефективно запобігати атакам типу DDoS. Використання штучного інтелекту представляє собою передовий підхід, який може легко пом'якшити наслідки будь-якого типу кібератак.

Конструкція штучного інтелекту, яка може імітувати когнітивну поведінку людини, пропонує новий підхід у вирішенні проблем кібербезпеки. За зразком природної інтелігенції, штучний інтелект вчиться на основі досвіду, адаптується до нових ситуацій і здатний розпізнавати та реагувати на аномальні патерни поведінки, які можуть свідчити про потенційні кіберзагрози.

Збираючи велику кількість даних і використовуючи алгоритми машинного навчання, штучний інтелект може ефективно виявляти нові загрози та навіть передбачати їхні майбутні варіації. Одним із ключових переваг цього підходу є його здатність вчитися на льоту, адаптуватися до змін у ландшафті кібербезпеки та впроваджувати вдосконалення без необхідності вручну програмувати нові правила чи алгоритми.

Коли штучний інтелект набуває достатнього досвіду, він може стати потужнішим, ніж традиційне програмне забезпечення. Здатність до аналізу

великих обсягів даних у реальному часі та визначення важливих зв'язків дозволяє йому ефективно реагувати на нові загрози та атаки, а також робить його більш адаптивним до стрімко змінюваного характеру кіберзагроз.

Однак важливо враховувати, що впровадження цього підходу може виявитися складним завданням. Існують проблеми з етикою та конфіденційністю даних, особливо коли ШІ використовується для аналізу особистих даних користувачів. Крім того, ефективність та здійсненність штучного інтелекту в кібербезпеці потребують подальших досліджень та тестувань. Іншим важливим аспектом є здатність зловмисників адаптуватися до нових захисних стратегій, що може створити постійний захистовий кільцевий процес.

У всій своїй потужності та перспективності, штучний інтелект в кібербезпеці вимагає уважного та комплексного підходу для забезпечення не лише ефективності виявлення та протидії атакам, але й збереження приватності та етичних норм використання цієї технології.[11]

2.6 Висновки за розділом

У даному розділі був проведений аналіз мережевого стеку інтернету протоколу який використовується в інфраструктурі Industry 4.0. Розуміння принципів та особливостей цього стеку є важливим етапом для ефективного забезпечення безпеки та надійності мережевих зв’язків.

Далі була розглянута класифікація мережевих загроз та проблем безпеки в мережі Industry 4.0. Встановлено, що з впровадженням цифрових технологій зростає ризик виникнення різноманітних атак на мережу, що вимагає системного підходу до їх виявлення та протидії.

Окрему увагу приділено DDoS-атакам, які можуть серйозно загрожувати функціонуванню мереж у контексті Industry 4.0.І за активного впровадження IoT, вони є вразливою цілю для даних атак. Розглядаючи ці атаки, визначено, що

важливою частиною заходів безпеки стає використання штучного інтелекту. Сучасні технології штучного інтелекту, такі як машинне навчання та аналіз великих даних, можуть допомогти виявляти та протидіяти DDoS-атакам шляхом аналізу поведінки мережі та автоматизованого реагування на небезпеку.

Отже, в результаті проведених досліджень висновується, що ефективна мережева безпека в індустрії 4.0 вимагає комплексного підходу, який включає в себе не лише аналіз мережевого стеку та класифікацію загроз, але й застосування передових технологій штучного інтелекту для виявлення та протидії потенційним атакам, забезпечуючи стійкість та надійність мережі Industry 4.0.

РОЗДІЛ 3
АНАЛІЗ ТЕХНОЛОГІЇ ШТУЧНОГО ІНТЕЛЕКТУ

3.1 Теорія машинного навчання

Машине навчання(ML) — це застосування штучного інтелекту, де комп'ютер навчається на основі минулого досвіду (вхідних даних) і робить прогнози на майбутнє. Основним принципом ML є здатність системи адаптуватися та вдосконалювати свою продуктивність, навчаючись на основі даних.

Моделювання моделей ML тісно пов'язане з обчислювальною статистикою, основною метою якої полягає в тому, щоб зосередитися на прогнозуванні за допомогою комп'ютерів. Воно також пов'язане з математичною оптимізацією, яка пов'язує моделі, програми та фреймворки зі сферою статистки.

ML може використовуватися для вирішення абсолютно різних завдань, які вимагають навчання з боку машини. Задача навчання має три особливості:

- класи завдань (завдання, яке потрібно вивчити);
- показник ефективності, що потребує покращення;
- процес набуття досвіду

Загальна модель машинного навчання складається з шести етапів, які не залежать від обраного алгоритму. Кожен компонент моделі має певне завдання, яке потрібно виконати:

1. Збір та підготовка даних — першочерговим завдання у процесів машинного навчання є збір і підготовка даних у форматі, який може бути використаний як вхідні дані для алгоритму. Для будь-якої проблеми може бути доступна велика кількість даних.
2. Відбір ознак — дані отримані на попередньому кроці, можуть містити численні ознаки, не всі з яких можуть мати відношення до процесу навчання.

3. Вибір алгоритму — не всі алгоритми машинного навчання призначені для всіх проблем. Вибір найкращого алгоритму машиного навчання для конкретної задачі є обов’язковим для отримання найкращих результатів.
4. Вибір моделі і параметрів — більшість алгоритмів машинного навчання вимагають деякого початкового ручного втручання для встановлення найбільш відповідних значень різних параметрів.
5. Навчання — після вибору відповідного алгоритму та відповідних значень параметрів, модель потрібно навчити, використовуючи частину набору даних як навчальні дані.
6. Оцінка ефективності — перед впровадженням системи в реальному часі, модель повинна бути протестована на невидимих даних, щоб оцінити, наскільки багато було засвоєно за допомогою різних параметрів продуктивності, таких як точність. [12]

Залежно від того як навчається алгоритм і на основі доступності результатів навчання, парадигми машинного навчання можна розділити на декілька основних категорій:

- З вчителем — в даному випадку на вхід системи подається набір даних з відомим результатом для кожного відповідного ходу. Далі вже сам модель намагається встановити зв’язок між вхідними даними та результатом і на основі цього будує свій зв’язок. Приклад наведений на рисунку 3.1.

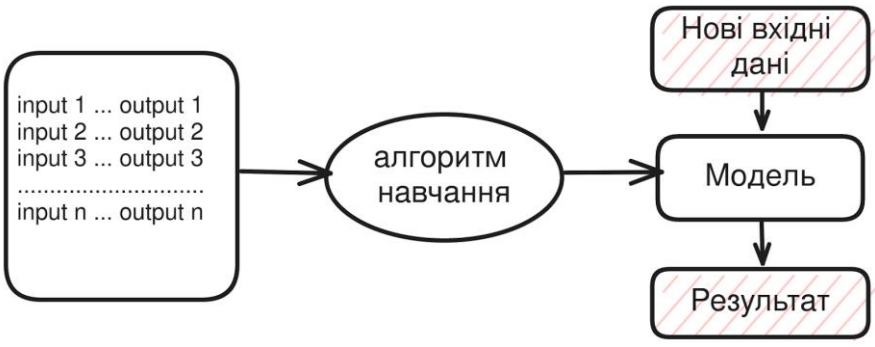

Рисунок 3.1 — Схема з вчителем

Без вчителя — дане навчання надає нам перевагу коли ми намагаємось вирішити задачу з невідомим результатом(або коли у нас недостатньо інформації про можливий результат). Формування результатів відбувається шляхом кластеризації даних на основі взаємозв'язку між вхідними значеннями. Тому головна завдання моделі є в тому, щоб створити групу даних з невідсортованої інформації на основі схожості, закономірності і відмінностей не маючи попередніх знано про навчальні дані. Приклад системи наведений на рисунку 3.2.

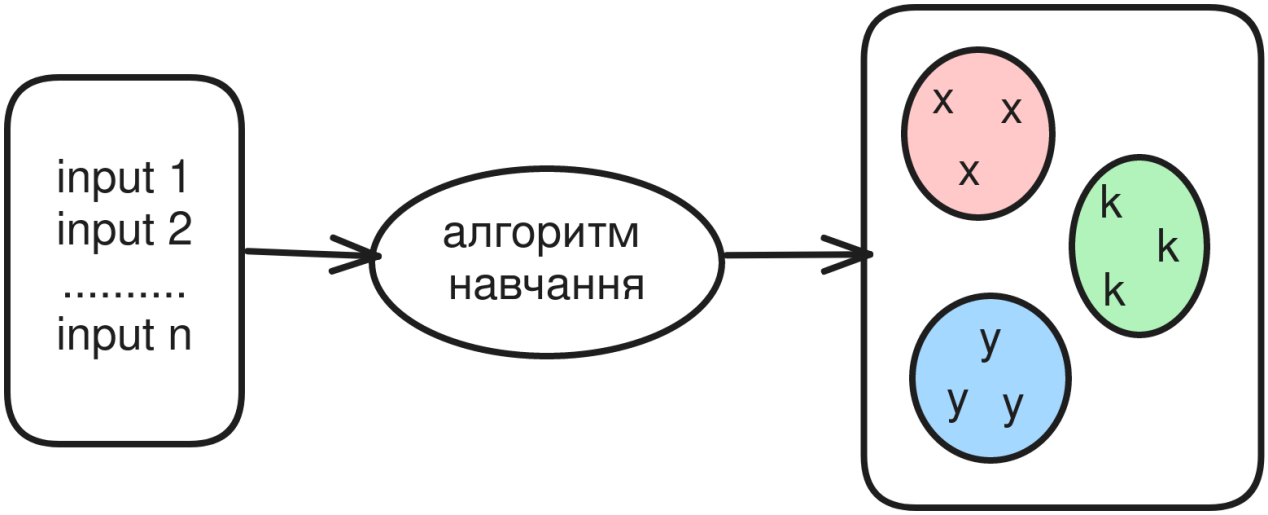

Рисунок 3.2 — Схема без вчителя

Також потрібно звернути увагу на навчання з підкріпленням, цей алгоритм має перевагу в використанні зворотного для навчання системи. Для прикладу машина на початку не має алгоритму дії на вхідні дані. Але подальші дії мають наслідки у вигляді штрафів і нагород за певні дії. Але даний алгоритм має головний нюанс в тому, що його застосовують саме до тих задач, які є послідовними. Навчання з підкріпленням повністю залежить від двох факторів — пошуку методом проб і помилок та відкладеного результату. На рисунку 3.3 наведений приклад навчання з підкріпленням. [13]

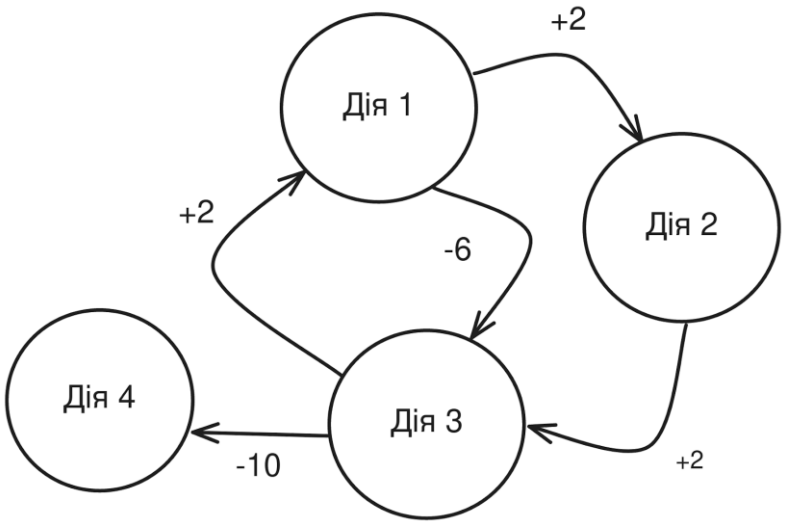

Рисунок 3.3 — Навчання з підкріпленням

Машинне навчання представляє собою застосування штучного інтелекту, де комп'ютери здатні навчатися на основі минулого досвіду, аналізуючи вхідні дані та роблячи прогнози для майбутнього. Цей підхід дозволяє системам автоматично вдосконалювати свої навички, не потребуючи явного програмування. Використовуючи різноманітні алгоритми та моделі, машинне навчання знаходить застосування у різних галузях, від обробки природної мови до медицини та фінансів. Успіх цієї технології залежить від якості навчальних даних, етичного використання та врахування прозорості та відповідальності у впровадженні.

3.2 Штучні нейронні мережі

Термін «штучна нейронна мережа» походить від біологічних нейронних мереж, які розвивають структуру людського мозку. Подібно до людського мозку, який має нейрони, які пов’язані між собою один з одним, штучні нейронні мережі також мають нейрони, які взаємопов’язані між собою. Ці нейрони відомі як вузли.

Штучна нейрона мережа — є частиною машинного навчання, головною особливістю якого є імітування структури та функції біологічних нейронних мереж. Основною частиною нейронних мереж є нейрон, який виконує роль математичної моделі(функції). Яка обробляє вхідні значення, і на виході дає єдиний результат.

Одношарові штучні нейронні мережі також називаються перцептронами. Перцептрон є основним компонентом ШНМ Перцептрон складається з п’яти компонентів, на рисунку 3.4 зображена структура :

1. Вхідні дані — це незалежні змінні (x), які ми отримуємо з навколишнього середовища або об’єкту.
2. Ваги — параметр (w) який контролює міцність зв’язку між вхідними даними та нейроном
3. Значення зсуву (b) — це постійне значення, яке дозволяє контролювати вхідне значення.
4. Функція активації (f) — встановлює вихід нейрону відповідно до конкретних умов.
5. Вихід — залежна змінна (y), яка є результатом отриманим з перцептрона.

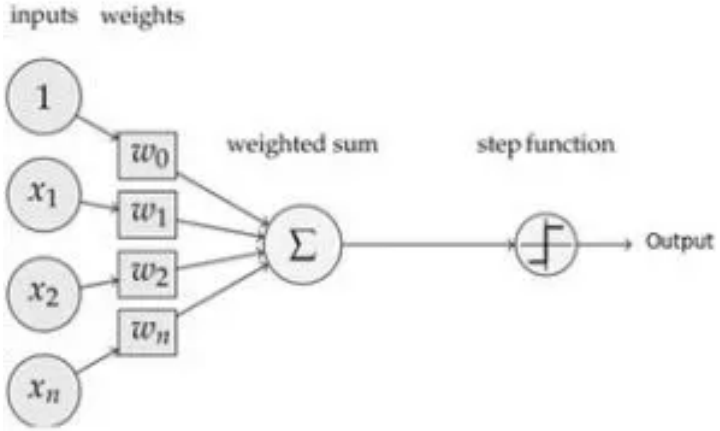

Рисунок 3.4 — Структура перцептрона

Також перцептрон можна представити формулою 3.1.

$$y = f(x \times w + b) \qquad 3.1$$

Процес відбувається наступним чином, зважена сума обчислюється шляхом спочатку множення ваг і вхідних даних, а потім їх додавання 3.2.

$$Z = \sum_{i=0}^{n} x_i \, \omega_i = x_0\omega_0 + x_1\omega_1 + \cdots + x_n\omega_n \qquad 3.2$$

Функція активації 3.3 яка застосовується до зваженої суми (Z) і є результатом. Використання перцептронів це крокова функція, тобто в результаті ми отримуємо значення один або нуль.

$$output(o) = f(x) = \begin{matrix} 1if\omega \cdot x + b > 0 \\ 0otherwise \end{matrix} \qquad 3.3$$

Область застосування перцептронів обмежена. Оскільки перцептрони зазвичай використовуються для простих задач бінарної класифікації, які є лінійно роздільними.[14]

3.2.1 Глибині нейронні мережі

Глибина нейронна мережа (DNN) — це штучна нейрона мережа із кількома прихованими рівнями між вхідним і вихідним рівнями. Багатошарові нейронні мережі, або багатошарові перцептрони, складаються з кількох шарів. Окрім перцептронів, їх можна використовувати для нелінійно роздільних задач. Вони досягають цього за допомогою функцій активації, які вони використовують у своїх шарах. Функції активації роблять вихід нейронів нелінійними. Таким чином, це дозволяє вирішувати більш складні проблеми. Багаторівнева нейрона мережа складається в основному з 3 рівнів які зображені на рисунку 3.5:

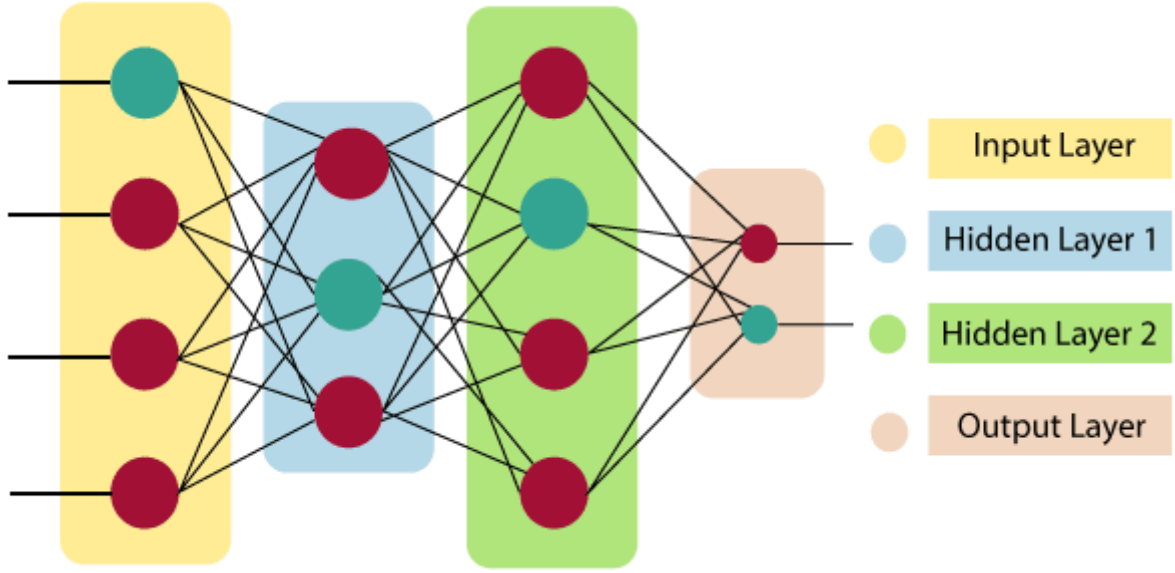

Рисунок 3.5 — Структура багаторівневої нейронної мережі

- Вхідні або видимі шари — це нижній рівень, який приймає вхідні дані з набору даних, або вже напряму з об’єкту.
- Приховані шари — є проміжним широм між вхідним і вихідним. Також прихованих шарів може бути декілька, в залежності від складності самої мережі. Він виконує всі обчислення щоб знайти приховані особливості та закономірності.
- Вихідний шар — є останнім шаром який формує результат роботи всієї багатошарової нейронної мережі.

Глибинні нейронні мережі використовуються для вирішення завдань машинного навчання та штучного інтелекту, шляхом вивчення представлення даних на різних рівнях абстракції. Замість того, щоб ручним чином визначати характеристики або ознаки вхідних даних, глибинні мережі автоматично вивчають корисні представлення на кожному рівні їхньої ієрархії. Кожен шар в мережі відповідає за визначення конкретних аспектів вхідних даних, інкапсулюючи їхні ознаки або атрибути.

Наприклад, у зображеннях перші шари можуть виявляти прості текстурні деталі чи лінії, тоді як більш глибокі шари можуть агрегувати ці прості функції

для визначення більш складних об'єктів або структур. Глибинність мережі дозволяє їй ієрархічно моделювати різноманітні рівні абстракції, а отже, враховувати складність та взаємозв'язки вхідних даних. Це робить DNN ефективними в різних областях, від розпізнавання образів до обробки природної мови, де ієрархічна репрезентація може відтворити природну структуру інформації.

Такі мережі можуть бути успішно використані в різних областях, таких як комп'ютерне зорове сприйняття (наприклад, розпізнавання обличчя або об'єктів), обробка природної мови (наприклад, машинний переклад або розпізнавання мови), медицина, фінанси та інші. Однією з найпопулярніших архітектур глибинних нейронних мереж є згорткові нейронні мережі (CNN) для обробки візуальної інформації та рекурентні нейронні мережі (RNN) для роботи з послідовнішими даними.

3.2.2 CNN

Згорткова нейронна мережа, також відома як CCN, є класом нейронної мережі, яка спеціалізується на обробці даних, що мають сіткову топологію для прикладу зображення. Цифрове зображення — це двійкове представлення візуальних даних. Він містить серію пікселів, упорядкованих у вигляді сітки, яка містить значення пікселів, щоб позначити, наскільки яскравим і якого кольору є кожен піксель На рисунку 3.6 наведений приклад представлення зображення у вигляді матриці.

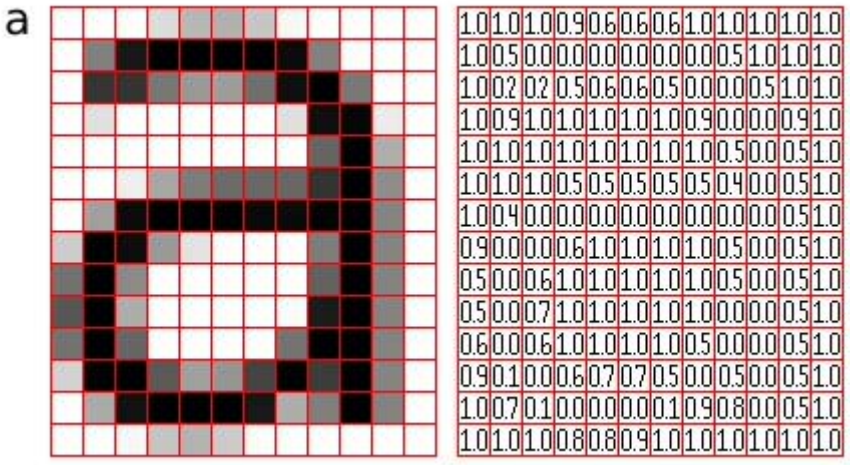

Рисунок 3.6 — Представлення зображення у вигляді сітки пікселів

Архітектура CNN аналогічна структурі підключення нейронів у людському мозку та була натхненна організацією зорової кори. Окремі нейрони реагують на стимули лише в обмеженій області поля зору, відомого як рецептивне поле. Набір таких полів перекривається, щоб охопити всю візуальну область.

CNN зазвичай має три рівня: згортковий рівень, рівень об’єднання та повністю зв’язаний рівень. На рисунку 3.7 представлена архітектура CNN мережі.

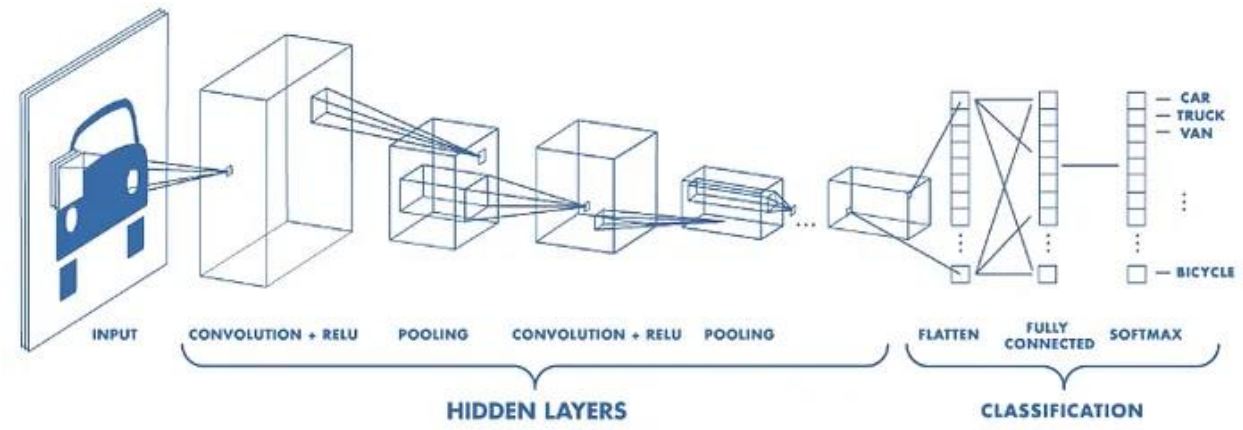

Рисунок 3.7 — Архітектура CNN

Рівень згортки є основним будівельним блоком CNN. Він несе основну частину обчислювального навантаження мережі.

На цьому етапі відбувається "скалярний добуток" між двома матрицями. Одна з цих матриць містить параметри, які можна навчити, і її називають ядром. Друга матриця представляє обмежену область вхідних даних, яку називають частиною сприйнятливого поля. Ядро є просторово меншим, ніж саме зображення, але має більшу глибину. Це означає, що якщо у зображенні є, скажімо, три кольорові канали (червоний, зелений, синій), то висота і ширина ядра будуть невеликими в просторовому плані, але його глибина охоплює всі три канали одночасно.

Під час прямого проходу ядро проковзує по висоті та ширині зображення, формуючи відображення цієї сприйнятливої області. Цей процес породжує двовимірне представлення зображення, відоме як карта активації. Карта активації відображає відповіді ядра на кожну просторову позицію в зображенні, приклад згортання зображений на рисунку 3.8.[15]

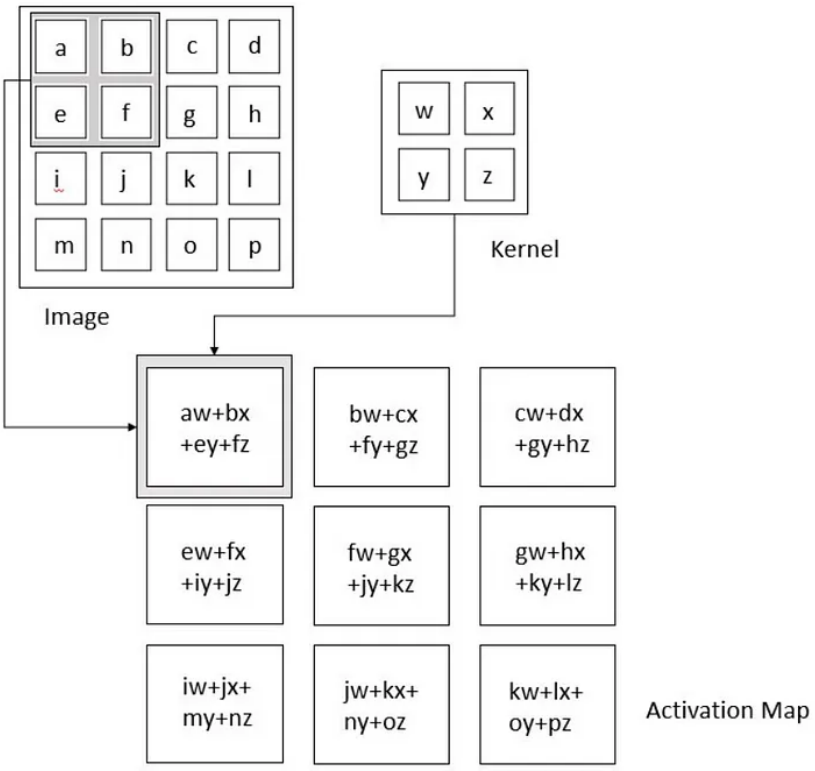

Рисунок 3.8 — Операція згортання

Подібно до згорткового рівня, рівень об’єднання відповідає за зменшення просторового розміру згорнутого об’єкта. Це до зменшити обчислювальну потужність, необхідну для обробки даних через зменшення розмірності. Крім того, це корисно для виділення домінантних рис які є обертальними та позиційними інваріантами, таким чином підтримуючи процес ефективного навчання моделі.

Згортковий рівень та рівень об'єднання спільно формують і-й рівень глибокої нейронної мережі. Залежно від складності зображень, кількість таких шарів може бути збільшена, щоб захопити ще більше деталей на низькому рівні, хоча це супроводжується збільшенням обчислювального завдання.

Тепер, коли ми адаптували наше вхідне зображення до відповідного формату для нашого багаторівневого персептрона, ми конвертуємо його у вектор-стовпець. Отриманий вирівняний результат передається в нейронну мережу прямого зв’язку, а зворотне розповсюдження застосовується на кожній ітерації процесу навчання. Протягом серії епох модель може визначати визначальні та конкретні риси низького рівня в зображеннях та проводити їх класифікацію.

Ключовою перевагою даної концепції є високий ступінь автоматизації у процесі виявлення характерних ознак об'єктів за допомогою нейромережі. Одна з важливих переваг полягає в тому, що нам вже не потрібно ручно відбирати характеристики об'єктів. Ми можемо вдівати нейромережу безліччю зображень різних об'єктів, надаючи їй можливість автоматично розпізнавати та аналізувати їхні характеристики. Наприклад, шляхом подачі мережі мільйонів зображень разом з відповідними анотаціями, вона здатна самостійно генерувати карту ознак, визначаючи ключові атрибути об'єктів. Такий підхід дозволяє нейромережі вчитися автономно та ефективно, а також розпізнавати різноманітні об'єкти без необхідності ручного втручання.[16]

3.2.3 RNN

Рекурентні нейронні мережі - є спеціальним типом штучних нейронних мереж, спроектованим для ефективної обробки даних часових рядів або послідовних даних. Зазвичай, звичайні нейронні мережі прямого зв'язку придатні для аналізу незалежних точок даних. Проте, коли маємо справу з послідовностями, де кожна точка даних залежить від попередньої, потрібно адаптувати нейронну мережу для врахування цих взаємозалежностей.

RNN впроваджують концепцію "пам'яті", яка дозволяє їм зберігати стани або інформацію з попередніх вхідних даних. Це сприяє створенню наступного виходу у послідовності, враховуючи контекст та взаємозв'язки між різними точками даних. За допомогою цієї "пам'яті" RNN можуть більш ефективно моделювати та узагальнювати інформацію з послідовних вхідних даних для передбачення наступних кроків у послідовності.

Простий рекурентний нейронний вузол (RNN) включає в себе контур зворотного зв’язку, що надає йому унікальні можливості в аналізі послідовних даних на рисунку 3.9 зображена архітектура контуру зворотного зв’язку . Контур зворотного зв’язку дозволяє інформації циркулювати в системі, створюючи можливість зберігання та використання попередніх станів у подальших обчисленнях.

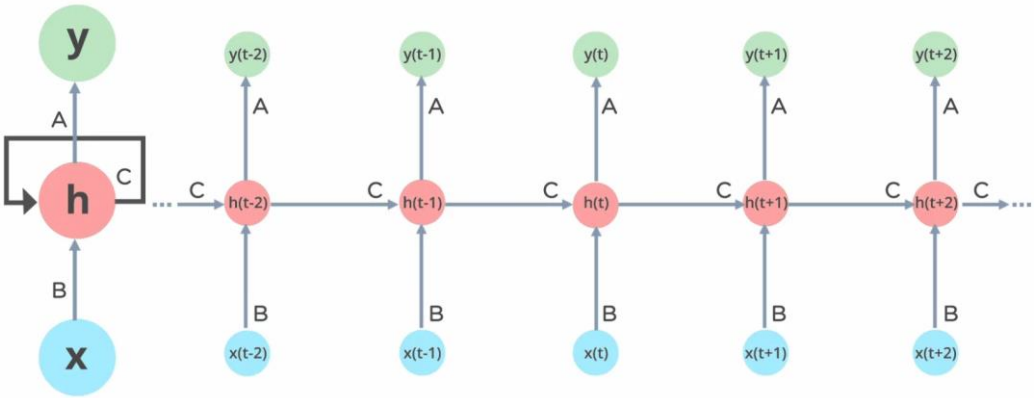

Рисунок 3.9 — Контур зворотного зв’язку

Коли вхідні дані подаються до простого RNN, він враховує не лише поточний вхід, а й інформацію з попередніх кроків послідовності. Кожен вузол у RNN здатний зберігати внутрішні стани, які можуть представляти попередні підсумки обчислень. Під час зворотного проходження, внутрішні стани передаються назад у часі, допомагаючи оптимізувати ваги мережі з урахуванням контексту попередніх вхідних даних.

Такий контур зворотного зв’язку дозволяє простим RNN ефективно моделювати та використовувати завдання залежностей в часі, що робить їх потужним інструментом для обробки послідовних даних у різних областях, таких як мовознавство, обробка сигналів та прогнозування часових рядів.

Архітектура RRN може відрізнятися залежно від проблем, яку потрібно вирішити, Нижче наведені приклади RNN архітектур:

- Один до одного — традиційна модель нейронних мереж, в якому лише одна пара.
- Один до багатьох — має один вхід у мережу та безліч виходів. Має застосування у сфері створення музики.
- Багато до одного — у цьому сценарії ми маємо безліч входів, і лише один вихід. Аналіз настрою є прикладом таких мереж.
- Багато до багатьох — цей тип приймає послідовність входів і генерує послідовність виходів. Машинний переклад є одним з таких прикладів.

Рекурентні нейронні мережі (RNN) вирізняються своєю здатністю ефективно опрацьовувати послідовні дані, такі як мовлення, часові ряди або музичні композиції. Завдяки зворотньому зв'язку, вони можуть враховувати інформацію з попередніх кроків у послідовності, дозволяючи ефективно використовувати контекст та розуміти динаміку даних.

Окрім того, RNN володіють внутрішньою пам'яттю, яка дозволяє їм зберігати стани або певні аспекти історії вхідних даних. Ця можливість зберігання станів виявляється корисною в задачах, де важлива інформація має

довготривалу важливість, що робить RNN потужним інструментом для обробки різноманітних послідовних даних.[17]

3.3 LSTM модель

LSTM (довгострокова короткострокова пам'ять) представляє собою архітектуру рекурентної нейронної мережі (RNN), широко використовувану в глибокому навчанні. Ця модель виникає в результаті вдалого поєднання здатностей рекурентних мереж та зворотного зв'язку, що робить її ідеальною для вирішення завдань, пов'язаних із передбаченням послідовностей.

Основна перевага LSTM полягає в його властивості ефективно фіксувати та використовувати довготривалі залежності в даних. Це стає особливо корисним у завданнях, де необхідно аналізувати та прогнозувати складні взаємозв'язки в послідовних даних.

Відмінності від традиційних нейронних мереж полягають у включенні зворотного зв'язку, що дозволяє LSTM обробляти не лише окремі точки даних, але й взаємодіяти з повними послідовностями. Це дозволяє моделі більш повно та детально розглядати вхідні дані, що робить її дуже ефективною для розуміння та прогнозування моделей у послідовних даних, таких як часові ряди, тексти та мова.

На високому рівні LSTM працює дуже схоже на комірку RNN. Архітектура LSTM складається з ключового елемента - блоку пам’яті, також відомого як блок LSTM. Кожен блок LSTM включає чотири нейронні мережі прямого зв'язку, що формують його комплексну структуру. Кожна з цих мереж складається з вхідного та вихідного рівнів, де кожен вхідний нейрон взаємодіє з усіма вихідними нейронами.

Така конфігурація дозволяє блоку LSTM ефективно моделювати та обробляти велику кількість взаємодій між вхідними та вихідними сигналами. За

рахунок чотирьох повністю з'єднаних шарів, блок LSTM забезпечує глибокий та детальний аналіз вхідних даних, що дозволяє йому успішно розпізнавати та використовувати складні взаємозв'язки в послідовних інформаційних потоках.

Три із чотирьох нейронних мереж прямого зв'язку відповідають за вибір та управління інформацією через ворота забуття, вхідні ворота та вихідні ворота. Кожен з цих шлюзів відповідає за конкретну операцію керування пам'яттю: видалення інформації з пам'яті (забутий шлюз), введення нової інформації в пам'ять (вхідний шлюз) та використання інформації, що зберігається в пам'яті (вихідний шлюз).

Четверта нейронна мережа, відома як пам'ять-кандидат, призначена для створення нової інформації-кандидата, яка впроваджується в пам'ять. Цей механізм дозволяє системі ефективно оновлювати та розширювати інформаційний контекст, забезпечуючи гнучкість та точність у роботі з послідовностями даних. На рисунку 3.10 наведена архітектура блоку LSTM.

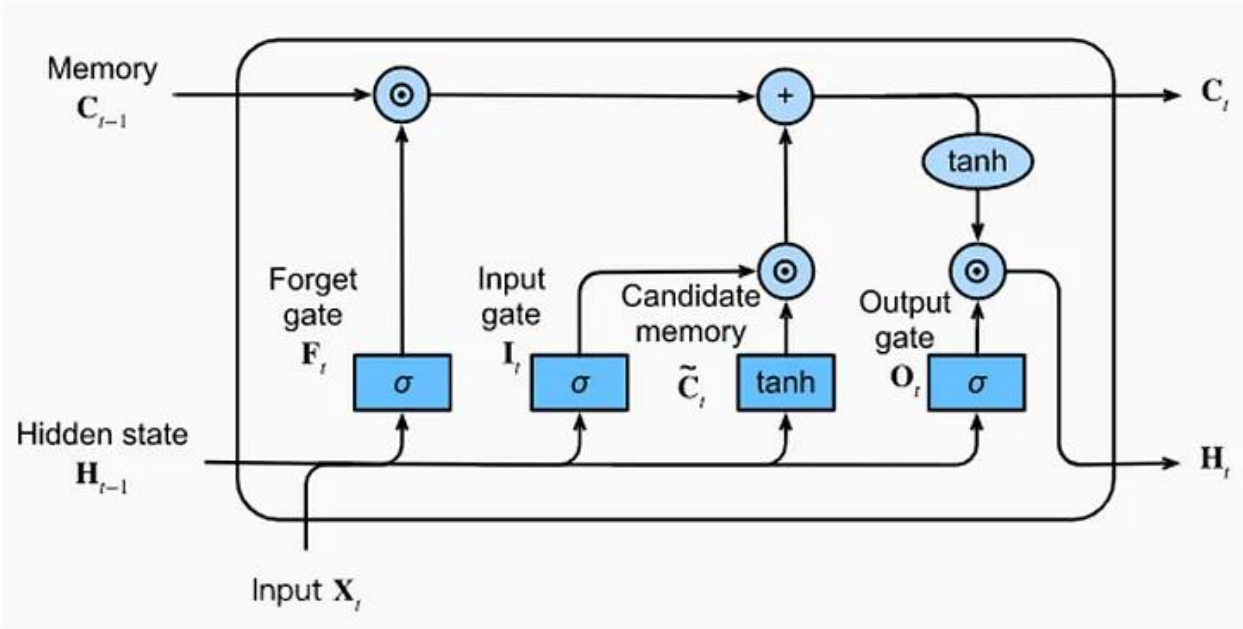

Рисунок 3.10 — Архітектура блоку LSTM

Блок LSTM отримує три вектори як вхідні дані. Два вектори надходять із самого LSTM і були згенеровані LSTM у попередній момент. Блок регулює, через

ворота, внутрішній потік інформації та перетворює значення стану комірки та векторів прихованого стану. Вектори, які будуть частиною вхідного набору LSTM у наступний момент. Контроль інформаційного потоку здійснюється таким чином, що стан комірки діє як довготривала пам’ять, тоді як прихований стан діє як короткочасна пам’ять.

Три ворота (ворота забуття, ворота введення та вихід) є інформаційними селекторами. Їхнє завдання — створювати вектори-селектори. Вектор-селектор — це вектор із значеннями від нуля до одиниці та близькими до цих двох крайніх значень.

Вектор-селектор створюється для поелементного множення іншого вектора того самого розміру. Це означає, що якщо вектор-селектор має значення 0 в певній позиції, то відповідна інформація в іншому векторі на цій позиції повністю відсутня при множенні. Якщо ж вектор-селектор має значення 1, то відповідна інформація в іншому векторі на цій позиції залишається без змін при множенні.

Усі три ворота є нейронними мережами, які використовують сигмоподібну функцію як функцію активації на вихідному рівні. Сигмоїдна функція використовується для отримання на виході вектора, що складається зі значень між нулем і одиницею та поблизу цих двох крайніх значень:

- Ворота забуття — На цьому етапі визначається, яку інформацію необхідно зберегти або вилучити з попереднього стану пам'яті, враховуючи при цьому поточний вхід. Забезпечує відсіювання непотрібної інформації.
- Ворота введення — визначається, яку нову інформацію слід включити в стан пам'яті та як саме стан пам'яті оновлюється. Оновлення здійснюється на основі ворота забуття та поточного входу, що дозволяє системі адаптуватися до нових умов та зберігати актуальну інформацію.
- Вихід — визначається, який вихід генерується на основі оновленого стану пам'яті, узгоджуючи його з поточним входом. Це дозволяє системі

правильно реагувати на поточні умови та виражати результати обробки вхідних даних.

LSTM став не лише потужним інструментом, але й справжнім каталізатором для революції у сфері штучного інтелекту та глибокого навчання. Його унікальна здатність до виявлення та використання довготривалих залежностей в даних визначає його як ключовий елемент для досягнення проривів у різних галузях.

Цей алгоритм дозволяє ефективно аналізувати складні структури та взаємозв'язки в послідовних даних, що розкриває раніше недооцінену інформацію. Застосування LSTM відбувається в широкому спектрі областей, включаючи фінанси, медицину, науку про дані та інші. У фінансовому секторі, наприклад, LSTM використовується для прогнозування ринкових тенденцій та оптимізації портфеля, тоді як у медицині він може допомагати в прогнозуванні розвитку захворювань на основі медичних даних.

Завдяки своїй універсальності і надзвичайній ефективності, LSTM збагачує різноманітні галузі знань, відкриваючи нові можливості для використання послідовних даних у розумінні, прогнозуванні та прийнятті стратегічних рішень. Його роль у розвитку інтелектуальних систем визнається як ключовий елемент, що визначає сучасний напрямок розвитку глибокого навчання та штучного інтелекту. Безперечною перевагою LSTM є їхні виняткові здатності в управлінні довгостроковими залежностями в даних, що робить їх особливо важливими для завдань, де важлива контекстуальна інтерпретація і довгострокові взаємозв'язки між даними [18]

3.4 Навчальні мережі

Нейронна мережа ефективно навчається через ітераційний процес, який включає кілька ключових етапів. На кожній ітерації шари моделі проймають обчислення виходу для кожного навчального прикладу у пакеті даних. Слідом за цим відбувається додатковий прохід через шари для визначення впливу кожного

параметра на кінцевий результат. Це досягається шляхом обчислення градієнта відносно кожного параметра.

Середній градієнт для партії, параметрів та певного стану оптимізації обчислюється, а потім передається в алгоритм оптимізації, наприклад, Адам. Саме цей алгоритм визначає параметри наступної ітерації, що призводить до трохи кращої продуктивності моделі на вашому наборі даних. Новий стан оптимізації для кожного параметра також обчислюється.

З кожним повторенням процесу на навчальних пакетах даних, модель поступово розвивається та удосконалюється, демонструючи точніші результати. Цей постійний цикл навчання взаємодіє з алгоритмами оптимізації, градієнтами та параметрами, сприяючи вдосконаленню моделі та її здатностей до ефективної обробки та узагальнення різноманітних завдань.

Стохастичний градієнтний спуск (SGD) - це ключовий алгоритм оптимізації в навчанні нейромереж. Процес починається з ініціалізації параметрів мережі, таких як ваги та зсуви. На кожному кроці SGD випадковим чином обирає навчальний приклад, обчислює градієнт функції втрат відносно параметрів та коригує їх, використовуючи величину кроку, відому як швидкість навчання. Цей процес повторюється для кожного нового прикладу, що дозволяє SGD ефективно оптимізувати параметри нейромережі та забезпечує його здатність працювати з великими обсягами даних. Однак, слід враховувати можливі проблеми, такі як шум та збіжність, які можуть вимагати додаткових модифікацій для покращення ефективності. На рисунку 3.11 наведений графік алгоритму градієнтного спуску. [19]

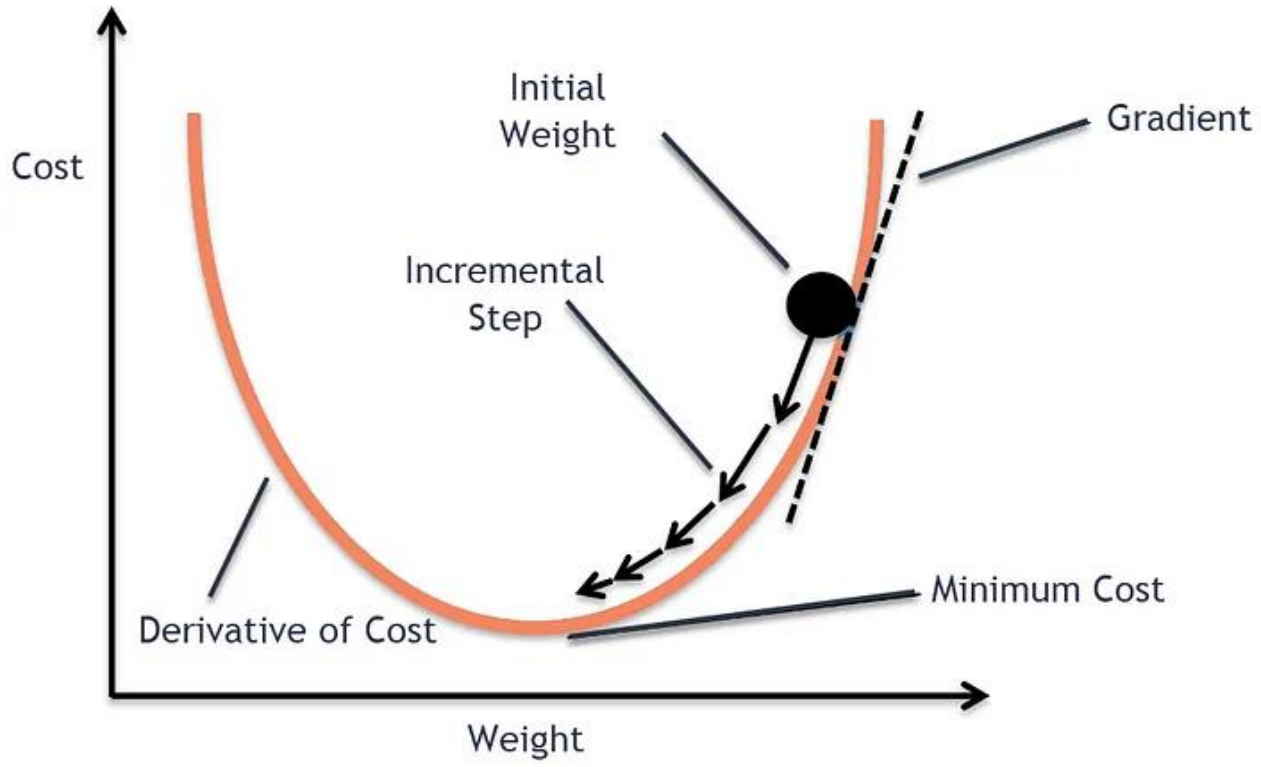

Рисунок 3.11 — Алгоритм градієнтного спуску

Алгоритм Адам (Adam) представляє собою вдосконалення стохастичного градієнтного спуску , спрямоване на подолання деяких його обмежень. Adam поєднує в собі ідеї методів моментів та адаптивного швидкісного градієнта, що робить його ефективнішим у випадках, коли SGD може стикатися з труднощами.

Головні переваги Адама включають адаптивний вибір швидкості навчання для кожного параметра та автоматичне коригування масштабів градієнтів. Він зберігає експоненційно згладжені середні моментів (середніх) та другі моменти градієнтів для кожного параметра, дозволяючи автоматично адаптувати швидкість навчання для кожного параметра окремо. Це особливо корисно в задачах з нерівномірною градієнтною структурою.

Алгоритм оптимізації Adam може виявити високу ефективність в ситуаціях, де маємо великі об'єми даних. Його адаптивна швидкість навчання дозволяє ефективно пристосовувати швидкість навчання для кожного параметра окремо, що є корисним у великих наборах даних із різноманітною структурою.

Крім того, Adam може бути особливо ефективним у завданнях зі складними градієнтами, такими як завдання з багатьма локальними мінімумами чи ярко вираженими градієнтними змінами. На рисунку 3.12 можемо побачити графік переваги Adam у порівнянні з іншими алгоритмами оптимізації багатошарових мереж.

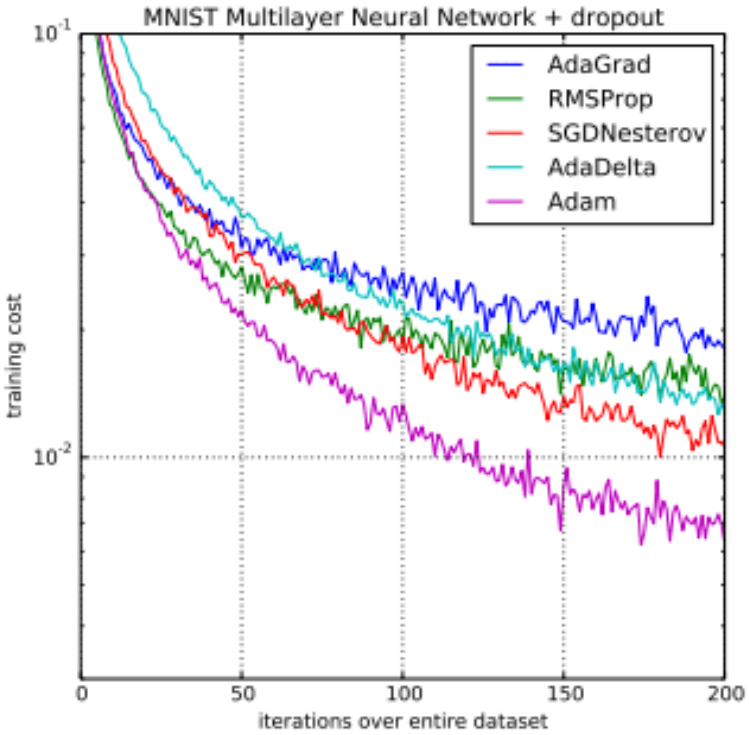

Рисунок 3.12 — Порівняння Adam з іншими алгоритмами

У випадках, коли важлива регуляризація та управління перенавчанням, Adam може також виявити переваги, завдяки врахуванню елементів природної регуляризації через історію квадратів градієнтів. Важливо зауважити, що вибір між Adam та іншими алгоритмами оптимізації варіюється залежно від конкретного контексту, і ефективність кожного алгоритму рекомендується експериментально перевіряти в контексті конкретної задачі та об'єму даних.[20]

3.5 Використання нейронних мереж для виявлення загроз

З виникненням нових технологій та розвитком цифрового віку в сучасному світі, динаміка кіберзагроз постійно еволюціонує, ставлячи під загрозу як особисту, так і корпоративну безпеку. В цьому стрімкому темпі технологічного прогресу необхідність вдосконалення методів виявлення та захисту від потенційних небезпек стає насущною.

Системи виявлення вторгнень (ID) мають вирішальне значення для кібербезпеки, оскільки вони допомагають організаціям швидко виявляти та припиняти злочинну діяльність у своїх комп'ютерних мережах, зменшуючи ризики, пов'язані з кібербезпекою. Атаки можуть призвести до незаконного доступу до мережі, який може бути використаний для викрадення особистих даних, інтелектуальної власності та інших важливих корпоративних даних. Системи ідентифікації можуть аналізувати та автоматично виділяти ознаки вторгнення, допомагаючи виявляти та запобігати кібератакам.

Застосування нейронних мереж, зокрема глибоких нейронних мереж, виявляється важливою стратегією в цьому контексті. Однією з основних переваг є їхній потенціал для аналізу великого обсягу різноманітних даних, що виникають з різних джерел. Ці мережі мають унікальну здатність вивчати складні залежності і патерни, які можуть залишатися непоміченими іншими методами.

Нейронні мережі дозволяють виявляти навіть тонкі та хитростворені атаки, що можуть викликати серйозні наслідки для безпеки. Здатність автоматичного вивчення та адаптації робить їх ефективним інструментом для боротьби з різними загрозами, які постійно змінюються.

Необхідність використання нейронних мереж обумовлена не лише зростанням обсягу даних, але й збільшенням складності атак, які тепер можуть використовувати штучний інтелект для маскування своєї активності. Це вимагає від сучасних систем виявлення загроз більш високої рівня інтелекту та гнучкості, яку можуть надати саме нейронні мережі.

Нейронні мережі виявляють значний потенціал у протидії DDoS атакам завдяки їх здатності аналізувати та розрізняти мережевий трафік. Вони можуть

навчитися розпізнавати типові патерни поведінки та ідентифікувати аномальні зміни, що можуть свідчити про спроби атаки. Алгоритми глибокого навчання в нейронних мережах дозволяють автоматично виявляти аномалії та швидко реагувати на зміни в мережевому трафіку.

Моделі поведінки, створені нейронними мережами, можуть служити базою для виявлення атак, що відрізняються від типових звичаїв. Наприклад, якщо мережевий трафік раптово зростає або виникає велика кількість запитів з конкретних джерел, нейронна мережа може визначити ці аномалії та ініціювати заходи для обмеження або блокування неправомірного трафіку. Такий підхід робить нейронні мережі ефективним інструментом у сфері кібербезпеки для протидії розподіленим атакам з обмеженням доступу.

Запропонована модель ідентифікації (ID) на основі архітектури LSTM RNN (рекурентних нейронних мереж з довгостроковою пам'яттю) виявилася передовою у порівнянні зі звичайними RNN, пропонуючи покращену точність та меншу частоту помилкових тривог. Використання LSTM дозволяє ефективніше моделювати залежності в довгострокових послідовностях, що особливо важливо для завдань ідентифікації, де важливі контекстуальні зв'язки.

Професійна продуктивність запропонованої моделі проявляється у високій точності виявлення, швидкості обробки та низькій частоті помилкових тривог. Ці аспекти роблять її не тільки ефективною, а й надійною у реальних умовах виявлення аномальної активності. Особливо слід відзначити, що дизайн RNN-LSTM виявився найбільш надійним для оцінки різноманітних типів атак, таких як атаки користувача до root (U2R), віддаленого користувача (R2U), зондування (Probe) і атаки через відмову в обслуговуванні (DoS) у складних мережевих оточеннях. Це робить запропоновану модель ідеальним вибором для задач ідентифікації та протидії кіберзагрозам у сучасних мережах.

Запропонована модель ідентифікатора, заснована на архітектурі LSTM RNN, може бути застосована до різних типів систем, оскільки вона розроблена, щоб оцінити, чи поведінка мережі змінена чи нормальна на основі попередніх

спостережень. Модель навчена та перевірена за допомогою набору даних NSL-KDD, який містить різноманітні вектори мережевих з’єднань. Модель LSTM може правильно вивчати функції, які були взяті з набору даних під час навчання, дозволяючи моделі точно розрізняти звичайний трафік і мережеві атаки. Таким чином, запропонована модель ID може бути застосована до різних типів систем, якщо модель навчена та протестована з використанням відповідного набору даних.[21]

3.6 Висновки за розділом

У розділі, присвяченому аналізу технології штучного інтелекту, були ретельно розглянуті ключові архітектури, зокрема Convolutional Neural Networks (CNN) і Recurrent Neural Networks (RNN), із фокусом на Long Short-Term Memory (LSTM) моделі. Цей аналіз дозволив визначити основні переваги та обмеження кожної архітектури в контексті обробки послідовних даних.

Зазначена архітектура LSTM RNN відзначається винятковою здатністю управляти довгостроковими залежностями в послідовних даних. Це особливо важливо в задачах, де взаємозв'язки між даними в часі визначають правильний контекст інтерпретації. Додатково, LSTM проявляє високу ефективність у вирішенні завдань прогнозування та аналізу послідовних шаблонів, роблячи його перспективним інструментом для різних сфер застосування.

Враховуючи вищезазначене, вибір LSTM RNN як подальшої моделі для нейромережі має під собою обґрунтовану основу. Його вдосконалені можливості адаптації до різноманітних структур даних та здатність до аналізу довгострокових залежностей надають цілеспрямовану інструментальну базу для розв'язання складних задач, пов'язаних із послідовними даними. Подальше використання LSTM RNN в рамках досліджень та розробок в галузі штучного

інтелекту обіцяє прискорення розвитку інтелектуальних систем і вдосконалення їхньої адаптивності та точності в різноманітних сценаріях застосування.

РОЗДІЛ 4
ВИЯВЛЕННЯ DDOS АТАКИ ЗА ДОПОМОГОЮ НЕЙРОННОЇ МЕРЕЖІ

4.1 Підготовка моделі

Для реалізації проекту будемо використовувати мову програмування Python, і використанням бібліотеки машинного навчання TensorFlow.

Python – це мова програмування високого рівня з інтерпретацією, орієнтована на об'єкти, що володіє динамічною семантикою. Її вбудовані високорівневі структури даних, у поєднанні з динамічною типізацією та динамічним зв'язуванням, роблять її привабливою для швидкого розроблення програм та використання як мови сценаріїв чи інструменту для об'єднання існуючих компонентів. Простий синтаксис Python, який легко вивчити, сприяє читабельності і, таким чином, зменшує витрати на обслуговування програм. Мова підтримує модулі та пакунки, що сприяють модульності програм та повторному використанню коду. Інтерпретатор Python та обширна стандартна бібліотека доступні у вихідній або двійковій формі безкоштовно для різних платформ і можуть вільно розповсюджуватися.[22]

TensorFlow є вільною та безкоштовною бібліотекою, спрямованою на машинне навчання та штучний інтелект, з основним акцентом на глибинному навчанні. Цю бібліотеку спочатку створила компанія Google для власного використання, а потім вона була відкрита для широкої громадськості. У сучасний час TensorFlow переріс у повноцінну платформу для машинного навчання, що пропонує різноманітні інструменти для тренування мереж, їх конструювання та використання. У подальших розділах тексту будуть представлені фрагменти коду на мові Python з використанням TensorFlow.

В проекті також використовуються наступні бібліотеки:

- pandas — програмна бібліотека призначена для маніпулювання даними і також їх подальшого аналізу.

- numpy — розширення для Python, яка додає підтримку багатовимірних масивів і матриць, і також має бібліотеку високорівневих математичних функції.
- matplotlib — бібліотека для візуалізації даних двовимірною графікою.
- sklearn — є також бібліотекою машинного навчання, яка надає функціонал для створення та тренування різних алгоритмів, класифікації, кластеризації та регресії.
- keras — відкрита бібліотека яка забезпечує взаємодію з штучними нейронними мережами

4.2 Підготовка і обробка даних

В якості набору даних для дослідження вибрано NSL-KDD з Канадського Інституту з Кібербезпеки [23]. Він включає в себе 125 973 пакети, які містять журнали 22 різних типів атак. Він містить 42 ознаки з мітками. Мітка показує, чи є пакет нормальний чи ні.

Щоб набір даних NSL-KDD був чистим для подальшого навчання і тестування, тобто не містив непотрібних ознак або рядкових значень, які ідентифікують трафік як «нормальний» або «аномальний». Проводиться попереднє перетворення даних.

Для навчання LSTM-моделі необхідні два різних набори тренування. Один з них це X-потяг, який містить характеристики пакетів, а інший Y-потяг, який містить мітки кожного з пакетів, тобто з класифікацією, що це пакет атаки. Для тестування також використовується X-тест і Y-тест, які містять 40% випадкових пакетів з навчального набору. Тестовий набір містить випадкові атакуючи пакети, щоб можна було об’єктивно виміряти точність моделі.

Набір даних NSL-KDD містить 42 характеристики. Це такі значення як тип протоколу, прапор, тип атаки, сервіс тощо. На рисунку 4.1 можна побачити повний перелік всіх наявних характеристик пакетів.

The 42 features of NSL-KDD Dataset.					
S/N	Field Name	Description	S/N	Field Name	Description
1	duration	continuous	22	is_guest_login	symbolic
2	protocol type	symbolic	23	count	continuous
3	service	symbolic	24	srv_count	continuous
4	flag	symbolic	25	serror_rate	continuous
5	src_bytes	continuous	26	rerror_rate	continuous
6	dst_bytes	continuous	27	srv_serror_rate	continuous
7	Land	symbolic	28	srv_rerror_rate	continuous
8	wrong_fragment	continuous	29	same_srv_rate	continuous
9	urgent	continuous	30	diff_srv_rate	continuous
10	Hot	continuous	31	srv_diff_host_rate	continuous
11	num_failed_logins	continuous	32	dst_host_count	continuous
12	logged_in	symbolic	33	dst_host_srv_count	symbolic
13	num_compromised	continuous	34	dst_host_same_srv_rate	continuous
14	root_shell	continuous	35	dst_host_diff_srv_rate	continuous
15	su_attempted	continuous	36	dst_host_same_src_port_rate	continuous
16	num_root	continuous	37	dst_host_srv_diff_host_rate	continuous
17	num_file_creations	continuous	38	dst_host_serror_rate	continuous
18	num_shells	continuous	39	dst_host_srv_serror_rate	continuous
19	num_access_file	continuous	40	dst_host_rerror_rate	continuous
20	num_outbound_cmds	continuous	41	dst_host_srv_rerror_rate	continuous
21	is_host_login	symbolic	42	Attack type	label

Рисунок 4.1 — Характеристика пакеті

Для подальшого навчання моделі, ми будемо виявляти саме DDoS атаки на модель, але також набір даних NSL-KDD має загалом 22 різни типів атак. Кожна атака має різні значення ознак атаки які зображенні на рисунку 4.2.

The 22 Attack Types of NSL-KDD Dataset.					
S/N	Name	Attack Type	S/N	Name	Attack Type
1	**back**	**DoS**	12	perl	u2r
2	buffer_overflow	u2r	13	phf	r2l
3	ftp_write	r2l	14	**pod**	**DoS**
4	guess_passwd	r2l	15	portsweep	probe
5	imap	r2l	16	rootkit	u2r
6	ipsweep	probe	17	satan	probe
7	**land**	**DoS**	18	**smurf**	**DoS**
8	loadmodule	u2r	19	spy	r2l
9	multihop	r2l	20	**teardrop**	**DoS**
10	**neptune**	**DoS**	21	warezclient	r2l
11	nmap	probe	22	warezmastert	r2l

Рисунок 4.2 — Типи атак в набору даних

4.3 Аналіз пакетів

TensorFlow має чудові бібліотеки, які містять RNN, DNN та багато різних методів машинного навчання. Для цього дослідження було використано Tensorflow Keras. Для глибокої нейронної мережі було використано 2 прихованих шари, а для перевірки втрат під час навчання було використано метод середньої абсолютної помилки. Для оптимізації DNN використовувався метод «adam». Наступний результати показані втрати вибірки та точність для набору даних, навченого методом LSTM.

```
Train on 125973 samples, validate on 52977 samples
Epoch 1/30
125973/125973 [==============================] - 19s 147us/sample - loss: 0.0531
- acc: 0.9619 - val_loss: 0.0517 - val_acc: 0.9701
Epoch 2/30
125973/125973 [==============================] - 17s 132us/sample - loss: 0.0379
- acc: 0.9737 - val_loss: 0.0347 - val_acc: 0.9808
Epoch 3/30
125973/125973 [==============================] - 17s 132us/sample - loss: 0.0304
- acc: 0.9856 - val_loss: 0.0278 - val_acc: 0.9867
Epoch 4/30
125973/125973 [==============================] - 17s 133us/sample - loss: 0.0266
- acc: 0.9883 - val_loss: 0.0270 - val_acc: 0.9880
Epoch 5/30
```

```
125973/125973 [==============================] - 17s 132us/sample - loss: 0.0238
- acc: 0.9897 - val_loss: 0.0248 - val_acc: 0.9896
Epoch 6/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0223
- acc: 0.9897 - val_loss: 0.0228 - val_acc: 0.9907
Epoch 7/30
125973/125973 [==============================] - 18s 142us/sample - loss: 0.0208
- acc: 0.9901 - val_loss: 0.0211 - val_acc: 0.9908
Epoch 8/30
125973/125973 [==============================] - 17s 139us/sample - loss: 0.0198
- acc: 0.9905 - val_loss: 0.0176 - val_acc: 0.9912
Epoch 9/30
125973/125973 [==============================] - 17s 135us/sample - loss: 0.0186
- acc: 0.9909 - val_loss: 0.0182 - val_acc: 0.9915
Epoch 10/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0180
- acc: 0.9913 - val_loss: 0.0171 - val_acc: 0.9919
Epoch 11/30
125973/125973 [==============================] - 17s 133us/sample - loss: 0.0173
- acc: 0.9914 - val_loss: 0.0158 - val_acc: 0.9921
Epoch 12/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0166
- acc: 0.9919 - val_loss: 0.0162 - val_acc: 0.9922
Epoch 13/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0160
- acc: 0.9921 - val_loss: 0.0155 - val_acc: 0.9925
Epoch 14/30
125973/125973 [==============================] - 17s 135us/sample - loss: 0.0157
- acc: 0.9923 - val_loss: 0.0176 - val_acc: 0.9926
Epoch 15/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0151
- acc: 0.9924 - val_loss: 0.0152 - val_acc: 0.9931
Epoch 16/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0150
- acc: 0.9925 - val_loss: 0.0141 - val_acc: 0.9928
Epoch 17/30
125973/125973 [==============================] - 17s 136us/sample - loss: 0.0147
- acc: 0.9925 - val_loss: 0.0151 - val_acc: 0.9929
Epoch 18/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0144
- acc: 0.9927 - val_loss: 0.0143 - val_acc: 0.9927
Epoch 19/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0141
- acc: 0.9927 - val_loss: 0.0144 - val_acc: 0.9928
Epoch 20/30
125973/125973 [==============================] - 17s 135us/sample - loss: 0.0140
- acc: 0.9929 - val_loss: 0.0139 - val_acc: 0.9930
Epoch 21/30
125973/125973 [==============================] - 17s 133us/sample - loss: 0.0137
- acc: 0.9930 - val_loss: 0.0141 - val_acc: 0.9932
Epoch 22/30
125973/125973 [==============================] - 17s 133us/sample - loss: 0.0135
- acc: 0.9929 - val_loss: 0.0131 - val_acc: 0.9933
Epoch 23/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0133
- acc: 0.9930 - val_loss: 0.0123 - val_acc: 0.9933
Epoch 24/30
125973/125973 [==============================] - 17s 134us/sample - loss: 0.0132
- acc: 0.9931 - val_loss: 0.0148 - val_acc: 0.9934
Epoch 25/30
```

```
125973/125973 [==============================] - 18s 143us/sample - loss: 0.0131
- acc: 0.9932 - val_loss: 0.0136 - val_acc: 0.9933
Epoch 26/30
125973/125973 [==============================] - 17s 137us/sample - loss: 0.0129
- acc: 0.9933 - val_loss: 0.0144 - val_acc: 0.9934
Epoch 27/30
125973/125973 [==============================] - 17s 137us/sample - loss: 0.0127
- acc: 0.9935 - val_loss: 0.0129 - val_acc: 0.9938
Epoch 28/30
125973/125973 [==============================] - 17s 135us/sample - loss: 0.0127
- acc: 0.9935 - val_loss: 0.0131 - val_acc: 0.9937
Epoch 29/30
125973/125973 [==============================] - 17s 136us/sample - loss: 0.0125
- acc: 0.9936 - val_loss: 0.0141 - val_acc: 0.9940
Epoch 30/30
125973/125973 [==============================] - 17s 136us/sample - loss: 0.0124
- acc: 0.9935 - val_loss: 0.0141 - val_acc: 0.9938

52977/52977 [==============================] - 3s 52us/sample - loss: 0.0141 -
acc: 0.9938

0.014085674540009275 0.9938086
```

Згідно з наступними результатами, з 52977 тестових пакетів вибірки значення втрати склало 0.0141, а значення точності склало 0.9939 відповідно. Цей результат оцінки показує, що LSTM RNN є майже ідеальним методом для запобігання та захисту від DDoS атаки.

4.4 Результати навчання

Після вже проходження 30 епох тренування, на рисунку 4.3 можна побачити надзвичайно низькі втрати моделі. Це означає, що модель ефективно функціонує і виявляє DDoS трафік. Кінцевим значення втрат становить 0.0141.

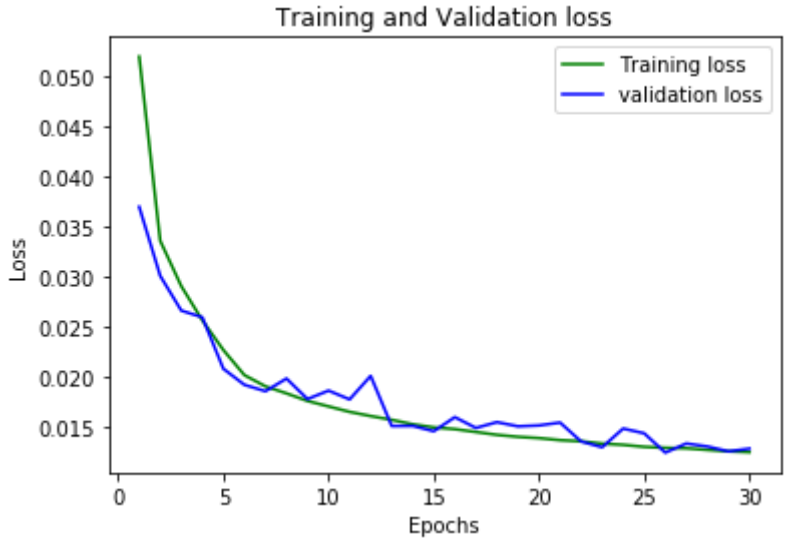

Рисунок 4.3 — Графік втрат при навчанні та валідації

А на рисунку 4.4 можна споглядати точність моделі яка наближається до 99.3%

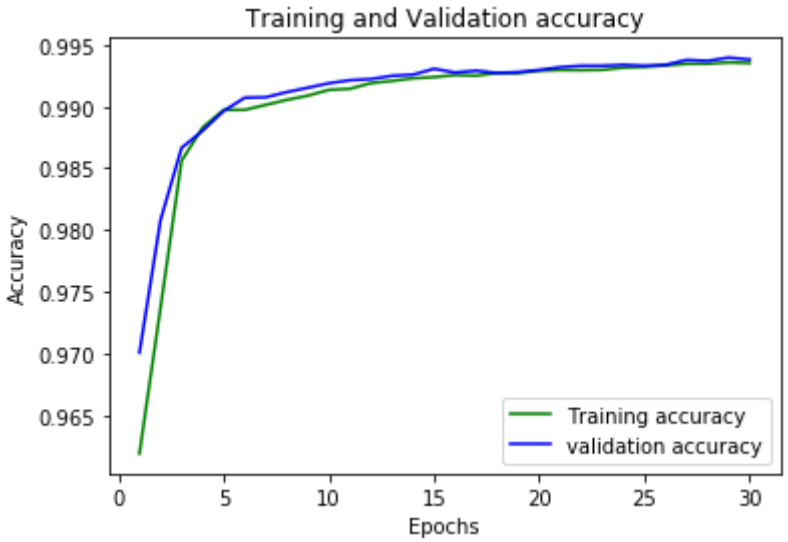

Рисунок 4.4 — Графік точності навчання та валідації

Після навчання моделі потрібно провести тестові моделювання на основі 1000 пакетів для визначення результату. Результати роботи програми приведений на рисунку 4.5.

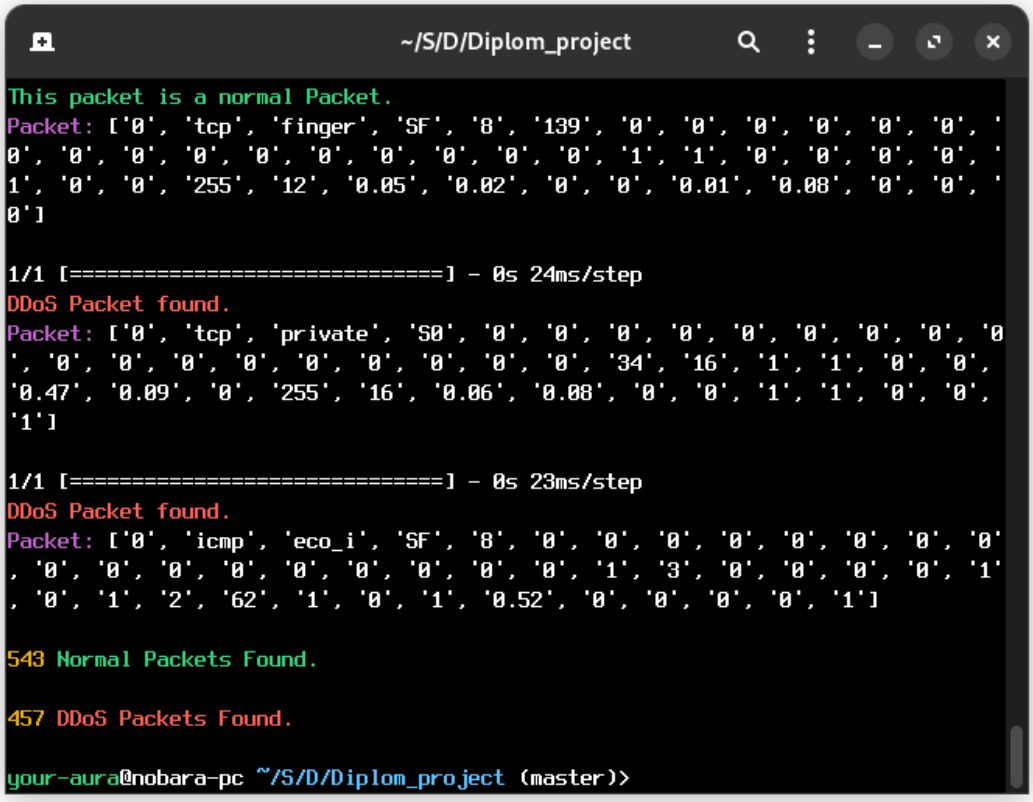

Рисунок 4.5 — Результат тестування

4.5 Висновки за розділом

У результаті проведених досліджень в даному розділі було досягнуто значущого прогресу у розробці та впровадженні програмного рішення, базованого на LSTM RNN моделі, для виявлення аномалій в мережевому трафіку. Основною метою цього проекту було створення системи, яка може ефективно виявляти та реагувати на неправильну або потенційно загрозливу активність у мережі.

Використовуючи широко визнаний набір даних NSL-KDD для проведення досліджень, ми здійснили успішне навчання моделі, досягнувши вражаючої точності на рівні 99.3%. Цей результат свідчить про високий рівень ефективності обраної нейронної мережі в розпізнаванні аномалій в мережевому трафіку.

Важливою особливістю нашого підходу є його застосовність у сфері індустріальних мереж, де ефективність системи була підтверджена.

Під час тестування розробленої моделі на 1000 пакетах мережевого трафіку було виявлено 457 аномальних пакетів, що можна віднести до атак типу DoS. Цей результат свідчить про високу чутливість моделі та її здатність ефективно розпізнавати та реагувати на потенційно шкідливі дії в реальних умовах експлуатації. Отже, наше програмне рішення виявляється не лише надійним, а й здатним забезпечити високий рівень безпеки мережі в умовах постійно зростаючих загроз кібербезпеки.

ВИСНОВКИ

В даній випускній роботі було проведено комплексний аналіз індустріальних мереж, зосереджуючись на ключових аспектах Industry 4.0, IIoT, SCADA та Big Data. Ці технології спрямовані на підвищення автоматизації, ефективності та конкурентоспроможності промислових підприємств. Висновок з першого розділу вказує на необхідність управління кібербезпекою для забезпечення стійкості та конфіденційності виробничих систем.

Другий розділ розглядає мережевий стек інтернету протоколу, класифікацію мережевих загроз та проблем безпеки в контексті Industry 4.0. Зазначено, що з впровадженням цифрових технологій зростає ризик атак, зокрема DDoS-атак, які можуть серйозно загрожувати функціонуванню мереж. Дослідження підкреслює важливість використання штучного інтелекту для виявлення та протидії таким атакам, що вказує на потребу в комплексному підході до мережевої безпеки.

Третій розділ аналізує технології штучного інтелекту, зокрема архітектури CNN і RNN, з фокусом на LSTM моделі. Обрана LSTM RNN як ефективний інструмент для обробки послідовних даних, що вказує на її потенціал для розв'язання складних завдань у різних областях застосування.

У четвертому розділі розроблено програмне рішення на основі LSTM RNN для виявлення аномалій в мережевому трафіку. Навчання моделі на наборі даних NSL-KDD показало високу точність (99.3%), свідчачи про ефективність моделі виявлення аномалій в індустріальних мережах. Тестування підтвердило високу чутливість та ефективність моделі в реальних умовах, здатної реагувати на потенційно шкідливі дії в мережі.

Отже, Ця випускна робота вносить значний внесок у розуміння та вирішення сучасних викликів у сфері кібербезпеки в індустрії 4.0. Спрямована на розвиток концепцій та технологій, які визначають новий етап еволюції виробничого сектору, робота акцентує на тому, що кібербезпека вже не може бути

відокремленим аспектом, але повинна стати невід’ємною частиною стратегії розвитку кожного промислового підприємства. Забезпечення стійкості та безпеки мережі стає критично важливим завданням у світі, де цифрові технології витісняють традиційні методи виробництва.

Комплексний підхід до мережевої безпеки, який включає в себе не тільки аналіз та класифікацію загроз, але й активне використання передових технологій штучного інтелекту, визначається як стратегічно важливий елемент в управлінні кібербезпекою. У світлі нестримного розвитку технологій та постійного зростання кількості кіберзагроз, нейромережі, такі як LSTM RNN, виявляються необхідним інструментарієм для ефективного виявлення та аналізу аномалій в мережевому трафіку. Високий рівень точності та чутливості, отриманий у випробуваннях, підкреслює потенційність застосування цих технологій для розв'язання проблем кібербезпеки в індустріальних умовах. Такий підхід може стати ключовим фактором для забезпечення безпеки та надійності промислових мереж у контексті індустрії 4.0.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A. Bashar Fakhri, S. Latteef Mohammed, I. Khan, A. Safaa Sadiq, B. Alkazemi *et al.*, "Industry 4.0: architecture and equipment revolution," *Computers, Materials & Continua*, vol. 66, no.2, pp. 1175–1194, 2021. URL: https://doi.org/10.32604/cmc.2020.012587
2. Kiravuo, T., Särelä, M., & Manner, J. (2013). A Survey of Ethernet LAN Security. IEEE Communications Surveys & Tutorials, 15, 1477-1491. https://doi.org/10.1109/SURV.2012.121112.00190.
3. Bagri, A., Netto, R., & Jhaveri, D. (2014). Supervisory Control and Data Acquisition. International Journal of Computer Applications, 102, 1-5. https://doi.org/10.5120/17848-8797.
4. Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial Internet of Things: Challenges, Opportunities, and Directions. IEEE Transactions on Industrial Informatics, 14, 4724-4734. https://doi.org/10.1109/TII.2018.2852491.
5. Vitturi, S., Zunino, C., & Sauter, T. (2019). Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G. Proceedings of the IEEE, 107, 944-961. https://doi.org/10.1109/JPROC.2019.2913443.
6. Mourtzis, D., Vlachou, E., & Milas, N. (2016). Industrial Big Data as a Result of IoT Adoption in Manufacturing. Procedia CIRP, 55, 290-295. https://doi.org/10.1016/J.PROCIR.2016.07.038.
7. PLC SCADA SKILLS. (2023, September 2). Ethernet/IP Protocol: enhancing industrial communication. https://www.plcscadaskills.com.au/ethernet-ip-protocol-enhancing-industrial-communication/
8. PLC SCADA SKILLS. (2023a, March 16). UDP/IP vs TCP/IP in IIoT and Industry 4.0: A Comparison. https://www.plcscadaskills.com.au/comparison-of-udp-ip-and-tcp-ip/

9. Prinsloo, J. G., Sinha, S., & Von Solms, B. (2019). A review of Industry 4.0 manufacturing Process Security Risks. Applied Sciences, 9(23), 5105. https://doi.org/10.3390/app9235105
10. What is DDoS Attack - Javatpoint. (n.d.). www.javatpoint.com. https://www.javatpoint.com/what-is-ddos-attack
11. Singh, J., & Tech, S. M. (2020). Mitigating DoS and DDoS based Attacks: An Artificial Intelligence Approach. *International Journal of Innovative Science and Research Technology*, *5*(5). https://www.ijisrt.com/assets/upload/files/IJISRT20MAY319.pdf
12. Alzubi, J. A., Nayyar, A., & Kumar, A. (2018). Machine Learning from Theory to Algorithms: An Overview. Journal of Physics, 1142, 012012. https://doi.org/10.1088/1742-6596/1142/1/012012
13. S. Chaudhary, S. Yadav, S. Kushwaha, Surya Ratan Pratap Shahi. "A brief review of machine learning and its applications." *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology* 12.SUP 1 (2020): 218-223. https://smsjournals.com/index.php/SAMRIDDHI/article/view/1941/870
14. Géron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly.
15. Mishra, M. (2021, December 15). Convolutional neural networks, explained - towards data science. Medium. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939
16. Saha, S. (2023, April 8). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. Medium. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
17. Kalita, D. (2023, November 7). A brief overview of Recurrent Neural networks (RNN). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/

18. Calzone, O. (2022, April 10). An Intuitive explanation of LSTM - Ottavio Calzone - medium. Medium. https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c
19. Lanham, M. (2018). Learn Arcore - Fundamentals of Google Arcore. Packt Publishing.
20. Brownlee, J. (2021, January 12). Gentle introduction to the ADAM Optimization Algorithm for deep learning. MachineLearningMastery.com. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/
21. Ibrahim, M., & Elhafiz, R. (2023). Modeling an intrusion detection using recurrent neural networks. Journal of Engineering Research, 11(1), 100013. https://doi.org/10.1016/j.jer.2023.100013
22. What is Python? Executive Summary. Python.org. https://www.python.org/doc/essays/blurb/
23. NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. https://www.unb.ca/cic/datasets/nsl.html

Додаток А

Зауваження нормоконтролера

Таблиця А.1 – Зауваження нормоконтролера

Сторінка	Позначення	Зміст зауваження
ПЗ		Нещільне заповнення сторінок
ПЗ		Невірні переліки
ПЗ		Невірні посилання на літературу
С. 62-63		Не може бути текстом - рисунок.

Додаток Б

Лістинг навчальної моделі

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import keras

df = pd.read_csv('./KDDTrain.csv')
df.info()
df_val = pd.read_csv('./KDDTest.csv')

# Визначення ознак та розділення даних
num_feat =
['src_bytes','dst_bytes','wrong_fragment','urgent','hot','num_failed_logins','num_comp
romised','root_shell','su_attempted','num_root','num_file_creations','num_shells','num
_access_files','num_outbound_cmds','count','srv_count','serror_rate','srv_serror_rate','
rerror_rate','srv_rerror_rate','same_srv_rate','diff_srv_rate','srv_diff_host_rate','dst_ho
st_count','dst_host_srv_count','dst_host_same_srv_rate','dst_host_diff_srv_rate','dst_h
ost_same_src_port_rate','dst_host_srv_diff_host_rate','dst_host_serror_rate','dst_host
_srv_serror_rate','dst_host_rerror_rate','dst_host_srv_rerror_rate']
feature_df = df
X = np.asarray(feature_df)
```

```
y = np.asarray(df['class'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 1, stratify = y)

#Перетворення функцій з об'єктним або рядковим значенням на числові значення
ord_feat = ['protocol_type', 'service', 'flag']
nom_feat = ['land', 'logged_in', 'is_host_login', 'is_guest_login']
num_feat = ['src_bytes','dst_bytes','wrong_fragment','urgent','hot','num_failed_logins','num_compromised','root_shell','su_attempted','num_root','num_file_creations','num_shells','num_access_files','num_outbound_cmds','count','srv_count','serror_rate','srv_serror_rate','rerror_rate','srv_rerror_rate','same_srv_rate','diff_srv_rate','srv_diff_host_rate','dst_host_count','dst_host_srv_count','dst_host_same_srv_rate','dst_host_diff_srv_rate','dst_host_same_src_port_rate','dst_host_srv_diff_host_rate','dst_host_serror_rate','dst_host_srv_serror_rate','dst_host_rerror_rate','dst_host_srv_rerror_rate']

X_train, y_train = df.drop(columns=['class'], axis=1, inplace=False), df['class'].values

ohe = OneHotEncoder(sparse=False)
oe = OrdinalEncoder()

ohe.fit(X_train[nom_feat].values)
oe.fit(X_train[ord_feat].values)

scalar = StandardScaler()
scalar.fit(X_train[num_feat].values)
X_train_nom = ohe.transform(X_train[nom_feat].values)
X_train_ord = oe.transform(X_train[ord_feat].values)
X_train_num = scalar.transform(X_train[num_feat].values)
```

```
X_train = np.concatenate([X_train_ord, X_train_num, X_train_nom], axis=1)

#Побудова та навчання моделі SVM
from sklearn import svm

classifier = svm.SVC(kernel = "linear")
classifier.fit(X_train, y_train)
y_predict = classifier.predict(X_test)

from sklearn import metrics
print("SVM ACCURACY : ",metrics.accuracy_score(y_test, y_predict))

#перетворити набір даних у 3d-масив
X_train = X_train.reshape((X_train.shape[0],1,X_train.shape[1]))
y_train = y_train.reshape((y_train.shape[0],1,1))

#Масштабування наборів даних num, ord, nom
X_test, y_test = df_val.drop(columns=['class'], axis = 1, inplace=False), df_val['class'].values

ohe.fit(X_test[nom_feat].values)
oe.fit(X_test[ord_feat].values)

#Перетворення тестового набору даних у 3d-масив
scalar.fit(X_test[num_feat].values)
X_test_nom = ohe.transform(X_test[nom_feat].values)
X_test_ord = oe.transform(X_test[ord_feat].values)
X_test_num = scalar.transform(X_test[num_feat].values)
```

```
X_test = np.concatenate([X_test_ord, X_test_num, X_test_nom], axis=1)
X_test.shape
X_test = X_test.reshape((X_test.shape[0],1,X_test.shape[1]))
y_test = y_test.reshape((y_test.shape[0],1,1))

# Побудова та навчання моделі LSTM
model = Sequential()
model.add(LSTM(units = 44, input_shape=(1,44), return_sequences=True))
model.add(Dense(1))
model.add(Dense(1))
model.compile(loss="mean_absolute_error", optimizer = 'adam', metrics = ["accuracy"])

history = model.fit(X_train, y_train, epochs=30, validation_data = (X_test, y_test))# validation_data=(X_test, y_test)

# Збереження та оцінка моделі
model.save("LSTM_DDoS_Detection.model")
val_loss, val_acc = model.evaluate(X_test, y_test)
print(val_loss, val_acc)

#Графік втрат при навчанні та валідації
loss_train = history.history['loss']
loss_val = history.history['val_loss']
epochs = range(1,31)
plt.plot(epochs, loss_train, 'g', label='Training loss')
plt.plot(epochs, loss_val, 'b', label='validation loss')
plt.title('Training and Validation loss')
plt.xlabel('Epochs')
```

```
plt.ylabel('Loss')
plt.legend()
plt.show()

#Графік точності навчання та валідації
loss_train = history.history['acc']
loss_val = history.history['val_acc']
epochs = range(1,31)
plt.plot(epochs, loss_train, 'g', label='Training accuracy')
plt.plot(epochs, loss_val, 'b', label='validation accuracy')
plt.title('Training and Validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

Додаток В

Лістинг програми виявлення загрози

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
from sklearn.model_selection import train_test_split
import tensorflow as tf
from sklearn.preprocessing import StandardScaler
import csv

class color:
   PURPLE = '\033[95m'
   CYAN = '\033[96m'
   DARKCYAN = '\033[36m'
   BLUE = '\033[94m'
   GREEN = '\033[92m'
   YELLOW = '\033[93m'
   RED = '\033[91m'
   BOLD = '\033[1m'
   UNDERLINE = '\033[4m'
   END = '\033[0m'

df = pd.read_csv("attack_test.csv")
with open("attack_test.csv", newline='') as f:
    reader = csv.reader(f)
    origin_X_test = list(reader)
```

```
# Перетворення функцій з об'єктним або рядковим значенням на числові значення
num_feat = ['src_bytes','dst_bytes','wrong_fragment','urgent','hot','num_failed_logins','num_compromised','root_shell','su_attempted','num_root','num_file_creations','num_shells','num_access_files','num_outbound_cmds','count','srv_count','serror_rate','srv_serror_rate','rerror_rate','srv_rerror_rate','same_srv_rate','diff_srv_rate','srv_diff_host_rate','dst_host_count','dst_host_srv_count','dst_host_same_srv_rate','dst_host_diff_srv_rate','dst_host_same_src_port_rate','dst_host_srv_diff_host_rate','dst_host_serror_rate','dst_host_srv_serror_rate','dst_host_rerror_rate','dst_host_srv_rerror_rate']
ord_feat = ['protocol_type', 'service', 'flag']
nom_feat = ['land', 'logged_in', 'is_host_login', 'is_guest_login']
num_feat = ['src_bytes','dst_bytes','wrong_fragment','urgent','hot','num_failed_logins','num_compromised','root_shell','su_attempted','num_root','num_file_creations','num_shells','num_access_files','num_outbound_cmds','count','srv_count','serror_rate','srv_serror_rate','rerror_rate','srv_rerror_rate','same_srv_rate','diff_srv_rate','srv_diff_host_rate','dst_host_count','dst_host_srv_count','dst_host_same_srv_rate','dst_host_diff_srv_rate','dst_host_same_src_port_rate','dst_host_srv_diff_host_rate','dst_host_serror_rate','dst_host_srv_serror_rate','dst_host_rerror_rate','dst_host_srv_rerror_rate']

# Підготовка даних
X_test, y_test = df.drop(columns=['class'], axis = 1, inplace=False), df['class'].values
ohe = OneHotEncoder(sparse=False)
oe = OrdinalEncoder()
scalar = StandardScaler()

ohe.fit(X_test[nom_feat].values)
oe.fit(X_test[ord_feat].values)
```

```
scalar.fit(X_test[num_feat].values)
X_test_nom = ohe.transform(X_test[nom_feat].values)
X_test_ord = oe.transform(X_test[ord_feat].values)
X_test_num = scalar.transform(X_test[num_feat].values)
X_test = np.concatenate([X_test_ord, X_test_num, X_test_nom], axis=1)

X_test = X_test.reshape((X_test.shape[0],1,X_test.shape[1]))
y_test = y_test.reshape((y_test.shape[0],1,1))

# Завантаження нейромережевої моделі
model = tf.keras.models.load_model("LSTM_DDoS_Detection.model")

#Класифікація пакетів та вивід результатів
N = int(input("\n"+color.GREEN+"How many packets do you want to check? We have
    "+ color.RED + str(len(df)) + color.GREEN + " packets you can check:
    "+color.END))
count_ddos = 0

count_normal = 0
for i, each in enumerate(X_test):
    if i < N:
        each = each.reshape((each.shape[0],1,each.shape[1]))
        prediction = model.predict([each])
        scale_pred = prediction[0][0][0]
        if scale_pred >= 0.5:
            print(color.RED + "DDoS Packet found." + color.END )
            print(color.PURPLE + "Packet: "  + color.END + str(origin_X_test[i+2])+"\n")
```

```
        count_ddos += 1
    elif scale_pred < 0.5:
        print(color.GREEN + "This packet is a normal Packet." + color.END)
        print(color.PURPLE + "Packet: " + color.END + str(origin_X_test[i+2])+"\n")
        count_normal += 1
else:
    print(color.YELLOW + str(count_normal) + color.GREEN + " Normal Packets
 Found." + color.END+"\n")
    print(color.YELLOW + str(count_ddos) + color.RED + " DDoS Packets Found."
 + color.END+"\n")
    break
```