**Міністерство освіти і науки України**
**«Київський національний університет імені Тараса Шевченка»**

**Факультет інформаційних технологій**
**Кафедра кібербезпеки та захисту інформації**

# ПОЯСНЮВАЛЬНА ЗАПИСКА

## кваліфікаційної роботи магістра

**галузь знань** *12 Інформаційні технології*
(шифр і назва галузі знань)
**спеціальність** *125 Кібербезпека*
(код і назва спеціальності)
**освітній ступень** *магістр*
(назва освітньої програми)
**освітньо-наукова програма** *кібербезпека*

**на тему:** «Моделі прогнозування мережевої безпеки»

**Виконавець:** студент ІІ курсу, групи КБм-21

(підпис) **Кирило ЗОЛОТАРЬОВ** (прізвище ім'я по-батькові)

| | Прізвище, ініціали | Оцінка | Підпис |
|---|---|---|---|
| **Науковий керівник** | Володимир НАКОНЕЧНИЙ | | |
| **Нормоконтроль** | Олена БОГУСЛАВСЬКА | | |

**Київ 2023**

**Міністерство освіти і науки України**
**«Київський національний університет імені Тараса Шевченка»**

**Факультет інформаційних технологій**
**Кафедра кібербезпеки та захисту інформації**

**ЗАТВЕРДЖЕНО:**
В.о. завідувача кафедри кібербезпеки
та захисту інформації
_____ Сергій ТОЛЮПА
«24» жовтня 2022 р.

# ЗАВДАННЯ
## на виконання кваліфікаційної роботи

**спеціальності** *125 Кібербезпека*
(код і назва спеціальності)

**студенту** КБм-21 (група) Золотарьову Кирилу Михайловичу (прізвище ім'я по-батькові)

**Тема дипломної роботи** Моделі прогнозування мережевої безпеки

### 1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 3 від 20.10.2022

### 2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

| | |
|---|---|
| **Об'єкт досліджень** | Процес виявлення вторгнень в мережі |
| **Предмет досліджень** | Системи виявлення та реагування на вторгнення |
| **Мета** | Удосконалення моделі прогнозування мережевої безпеки за рахунок використання концепцій машинного навчання |
| **Вихідні дані для проведення роботи** | Рекомендації для підвищення ефективності системи виявлення вторгнень, унікальна моделі виявлення аномалій трафіку |

## 3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

**Наукова новизна** Збільшення відсотку точності класифікації моделі на основі концепцій машинного навчання.

**Практична цінність** Покращена модель виявлення вторгнень з високим показником точності класифікації як шаблон для побудови СВВ. Рекомендації по впровадженню машинного навчання в процеси виявлення вторгнень.

## 4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

## 5. ЕТАПИ ВИКОНАННЯ РОБОТИ

| Найменування етапів робіт | Строки виконання робіт (початок-кінець) |
|---|---|
| Розробка плану для досягнення мети роботи | 24.10.2022 – 23.01.2023 |
| Аналіз літературних джерел | 24.01.2023 – 14.02.2023 |
| Розробка комбінованої моделі виявлення вторгнень на основі архітектур машинного навчання | 15.02.2023 – 24.04.2023 |
| Оформлення і друк пояснювальної записки | 25.04.2023 – 19.05.2023 |

## 6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

**Економічний ефект** Зниження збитків з причини пропуску шкідливих пакетів.

**Соціальний ефект** Підвищення ефективності систем виявлення вторгнень для організацій різного розміру та типу власності.

## 7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ Володимир НАКОНЕЧНИЙ
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ Кирило ЗОЛОТАРЬОВ
(підпис) (прізвище, ініціали)

Дата видачі завдання: 24.10.2022 р.

Термін подання кваліфікаційної роботи до ЕК 19.05.2023 р.

**УДК. 004.432.16**

# РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Моделі прогнозування мережевої безпеки»: 68 сторінок основного тексту, 32 рисунка, 5 таблиць, та 28 літературних джерел.

Об’єкт дослідження – процес виявлення вторгнень в мережу.

Мета роботи – удосконалення моделі прогнозування мережевої безпеки за рахунок використання концепцій машинного навчання.

Методи дослідження – експериментальний аналіз наявних методів виявлення аномалій в мережі.

У кваліфікаційній магістерській роботі досліджено сучасні загрози інформаційній безпеці, методи їх вирішення за рахунок ідентифікації на основі системних методів, а також на основі концепцій машинного та глибоко навчання. Проведено експериментальний аналіз наявних варіантів систем виявлення вторгнень на основі вищезазначених концепцій та їх оцінку. Запропоновано комбіновану модель для виявлення аномалій, алгоритм її впровадження у процеси виявлення вторгнень, а також наведено рекомендації щодо правильної оцінки ефективності будь-якої з обраних систем та способів їх впровадження.

Наукова новизна: запропоновано модель, що поєднує згорткову, глибоку нейронну мережу з нейромережею довгострокової короткочасної пам’яті для покращення показників ефективності та точності класифікації.

Актуальність теми обумовлена потребою будь-якої організації в надійній системі, що може своєчасно виявляти шкідливі дії, сповіщати про них, та вживати заходи щодо їх блокування. Наявні системні методи виявлення вторгнень не є ефективними, тому кращим варіантом буде використання методів машинного навчання, глибоких моделей.

Ключові слова: система виявлення вторгнень, аномалія, вторгнення, мережева безпека, нейронна мережа, машинне навчання, глибоке навчання, класифікація даних, прогнозування пакетів .

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

| | | |
|---|---|---|
| **СВВ** | – | Система виявлення вторгнень |
| **МН** | – | Машинне навчання |
| **ГН** | – | Глибоке навчання |
| **НМ** | – | Нейронна мережа |
| **ІС** | – | Інформаційна система |
| **LTSM** | – | Нейронна мережа довгострокової короткочасної пам’яті |
| **CNN** | – | Згорткова нейронна мережа |
| **DNN** | – | Глибока нейронна мережа |
| **ПЗ** | – | Програмне забезпечення |
| **NIDS** | | Network based intrusion system |
| **GrIDS** | | Graph based intrusion system |
| **OIDS** | | Operation based intrusion system |
| **HIDS** | | Host based intrusion system |

# ЗМІСТ

# ВСТУП

У зв'язку з розвитком інформаційних технологій в сьогоденних реаліях постає потреба аналізу трафіку всередині мережі для виявлення потенційних загроз. Відомі статистичні методи виявлення вторгнень є ефективними, за умови наявності детальних характеристик та ознак атаки. Проте атаки мають тенденцію змінюватись, тому статистичні методи є неефективним засобом для виявлення нових видів вторгнень в інформаційну систему.

Поняття інформаційної системи слід розуміти, як комбіновану модель, що складається з компонентів різного рівня автономності, які пов'язані один з одним і здійснюють обмін даними. Переважна більшість компонентів такої системи здатна піддаватись сторонньому втручанню. Кількість інцидентів в області інформаційної безпеки постійно збільшується, тому розробка і вдосконалення засобів захисту стає одним з першочергових питань для забезпечення безпеки.

Ефективний комплекс інформаційної безпеки має систему виявлення вторгнень(СВВ) як один з основних елементів. СВВ має дві основні задачі - аналіз джерел інформації та адекватна, швидка реакція на виявлену загрозу. Для підвищення продуктивності таких систем пропонують використовувати методи виявлення вторгнень на основі архітектур машинного навчання.

Основними завданнями даної роботи є:

- Визначення способів ідентифікації загроз інформаційній безпеці на основі концепцій машинного навчання.
- Визначення показників точності та ефективності для різних моделей виявлення вторгнень.
- Наведення рекомендацій для адекватної оцінки впровадженої архітектури виявлення аномалій.
- Створення комбінованої моделі глибокого навчання для впровадження у процеси виявлення аномалій трафіку.

# РОЗДІЛ 1
# КОНЦЕПЦІЇ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ КІБЕРЗАГРОЗ

## 1.1 Підходи до виявлення загроз інформаційній безпеці

Будь-яка інформаційна система повинна володіти необхідним рівнем стійкості до несанкціонованих впливів, спрямованих на порушення однієї з трьох характеристик інформації, що передається, обробляється або зберігається в системі [1]:

- доступності (гарантія отримання необхідної інформації авторизованими користувачами);
- цілісності (гарантія збереження даними правильних значень);
- конфіденційності (гарантія доступу до чутливої інформації лише; користувачам з необхідним рівнем).

Під загрозою інформаційній безпеці треба розуміти можливі події, процеси, явища, що можуть завдати шкоди інтересам об’єктів інформаційних відносин.

Атака на систему – це комплекс дій, який виконує зловмисник для пошуку і використання вразливостей в мережі. Фактично, атака – це реалізація загрози інформаційній безпеці.

Системи, що виконують функцію захисту інформації повинні включати в себе програмні та апаратні засоби, що виконують моніторинг, аналіз та контроль інформації. До таких засобів відносяться: фаєрволи, антивірусні системи, системи виявлення та реагування на вторгнення.

Підходи щодо побудови системи захисту необхідно обирати на основі масштабів інформаційної системи організації. Для невеликих організацій підійде налаштування фаєрволу та антивірусної системи, для великих - необхідно використовувати більш серйозні концепції захисту, такі як, - системи виявлення та/або запобігання вторгнень.

## 1.2 Системи виявлення вторгнень

Система виявлення вторгнень – програмний або апаратний засіб створений для моніторингу та аналізу мережі з метою виявлення вторгнень і зловмисної активності до того, як ці події нанесуть збитки системі та/або пошкодять інформаційні ресурси. Ефективний комплекс інформаційної безпеки має систему виявлення вторгнень як основний елемент, оскільки СВВ розпізнає та оповіщає про атаки ще до того, як зловмисник здійснить вплив на інформаційну систему. СВВ має дві основні задачі: аналіз джерел інформації та адекватна, швидка реакція, заснована на результатах цього аналізу.

Як вже зазначалось, СВВ виявляють початок атаки на мережу, після чого виконують прописані дії, щодо сповіщення про небезпеку. Після сповіщення СВВ або чекає на подальші вказівки від оператора/адміністратора системи, або самостійно застосовує алгоритм реакції на потенційну загрозу, тобто блокує атаку - розрив з'єднання або виконання скрипту, заданого адміністратором .

Для ефективного виконання поставлених завдань СВВ здійснює наступні дії [2]:

- аналіз активності користувачів системи;
- перевірка налаштувань системи та пошук потенційних вразливостей;
- перевірка цілісності критично важливих системних файлів;
- аналіз поточного стану системи, його порівняння зі станами систем, які піддавались атакам;
- загальний аудит операційної системи.

Використання СВВ має на меті досягнення таких цілей [2]:

- Ідентифікація атаки або потенційного вторгнення в ІС.
- Забезпечення контролю якості адміністрування.
- Прогнозування потенційних атак на основі вразливостей системи.
- Виділення корисної інформації про вторгнення, створення шаблонів атак.
- Визначення джерела атаки відносно локальної мережі.

За характером реакції на ідентифікований аномальний трафік, СВВ можна класифікувати на [2]:

- Пасивні системи виявлення - після ідентифікації підозрілого трафіку система сповіщає адміністратора.
- Активні системи запобігання – блокування атаки, шляхом розриву з’єднання або зміною правил фаєрволу, блокування підозрілого трафіку.
- Гібридні системи - виявлення та реакція на вторгнення в автоматичному режимі.

За рівнем виявлення атак СВВ поділяють на такі види [3]:

- NIDS (англ. Network Intrusrnn Detection Systems) - виявляє вторгнення в систему шляхом перевірки всього трафіку мережі за допомогою наданого адміністратором доступу до концентратора/комутатора. NCBB покриває велику частину трафіку в мережі, а також надає можливість централізованого керування.
- GrIDS (англ. Graph-Based Intrusion Detection System)- удосконалена версія NCBB, яка збирає інформацію з кожного сегмента мережі за допомогою встановленого сніфера, після чого поєднує отримані дані та надає адміністратору системи. Для ефективної роботи системи необхідно інтегрувати її в необхідний сегмент мережі, та налаштувати адаптер на дублювання всіх пакетів всередині мережі.
- OIDS (англ. Operational Intrusion Detection Systems) – система, що спеціалізується на внутрішніх атаках. Шляхом порівняння активності користувача у даний момент часу та його порівняння зі станом норми, система видає результати щодо підозрілості дій та можливості вторгнення.
- HIDS (англ. Host-based Intrusion Detection System)- система, що збирає інформацію стосовно якогось конкретного елемента мережі. Завдяки локальному розташуванню аналіз дій конкретного користувача відбувається з дуже високою точністю, що, в свою чергу, дозволяє виділити лише ті процесі, які мають безпосереднє відношення до атаки.

## 1.3 Постановка задачі

Для забезпечення ефективного захисту не достатньо лише превентивних заходів, які виконують сучасні системи виявлення вторгнень, а саме:

- Сегментація мережі.
- Визначення довірених IP-адрес.
- Двохфакторна ідентифікація.
- Налаштування політики доступу.
- Створення бази шаблонів атак.

У випадку масової кібератаки надзвичайно складно визначити де і як саме почалася атака, окрім цього не слід забувати, що потрібна негайна, швидка реакція, якою, нажаль, існуючі системи виявлення вторгнень(СВВ) не володіють. Основна проблема існуючих СВВ – зв’язок з експертною системою та очікування інструкцій від адміністратора, а також відсутність шаблонів в базі даних для визначення нових типів кібератак.

З метою підвищення ефективності захисту можна автоматизувати процес виявлення та реакції на вторгнення, оскільки це зменшить час, який шкідливе програмне забезпечення (ПЗ) буде функціонувати всередині мережі, та допоможе уникнути суттєвих збитків.

Одним з кращих підходів для автоматизації процесу виявлення вторгнень є використання алгоритмів машинного та глибокого навчання. Моделі на основі цих алгоритмів, окрім автоматичного виявлення та реакції на інцидент, здатні навчатись, самостійно виділяти нові шаблони атак та прогнозувати стан захищеності мережі.

Метою даної роботи є розробка моделі виявлення аномалій мережевого трафіку на основі методів машинного навчання, алгоритм її впровадження в процеси СВВ та надання рекомендацій щодо ефективної оцінки алгоритму виявлення аномалій трафіку.

## 1.4 Концепція машинного навчання

У міру розвитку машинного навчання та завдяки численним прикладам його практичного використання науковці, що працюють у галузі комп'ютерної безпеки, почали працювати над інтеграцією концепцій та технологій машинного навчання, а також інтелектуального аналізу даних для розширення і модернізації можливостей СВВ з метою зміни уявлення про процес обробки даних, та звичної логіки виявлення аномалій в ІС.

Машинне навчання – сфера штучного інтелекту, в якій створюються комп'ютерні моделі, що здатні навчатися з певного набору даних з мінімальним втручанням людини.

Алгоритми машинного навчання загалом поділяються на чотири основні категорії [4] :

- контрольоване навчання;
- неконтрольоване навчання;
- напівконтрольоване навчання;
- навчання з підкріпленням.

### 1.4.1 Алгоритми машинного навчання

Контрольоване навчання аналізує визначені набори даних, які називають навчальним набором, і шляхом порівняння з наявними шаблонами оцінює прогнозований результат. Інформація із відомими даними на вході, називається позначеним набором. Надане в алгоритм значення аналізується та порівнюється з наявним у системі шаблоном для прогнозування відповіді. На першому кроці інформація подається в алгоритм машинного навчання, після чого, за допомогою навчального набору, нейронна мережа запам'ятовує дані, аналізує їх, та самостійно налаштовується. На наступному кроці модель використовує вже вивчені дані на новому вхідному наборі для прогнозування наступного результату.

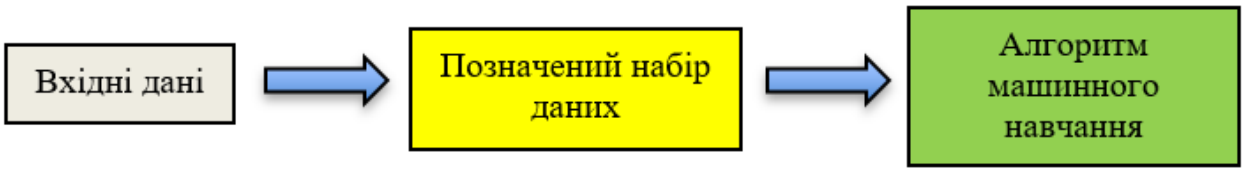


Рисунок 1.1 – Алгоритм роботи контрольованого навчання

У моделі неконтрольоване навчання (UL) немає позначених, відповідно до вхідного набору, даних. Алгоритм самостійно аналізує та навчається на основі прихованих закономірностей та параметрів вхідних даних. Тобто основна відмінність полягає у відсутності позначеного набору даних для процесу навчання, тому можна сказати, шо такий алгоритм працює цілком незалежно і самостійно. Цей алгоритм часто використовують для виявлення шаблонів у масивах даних, які людині складно виділити шляхом створення логічних моделей. В процесі виявлення патернів шаблонів, закономірностей у інформації, модель самостійно пристосовується та корегує власні параметри, після чого формує кластери подібних вхідних даних.

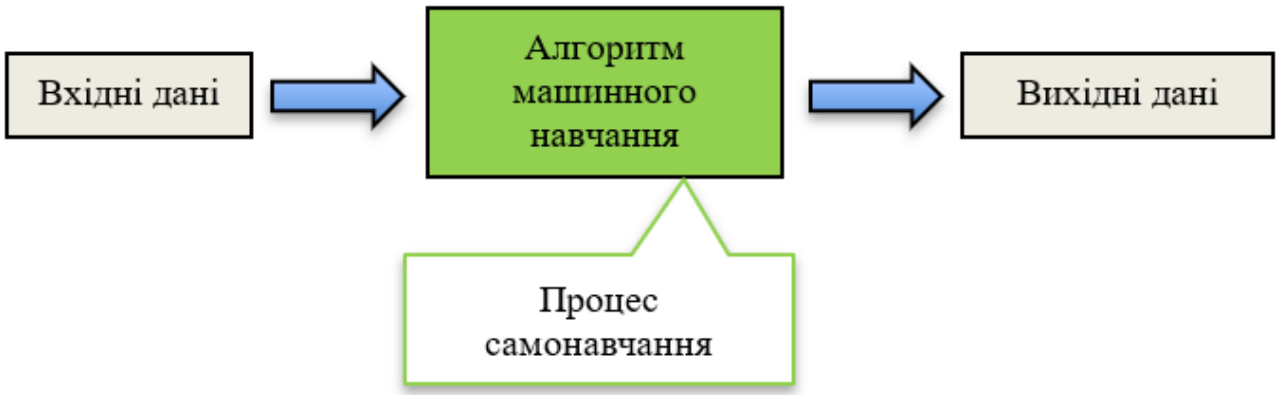


Рисунок 1.2 – Алгоритм роботи неконтрольованого навчання

Напівконтрольоване навчання (SSL) – алгоритм, що поєднує позначені і непозначені вхідні дані для створення класифікатора. Дані для класифікатора

позначаються, аналогічно тому, як це робиться в алгоритмі контрольованого навчання. Після чого подані непозначені на вхід класифікатора масиви даних розподіляються згідно вивчених патернів, а позначені перед входом дані набувають статус істинності, або хибності.

Навчання з підкріпленням (RL) – алгоритм при якому навчання відбувається за допомогою зворотнього зв'язку. Перед кожним, окрім першого, кроком алгоритм отримує відповідь від вчителя (системи управління підкріпленнями), для того, щоб на основі цього зворотнього зв'язку передбачити найкращі дії для наступного кроку.

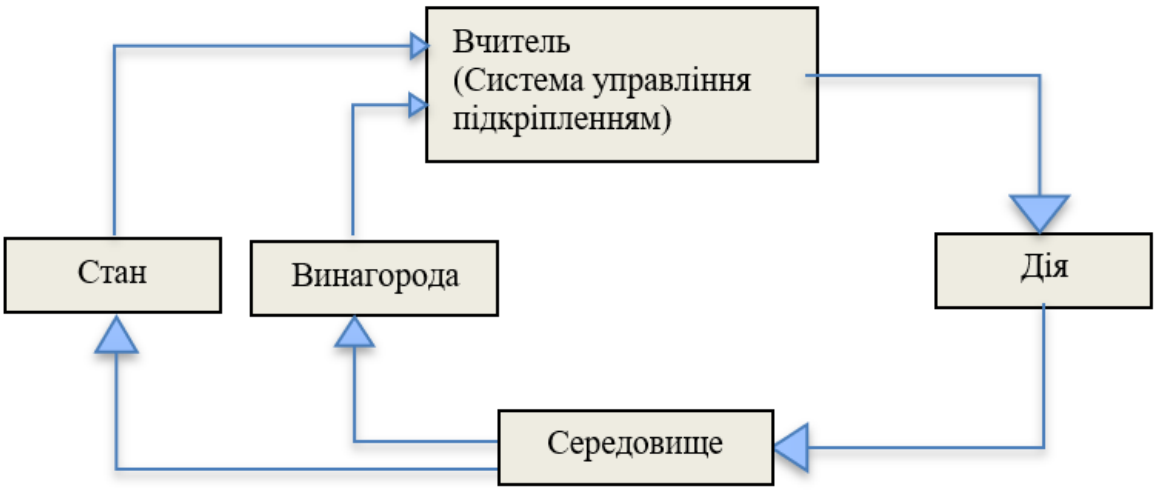


Рисунок 1.3 – Алгоритм роботи навчання з підкріпленням

Такий алгоритм в деякій літературі називають алгоритмом спроб і помилок для досягнення мети. Загалом цей алгоритм можна описати як ітеративний процес, якість і точність якого напряму залежить від кількості зворотного зв'язку або досвіду.

### 1.5 Нейронна мережа

Нейронна мережа — це модель машинного навчання, що функціонує аналогічно нейронам у мозку людини. Нейрон у сенсі машинного навчання — це обчислювальна одиниця, яка має скалярні входи та виходи та ваговий параметр. [5] Нейрон множить вхідну одиницю даних на її встановлений ваговий параметр, після

підсумовує їх та застосовує нелінійну функцію до результату для отримання вихідних даних.

В розумінні нейромережі, будь-який параметр - це нейрон, який представляє з себе функцію, що отримує та аналізує вхідні дані, робить висновок, після чого передає їх наступному нейрону (або шару нейронів), який використовують отримані дані як вхідні для власної функції, після чого також аналізує їх та створює додатковий висновок, щодо істинно вхідних даних. Процес передачі вхідних даних між нейронами триває, поки кожен нейрон, або шар нейронів не буде пройдений.

Кількість шарів та кількість нейронів визначається відповідно архітектурою нейромережі. В залежності від того, яка кількість нейронів змінюється під час ітерації, нейронні мережі можна поділити на синхронні (зміна стану одного нейрона) та асинхронні (зміна стану шару нейронів).

### 1.5.1 Структура нейрона

Нейрон складається з синапсиса суматора і нелінійного перетворювача. Синапси відповідають за забезпечення постійного зв’язку між елементами нейронної мережі, а також множать вхідний сигнал на вагу синапса. Функція виходу суматора активується нелінійним перетворювачем.

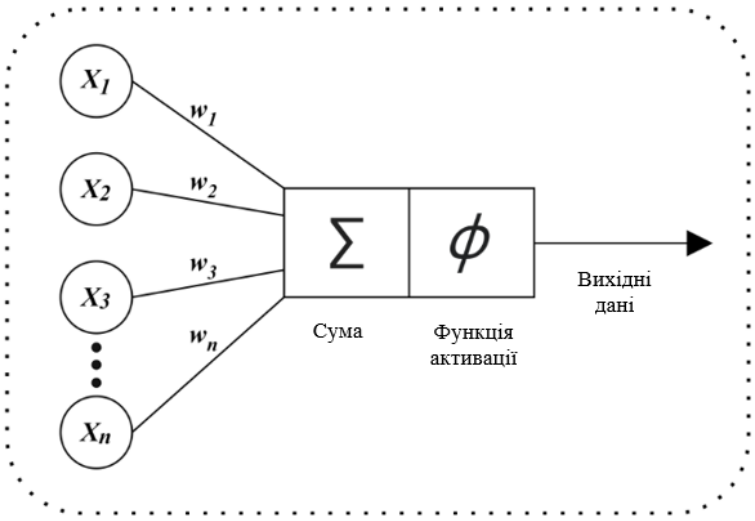


Рисунок 1.4 – Структура штучного нейрона

Суматор сумує подані через синапсичні зв’язки сигнали. Функція виходу суматора активується нелінійним перетворювачем.

### 1.6 Глибоке навчання

Глибоке - клас алгоритмів машинного навчання, в якому застосовується декілька шарів нейронів з метою вивчення та знаходження функцій у поданому наборі вхідних необроблених даних при поступовому русі від нижнього до верхнього рівня мережі. З кожним кроком моделі, дані стають більш точними, що допомагає у подальших завданнях, наприклад – прогнозуванні, класифікації.

Глибокі моделі машинного навчання мають ряд відмінностей від звичайних, а саме [6]:

- Тривалість роботи. Час виконання поставленого завдання зазвичай вище ніж у звичайних моделей, оскільки навчання та випробування глибокої моделі це більш складний та тривалий процес.
- Кількість параметрів. У моделях глибокого навчання визначено два типи параметрів: параметри, які можна визначити, та гіперпараметри. Перший тип обчислюється під час процесу навчання мережі, а гіперпараметри встановлюються адміністратором вручну.
- Подання функції. Для подання на вхід у звичайних моделях необхідно нормалізувати дані до виду вектора функції, моделі глибокого навчання, в свою чергу, здатні приймати необроблені масиви даних.
- Навчальна здатність. Моделі глибокого навчання комплексні і складаються з великої кількості параметрів, тому для процесу навчання їм необхідно в рази більше інформації, ніж для звичайних моделей.

Нейронні мережі навчені за допомогою архітектури глибокого навчання є набагато ефективнішими та допомагають вирішувати досить складні поставлені задачі. Для подальшої роботи над питанням вдосконалення моделі прогнозування мережевої безпеки далі буде детально розглянуто декілька типів нейромереж, шо

використовують модель глибокого навчання, а саме: глибока нейронна мережа, згорткова НМ, НМ з довгостроковою короткочасною пам’яттю.

### 1.6.1 Глибока нейронна мережа

Глибока нейронна мережа – одна з найпростіших моделей глибокого навчання. В цій моделі вхідні дані передаються в одному напрямку, не обмінюючись отриманою інформацією із попереднім шаром нейронів. Тобто фактично дані рухаються з вхідного шару через прихований одразу на шар вихідних даних. На рисунку 1.5 схематично зображено роботу глибокої нейронної мережі, яка складається з трьох шарів.

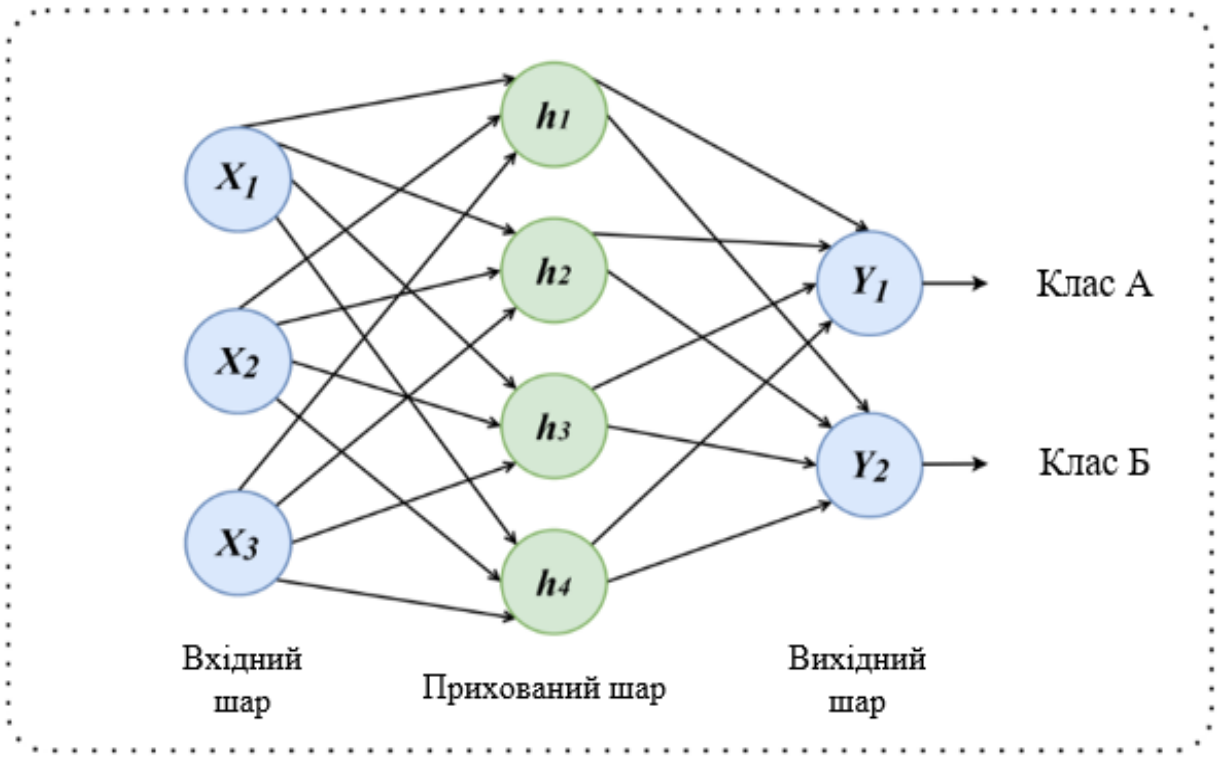


Рисунок 1.5 – Схема роботи глибокої НМ

На прихованому шарі обчислюється сума добутків вхідних значень і вагових коефіцієнтів кожного елемента, що подається на вихідний шар. Вихідне значення отримується шляхом оцінки стану нейрона – віднесення його до одного з оголошених

класів. Наприклад, для всіх значень порогова величина дорівнює нулю, тому якщо нейрон визначено активним, то його значення змінюється на 1, якщо він не спрацьовує, то значення -1.

### 1.6.2 Згорткова нейронна мережа

Згорткова НМ – архітектура глибокого навчання яка направлена на розпізнавання образів даних. Така модель глибокого навчання (ГН) успішно використовується у процесах розпізнавання та класифікації зображень. В основі моделі лежить процес згортки, основна суть якого – кожен елемент множиться на ядро згортки, після чого отриманий результат за кожним елементом підсумовується і записується у відповідний елемент вихідного набору даних.


Рисунок 1.6 – Схема роботи згорткової НМ

За реалізацію процесу згортки відповідає певний клас, в якому саме і зберігається ядро. При кожному проході вхідних даних, кількість нових даних на виході моделі зменшується.

### 1.6.3 Нейронна мережа з довгою короткочасною пам’яттю

НМ з довгостроковою короткочасною пам’яттю – це ще одна із архітектур глибокого навчання, що була створена і ефективно використовується у процесах

визначення довготривалих залежностей. Основна відмінність від будь-якої іншої архітектури – такий вид НМ не перезаписує збережені значення на кожному кроці алгоритму.

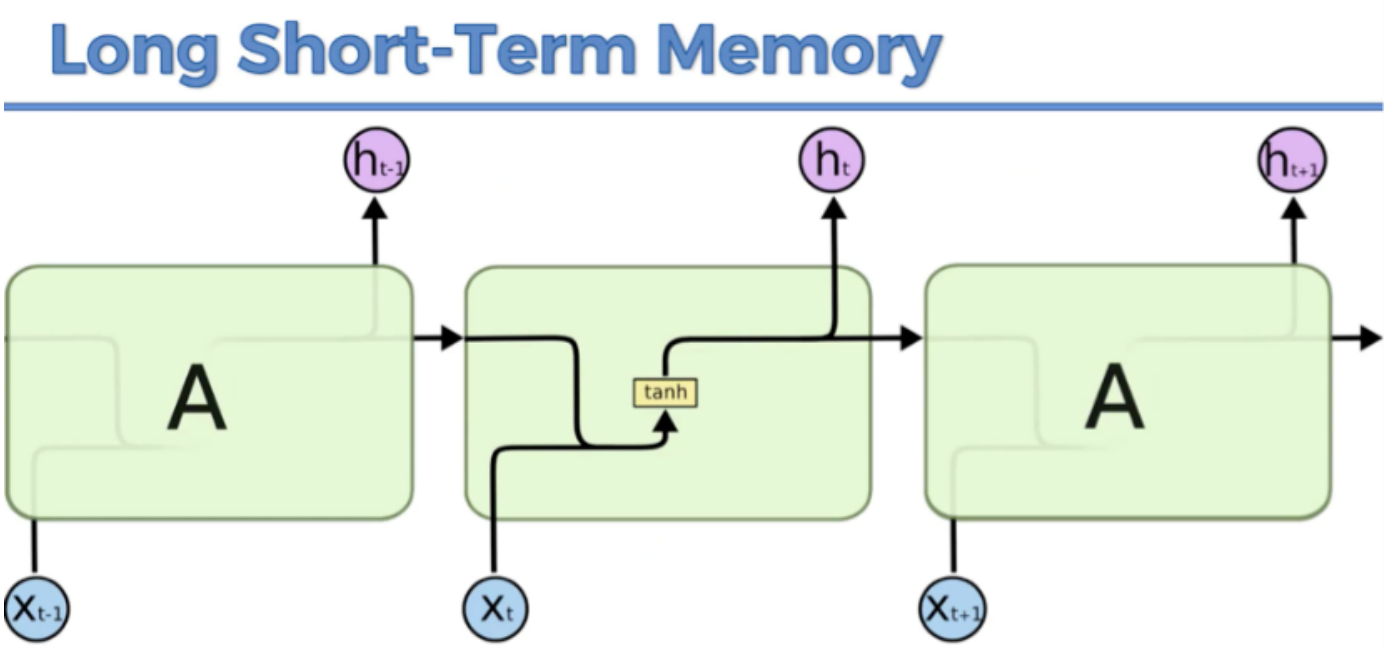


Рисунок 1.7 – Схема роботи НМ з довгою короткочасною пам’яттю

Перевагою моделі над є збільшення показника контролю над вихідним результатом, це означає, що відповідно до цього алгоритм зможе уникнути великої кількості помилок. Окрім цього така модель дозволяє виявляти довготривалі віддалені залежності певних наборів даних. НМ такого виду вдало підходить для завдання навчання з класифікацією, та/або прогнозування.

## Висновки за розділом 1

В першому розділі даної магістерської кваліфікаційної роботи було розглянуто загальні методи виявлення кіберзагроз, проаналізовано методи виявлення вторнень в мережу, визначено особливості побудови системи захисту відповідно до розміру організації. Також було досліджено процес ідентифікації загроз в системах виявлення

вторгнень, визначено основні недоліки у СВВ, згідно чого було поставлено завдання на виконання у подальших розділах даної роботи.

Існуючі статистичні методи виявлення вторгнень не дають в повній мірі розуміння, які конкретні фактори впливають на стан атаки, не можуть прогнозувати її подальші можливі дії. Для вирішення завдань прогнозування мережевої безпеки, а також визначення головних чинників атаки, було запропоновано використання методів машинного навчання для задач прогнозування та класифікації трафіку.

Використання концепцій машинного навчання у процесах СВВ дає змогу автоматично виявляти шаблони у масивах даних для їх подальшого використання в процесі виявлення аномалій трафіку або для прийняття рішень щодо реакції на виявлені загрози.

Впровадження архітектур машинного навчання, а особливо – нейромереж, у процеси виявлення вторгнень є потенційно досить ефективним методом, що має забезпечити ряд переваг при виявленні аномалій трафіку, прогнозуванні безпеки мережі.

# РОЗДІЛ 2
# АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ МЕРЕЖЕВОЇ БЕЗПЕКИ ТА ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

## 2.1 Аналіз ефективності сигнатурних систем виявлення вторгнень

Як вже зазначалось, існуючі СВВ не забезпечують належний рівень захисту, усі наявні методи мають ряд недоліків. З метою захисту мережі від неідентифікованих атак пропонують використовувати різні евристичні методи , сигнатури для системних СВВ, але такі методи мають велику ймовірність пропуску вторгнень, а також велику кількість хибних спрацювань, тому це ставить під сумнів їх ефективність.

Загальна кількість трафіку в мережі надзвичайно велика, крім того трафік постійно змінюється, тому виявити в ньому потрібну невелику частину, що містить аномалію, - дуже складно. Основна мета виявлення аномалії полягає в тому, щоб виокремити відносно малу частину, що потенційно може завдати шкоди, з надзвичайно великого загального обсягу трафіку в мережі. Згідно цього, можна зробити висновок, шо швидке виявлення аномалій є дуже важливим фактором у питання забезпечення безпеки мережі.

Більша частина наявних СВВ побудована на моделі зіставлення сигнатури для пошуку вторгнення. Іншими словами, коли сигнатура знайдена в трафіку співпадає з наявною сигнатурою відомого вторгнення, то система ідентифікує таку сигнатуру як загрозу [7]. Проте сигнатурний метод не є достатньо ефективним та має недоліки у виявленні атак "нульового дня", з тієї причини. Це пояснюється тим, що в базі сигнатур не існує сигнатури, що визначала б поточну атаку. Також сигнатурна модель не може ефективно охопити велику кількість пакетів та точно ідентифікувати атаку, що не є плюсом, оскільки шкідливе ПЗ сьогодні стає дедалі комплексним та складним.

## 2.2 Системи виявлення вторгнень на основі визначення аномалій трафіку

Як вже зазначалося у попередньому пункті данної роботи, практична неможливість визначення атак нульового дня зробила системні методи СВВ практично не ефективними. Потенційним вирішенням є використання методів профілювання. Тобто профілювання адекватної і аномальної поведінки [8].

Цей підхід надає можливість уникнути обмежень, які було описано у сигнатурному методі. СВВ, які мають за основу використання цього підходу, зазвичай використовують алгоритми машинного навчання у поєднанні зі статичними методами для визначення терміну адекватної і аномальної поведінки. Відхилення від встановленого стану адекватності аналізується алгоритмом МН, на наступному кроці створюється висновок щодо можливості вторгнення в ІС. В основі цього методу лежить твердження, що поведінка зловмисника відрізняється від поведінки стандартного користувача, тому може бути ідентифікована як аномалій, згідно цього – як потенційне вторгнення.

Для використання СВВ на основі методу виявлення аномалій необхідно пройти певні етапи при розробці, а саме етап навчання та етап тестування. На етапі навчання звичайні дані трафіку використовуються для навчання системи адекватній поведінці, після чого на етапі тестування в систему подається новий набір даних, що не приймав участі в навчанні, і оцінюється здатність системи до узагальнення нових наборів даних та виявлення аномалій.

Головною перевагою над сигнатурним методом є , як вже зазначалось, можливість ідентифікувати нові види атак, тому що часто дії зловмисників не підходять під певний шаблон або сигнатуру. СВВ сповіщає про потенційну небезпеку, якщо у разі аналізу трафіку було визначено, що поведінка дослідженого користувача або сегменту мережі відрізняється від стану звичайної поведінки.

Іншою, не менш важливою перевагою, є те, що зловмиснику достатньо складно зрозуміти що є адекватною поведінкою користувача системи, тому йому буде набагато складніше реалізувати шкідливі дії не викликавши підозр у системи.

## 2.3 Типи аномалій мережевого трафіку

Аномалія трафіку – певна кількість даних у загальному трафіку, що не підпадає ні під один із чітко визначених класів, сигнатур адекватної поведінки. Важливим фактором у процесі виявлення аномалії є фактор походження аномалії.

Загалом, в літературі мережеві аномалії класифікують на [9]:

- Точкова аномалія (рис 2.1) - конкретний екземпляр із вхідного набору даних не співпадає із станом норми.
- Контекстна аномалія (рис 2.2) – конкретний екземпляр із вхідного набору даних веде себе аномально в певному контексті
- Колективна аномалія (рис 2.3) – група (набір) конкретних екземплярів із вхідного набору даних поводиться аномально відносно до загального набору

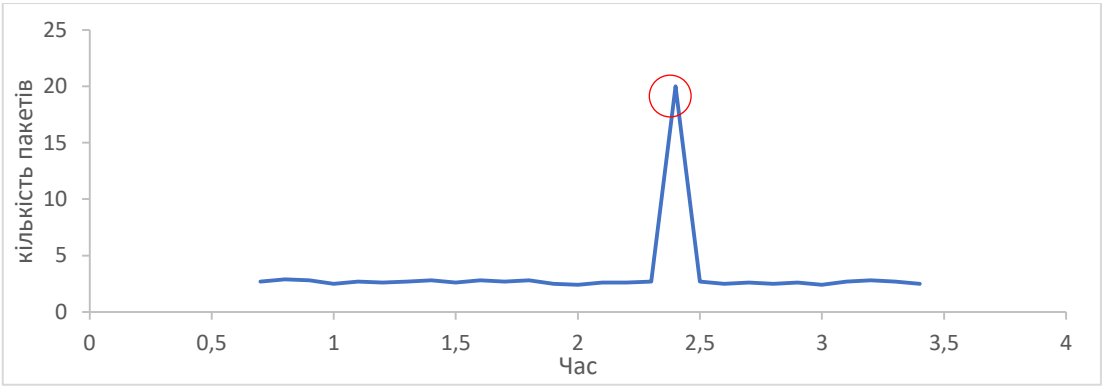


Рисунок 2.1 – Точкова аномалія

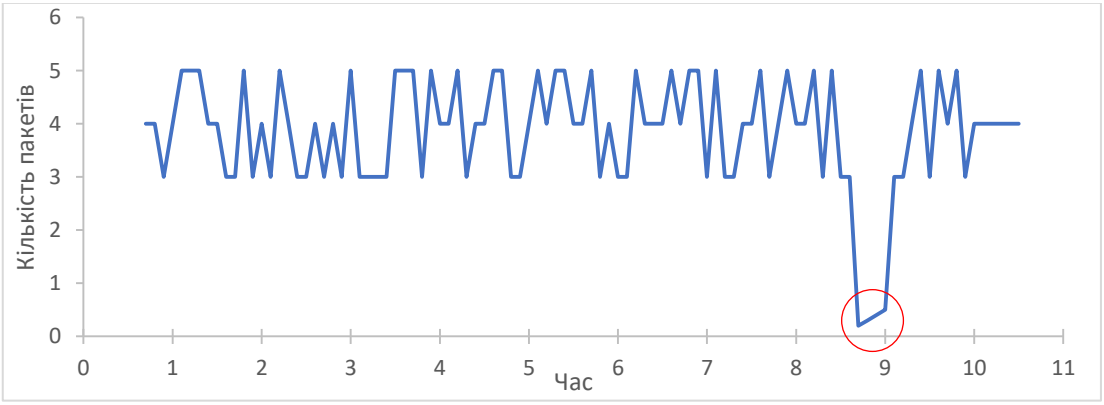


Рисунок 2.2 – Контекстна аномалія

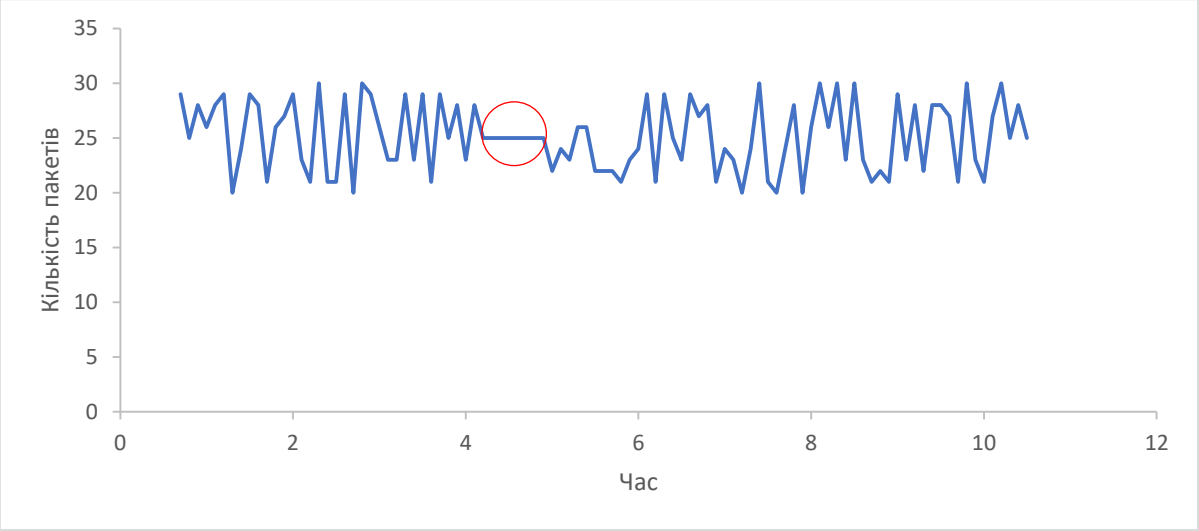


Рисунок 2.3 – Колективна аномалія

На основі розглянутих типів аномалій на рисунку 2.4 наведено схему, де продемонстровано зв’язок між аномаліями і видами мережевих атак.

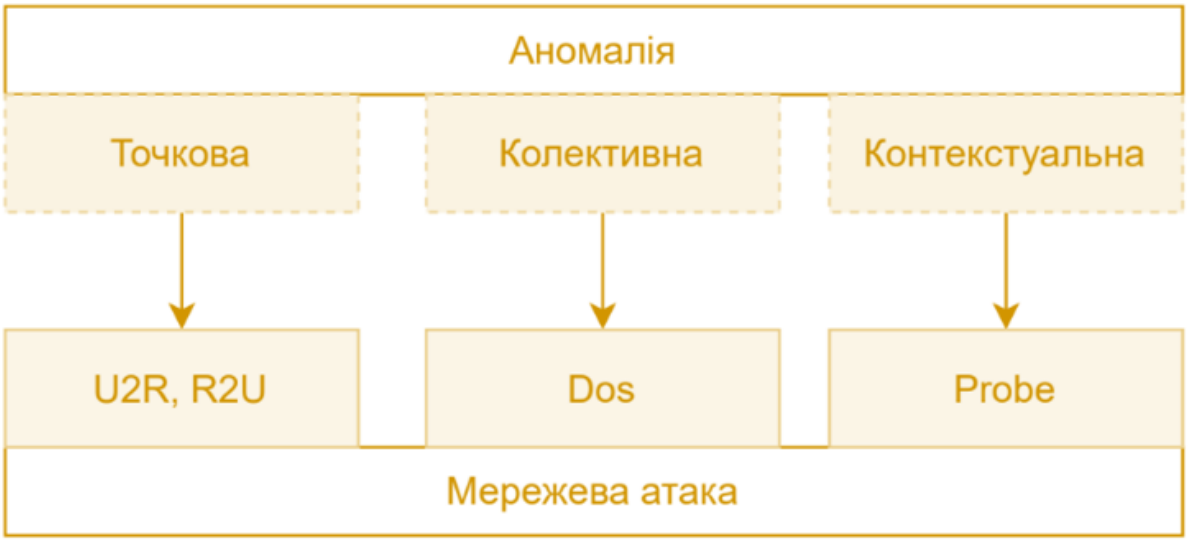


Рисунок 2.4 – Зв’язок між аномалією і видом атаки

Характеристикам точкової аномалії відповідають атаки типу U2R, R2U. DOS можна віднести до колективної аномалій, а атаки типу Probe є контекстною аномалією.

## 2.4 Способи підвищення точності систем виявлення вторгнень за рахунок використання нейронних мереж

У попередніх пунктах даної роботи зазначалось, що підвищення рівня ефективності СВВ можна досягти шляхом застосування алгоритмів МН. Якщо розглядати більш детально, то найкращою архітектурою для застосування у СВВ є нейронні мережі. Інтеграція нейронних мереж у процеси СВВ можлива двома способами [10].

Одним із способів є додавання нейронної мережі в існуючі експертні системи. В такому способі нейромережі використовуються як компонент, що заміняє існуючі в системі моделі статистичного аналізу. Тобто, НМ використовується для фільтрації вхідних даних, після чого на основі аналізу відправляти аномальні екземпляри в експертну систему.

Такий спосіб використання зменшує кількість хибних спрацювань основної системи за рахунок додаткової комплексної фільтрації даних штучним інтелектом. При класифікації даних НМ встановлює імовірність того, що екземпляр є показником атаки, після чого направляє його в експертну систему для додаткового аналізу і подальшої реакції на інцидент. Ефективність експертної системи підвищується за рахунок того, що до неї надходять лише дані про події, які заздалегідь розглядаються як підозрілі.  Ця конфігурація підходить для організацій, які використовують експертні системи на основі правил та політик.

Недоліком цього підходу є те, що, при модифікації, навчанні та набутті нових можливостей нейронною мережею, для виявлення нових типів атак експертна система також потребує оновлення даних. Тільки в такому випадку поєднання цих моделей буде видавати точні показники щодо обраного набору даних.

В іншому підході НМ використовується як автономна СВВ. В такій моделі НМ приймає інформацію з мережевого потоку, після чого проводить їх класифікацію, аналіз на наявність вторгнень. Екземпляри або набори екземплярів, що були ідентифіковані як потенційна загроза спрямовуються адміністратора, або НМ самостійно вживає заходів щодо реакції на загрозу.

Такий підхід є набагато швидшим, оскільки використовується один шар аналізу даних – сама неромережа. На додаток до цього мережа автоматично навчається з кожним новим входом даних. Така модель не обмежується здібностями експертної системи і може створити нові додаткові шаблони.

Виявлення аномалій і потенційних вторгнень залежить від суб'єктивних властивостей ІС. Це означає, що модель нейромережі пристосована під певну мережу може видавати хибні значення для іншої, тому вона має бути дещо видозмінена або модифікована. При цьому при моделюванні СВВ з використанням нейромережі необхідно зробити її якомога універсальною, щоб адекватно оцінювати потенціал такої системи і рівень захисту, який вона забезпечує.

### 2.4.1 Переваги та недоліки використання нейронних мереж у системах виявлення вторгнень

Однією з переваг використання НМ у СВВ можна зазначити гнучкість системи до вхідних даних. Тобто система готова приймати дані, які не співпадають із нормою входу, або якщо дані штучно зашумлені або неповні. Також важливо, що джерел, з яких надходить ця інформація на вхід НМ, може бути декілька. Це пояснюється тим, що багато кібератак проводиться з залученням декількох пристроїв або кінцевих точок.

Наступною перевагою можна виділити швидкість. Як вже зазначалось, НМ постійно моніторить трафік в мережі і має можливість своєчасно виявити початок атаки. Своєчасне виявлення загрози не дозволить їй повністю реалізувати свій потенціал та забезпечить належний рівень безпеки.

Вихідні дані НМ подаються як імовірність, що дає можливість адміністратору системи передбачити можливі наступні вторгнення. Також СВВ на основі НМ можуть самостійно прораховувати імовірність нападу на мережу. При ефективному навчанні НМ, здатність прогнозувати можливі шкідливі події буде тільки збільшуватись, а також, можливо, НМ автоматично зможе вираховувати звідки станеться наступна атака. Вихідні дані роботи НМ можуть бути використані для створення та аналізу

потенційно можливих послідовностей атак, що теоретично дозволить НМ зреагувати до того, як атака відбудеться.

Раніше НМ не мали такої популярності та не так широко використовувались у СВВ. Це пов’язано з тим, що процес навчання може бути досить складним та тривалим, а виявлення аномалій нейромережею напряму залежить від якості навчання. Тобто, процес навчання для НМ, як і загалом для алгоритмів МН є критично важливим. Також, до недоліків можна віднести те, що НМ працює самостійно без втручання людини. Тобто під час роботи по виявленню аномалій НМ самостійно приймає рішення, отримує досвід та робить висновки. Після чого такий досвід та навички застосовуються для подальшої роботи над встановленим завданням.

## Висновки за розділом 2

У другому розділі магістерської кваліфікаційної роботи було детально досліджено процес ідентифікації загроз у системних СВВ, визначено основні недоліки таких систем, згідно чого запропоновано методи підвищення ефективності визначення вторгнень. На основі аналізу методу визначення загроз шляхом виявлення аномалій трафіку було визначено шляхи впровадження архітектур машинного та глибокого навчання у моделі виявлення аномалій трафіку.

Методи ідентифікації кібератак на основі НМ інтегрованої в СВВ застосовуються для попередньої класифікації даних, подальшого аналізу та виявлення аномалій. Такі методи засновані на визначені адекватної поведінки мережі за допомогою функції розподілу та отримання пакетів всередині мережі.

Аномальне відхилення визначається тоді, коли ступінь довіри НМ до висновку щодо класифікації конкретного екземпляра даних становить відсоток, який меньше ніж відсоток ступеня довіри. Для реалізації адекватного захисту НМ повинна пройти процес навчання та тестування. Такі процеси є критично важливими для потенційного виявлення атак на невідомих масивах вхідних даних, а також прогнозування ступеня безпеки мережі.

Розглянувши недоліки та переваги НМ у СВВ можна зробити висновок, що їх використання значно покращує процес виявлення аномалій, також здатність до навчання є беззаперечним пріоритетом, оскільки після навчання НМ здатна самостійно ідентифікувати вразливості, нові види атак, прогнозувати можливі атаки. До недоліків маємо віднести складний математичний апарат, та складність процесу навчання.

Проаналізувавши надану в другому розділі цієї роботи інформацію, можна зробити висновок, що для покращення моделі мережевої безпеки, кращим рішенням буде використання алгоритмів МН, тому, для створення ефективного алгоритму прогнозування мережевої безпеки, у наступному розділі роботи будуть експериментально проаналізовані можливі способи використання концепцій машинного навчання у завданнях виявлення аномалій.

# РОЗДІЛ 3
# УДОСКОНАЛЕННЯ МОДЕЛІ ПРОГНОЗУВАННЯ МЕРЕЖЕВОЇ БЕЗПЕКИ.

В попередньому розділі поданої роботи було розглянуто нейронну мережу як автономний засіб для виявлення аномалій, а також визначено, що такий засіб теоретично є кращим за наявні сигнатурні методи виявлення вторгнень в систему.

Згідно цього, для розроблення рекомендацій щодо прогнозування безпеки мережі, в цьому розділі буде проведений експериментальний аналіз та оцінка ефективності процесу ідентифікації аномалій за допомогою різних архітектур машинного навчання. Також, на основі створених рекомендацій буде запропоновано власну модель ідентифікації аномалій трафіку.

## 3.1 Алгоритм ідентифікації аномалій мережевого трафіку

Для побудови ефективної моделі виявлення вторгнень необхідно визначити згідно якого алгоритму буде проходити процес ідентифікації аномалій та прогнозування можливих вторгнень. Згідно використаної літератури та існуючих підходів, можна визначити такі етапи алгоритму ідентифікації аномалій:

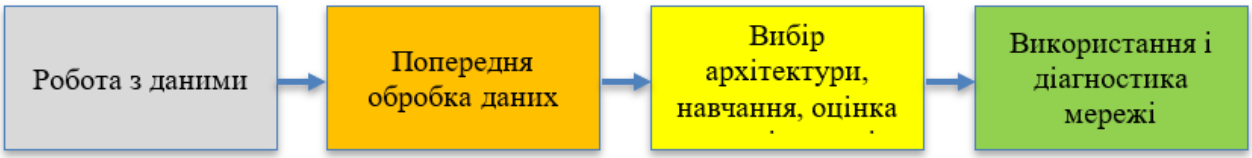


Рисунок 3.1 – Алгоритм ідентифікації аномалій трафіку

1) Робота з даними.

- Створити базу прикладів, що мають відношення до поставленого завдання.
- Поділити дані на дві підмножини, а саме на навчальну і тестову.

2) Попередня обробка вхідних даних.

- Визначити систему класів та ознак необхідних для реалізації поставленого завдання. Модифікувати/Змінити вхідні дані для подачі на вхід НМ (нормалізація, стандартизація, фільтрація даних).
- Створити множину шаблонів (зразків).

3) Вибір архітектури, процес навчання, оцінка ефективності.

- Вибір архітектури мережі.
- Вибір функції активації нейронів та зв’язків.
- Вибір алгоритму навчання.
- Оцінка ефективності на тестовій множині.

4) Використання та діагностика.

- Визначити, чи достатній показник точності класифікації.
- Практичне використання алгоритму, тестування у реальній системі.

### 3.2 Вхідний набір даних та його обробка

Для виявлення аномалій у даній роботі було використано набір даних UNSW-15 dataset, який був створений шляхом запису необроблених мережевих пакетів за допомогою IXIA PerfectStorm. На рисунку 3.2 показано що набір містить реальні дані трафіку, а також в ньому міститься 9 типів кібератак (DOS, Generic, Fuzzers, Shellcode, Backdoors, Worms). Загалом набір даних включає в собі 2 мільйони записів мережевих пакетів, які в свою чергу розділені на декілька CSV файлів (підмножин).

Для подальшого дослідження та опрацювання в цій роботі буде використана підмножина, що містить 257 673 записів. Також буде використано дві множини даних по ~50 000 записів у процесах навчання та тестування, для того щоб адекватно оцінити ефективність обраної моделі.

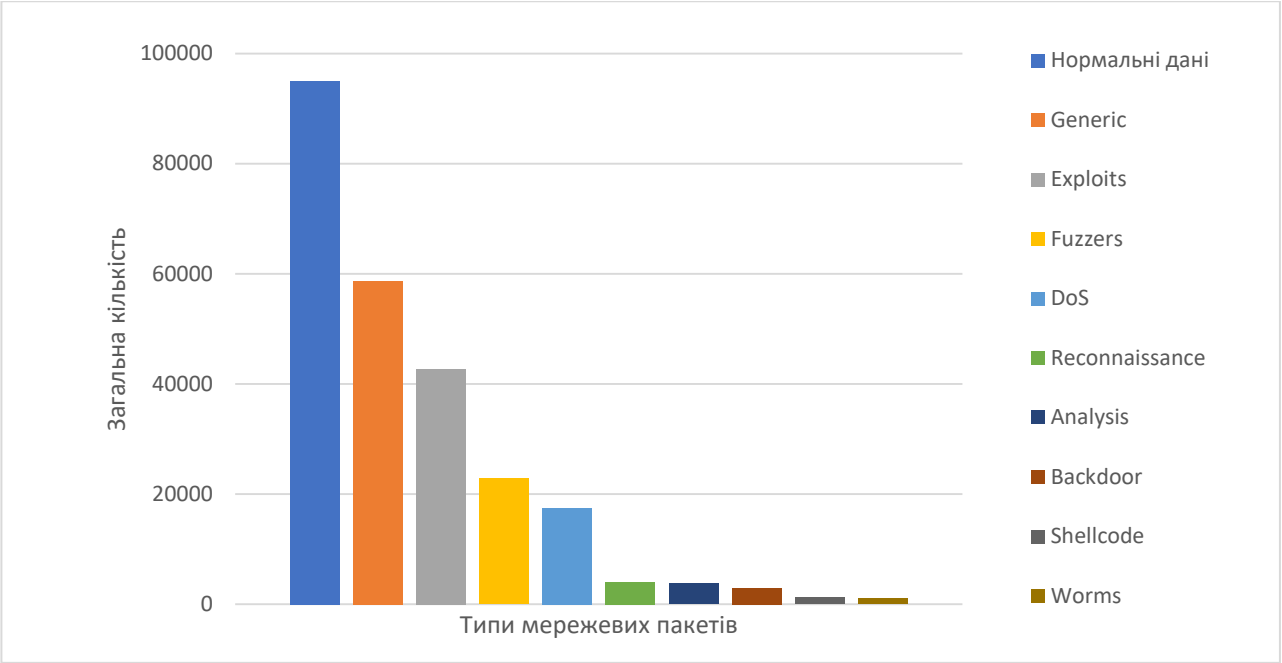


Рисунок 3.2 – Кількість пакетів кожного типу в обраному наборі даних

Файл підмножини, що використовується, складається з реальних данних трафіку а також невизначеної на початку досліджень кількості аномальних пакетів. Кожен файл відображає значення у різні моменти часу, на основі чого вони були відповідно віднесені до аномальних або нормальних значень.

Перший метод попередньої обробки даних, який буде розглянуто, це вибір ознак. Загалом набір даних UNSW-15 містить 49 ознак із відповідними позначеннями класів. Для того, щоб оптимізувати вхідний набір даних необхідно відфільтрувати певну кількість пакетів. Такий підхід забезпечить більшу ефективність та швидкість роботи, оскільки будуть відсіяні дані, які не мають відношення до поставленого завдання.

Для того щоб якісно відфільтрувати, а в подальшому нормалізувати дані, потрібно виконати перевірку важливості ознаки, яка виділить необхідні для досліджень характеристики.

На рисунку 3.3 відображено назву функції та її оціночне значення.

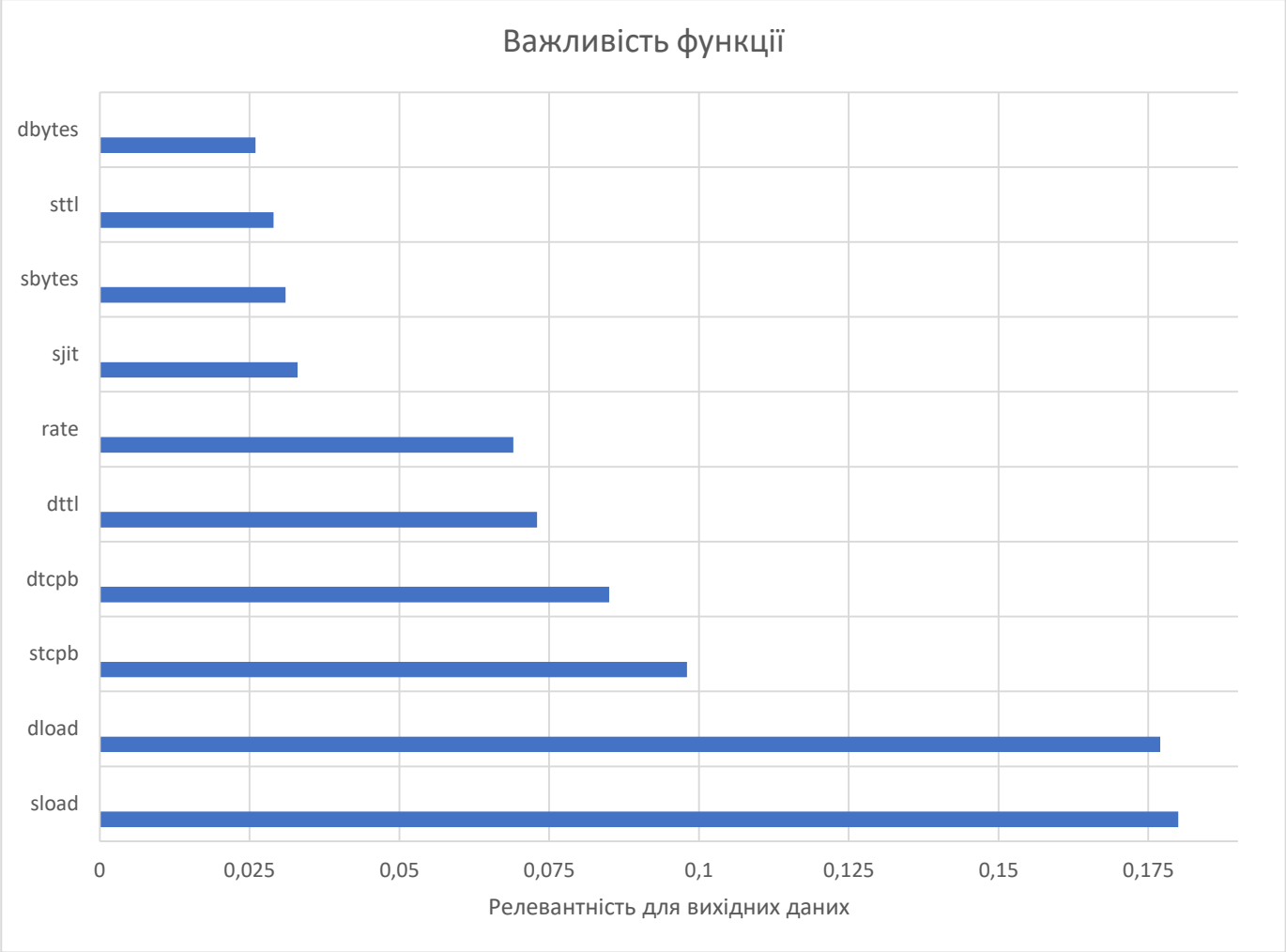


Рисунок 3.3 – Релевантність функції

Виконавши зазначену оцінку ознак було відокремлено з усього набору даних ті, що не мають відношення до поставленого завдання. Також оцінка допоможе у тестуванні моделей, оскільки обчислювальні ресурси будуть адекватно розподіленні через відсутність потреби у обробці нерелевантних даних.

У таблиці 3.1 наведено перелік ключових функцій, які були виділені під час опрацювання вхідного набору даних, їх опис, а також значущість у процентному співвідношенні.

Для того, щоб показати залежність кожної функції від іншої на рисунку 3.4 запропоновано ілюстрацію матриці кореляції.

Таблиця 3.1

Ключові функції у вхідній множині даних

| **Назва функції** | **Тип** | **Опис** |
|---|---|---|
| **sload** | Float | Source bits per second |
| **dload** | Float | Destination bits per second |
| **stcpb** | Integer | Source TCP base sequence number |
| **dtcpb** | Integer | Destination TCP base sequence number |
| **sbytes** | Integer | Source to destination transaction bytes |
| **dbytes** | Integer | Destination to source transaction bytes |
| **sttl** | Integer | Source to destination time to live value |
| **dttl** | Integer | Destination to source time to live |
| **swin** | Integer | Source TCP window advertisement value |
| **dwin** | Integer | Destination TCP window advertisement value |
| **sjit** | Integer | Source jitter |
| **djit** | Float | Destination jitter |
| **spkts** | Integer | Source to destination packet count |
| **dpkts** | Integer | Destination to source packet count |

Матриця відображає зв’язок між двома функціями в наборі даних, що використовується. Основною метою ілюстрації є розуміння закономірностей в даних,

оскільки, якщо функції мають зв’язок, то зміна однієї приведе до зміні відповідної іншої.

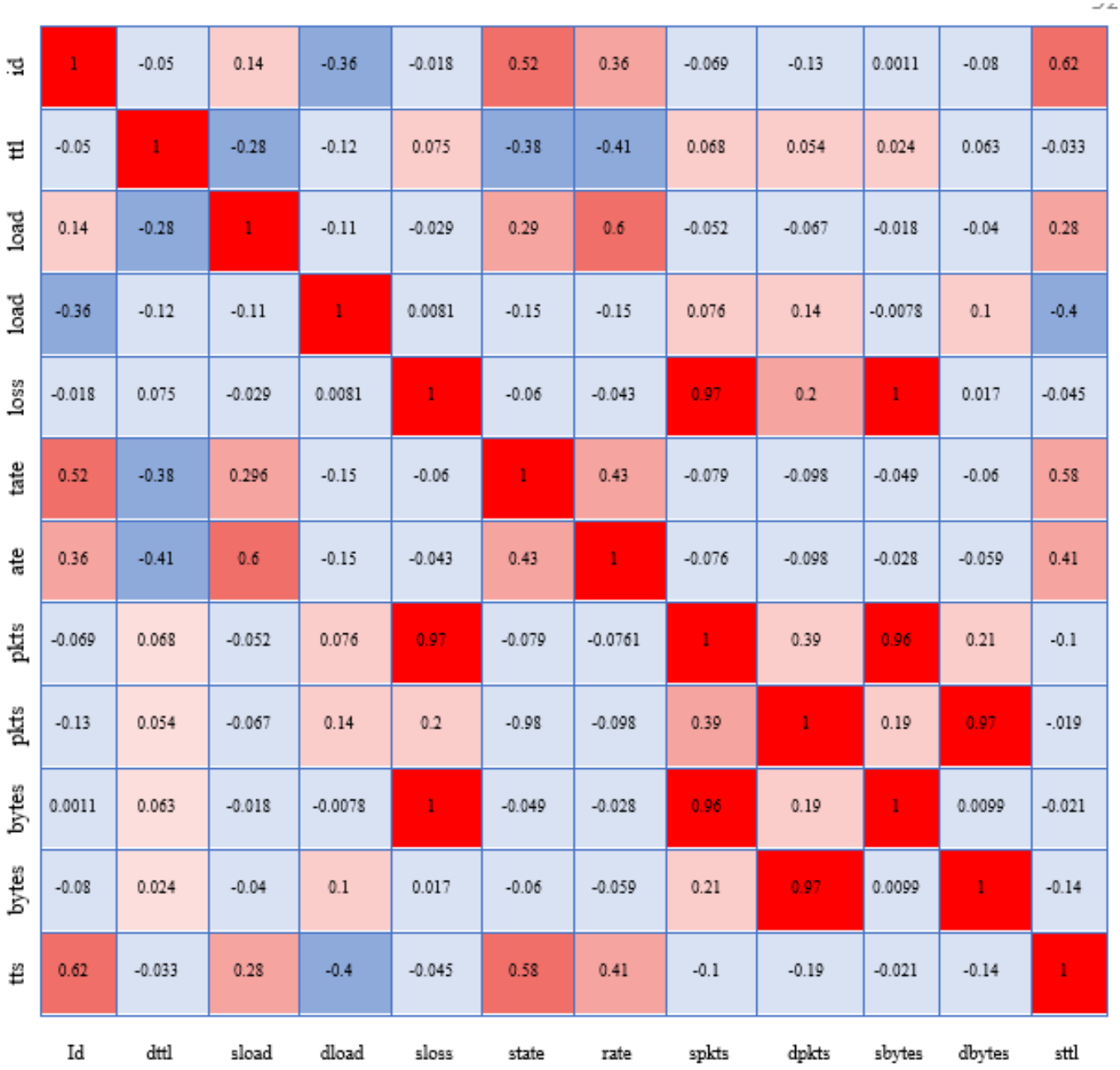


Рисунок 3.4 – Матриця кореляції функцій вхідного набору даних

Цей етап попередньої обробки даних вважається важливим, оскільки можна виявити шаблони залежностей для того, щоб використовувати їх при побудові прогнозних моделей.

### 3.2.1 Нормалізація вхідних даних

Нормалізація вхідних даних – це процес попередньої оброки, при якому всі подані вхідні значення конвертуються або модифікуються для їх подальшого використання у навчанні обраної моделі.

Без нормалізації вхідних даних, великі масиви теоретично матимуть більший вплив на процес навчання моделі, згідно цього – на вихідні дані. Це може призвести до того, що більш важливі об’єкти, що містяться в достатньо невеликих масивах даних, можуть бути проігноровані, а це, в свою чергу, приведе до неефективних висновків на виході роботи обраної моделі. Щоб зробити всі вхідні функції однаково ефективними, їх потрібно нормалізувати, крім того, такі дії допоможуть оптимізувати та пришвидшити алгоритм роботи.

Нормалізація модифікує наявні дані за функціями з їх діапазону в стандартний діапазон, в якому для кожної функції мінімальне значення буде рівно нулю, а максимальне значення – одиниці. Такі дії забезпечать всім функціям рівні права, що позитивно позначиться у процесі створення статистичних висновків. Формулу нормалізації можна подати у вигляді[11]:

$$x = \frac{(x' - x_{min})}{(x_{max} - x_{min})} \tag{3.1}$$

На рисунку 3.5 зображено оригінальні вхідні дані, а на рисунку 3.6 – вже нормалізовані дані. Цей етап обробки даних вважається критично важливим, тому що більшість алгоритмів машинного навчання вимагають специфічних даних для подачі на вхід, тобто це означає, що вхідні дані мають бути конвертовані та нормалізовані.


Рисунок 3.5 – Графік стану оригінальних даних


Рисунок 3.5 – Графік стану нормалізованих даних

Цей етап обробки даних вважається критично важливим, тому що більшість алгоритмів машинного навчання вимагають специфічних даних для подачі на вхід, тобто це означає, що вхідні дані мають бути конвертовані та нормалізовані.

### 3.3 Класифікація даних. Точність класифікації

Точність – це величина оцінки, що застосовується до класифікаційних моделей, в яких порівнюється кількість правильних прогнозів із загальною кількістю прогнозів, що видає обрана модель.

Формула точності класифікації виглядає наступним чином [11] :

$$\text{Точність} = \frac{\text{Число правильних прогнозів}}{\text{Загальне число прогнозів}} \times 100\% \qquad (3.2)$$

Використовуючи цю формулу можна отримати процентне значення, що може бути використане для оцінки ефективності моделі. Слід зазначити, що формула не враховує дисбаланс класів, що може бути присутнім в наборі даних. Тому крім цього слід визначити додаткові показники ефективності для оцінки моделі.

#### 3.3.1 Матриця плутанини

Матриця плутанини — це візуальне представлення продуктивності моделі класифікації. Фактично, матриця плутанини являє собою таблицю, що складається з чотирьох комбінацій прогнозованих і фактичних значень. Результат моделі класифікації подається у 4 можливих поля матриці плутанини категорії [12] :

1. Істинно позитивний результат:  Значення, що були визначені моделлю як позитивні, і вони дійсно виявились позитивними та правильними. На прикладі СВВ – модель визначила пакет даних як шкідливим, після аналізу він дійсно виявився шкідливим. Це означає, що СВВ зробила правильне визначення, тому його можна назвати істинно позитивним.

2. Помилково позитивний результат: Значення, які були визначені як позитивні, але виявились негативними, тобто помилковими. На прикладі СВВ – модель визначила пакет як шкідливий, хоча насправді пакет був звичайним. Високі помилково негативні результати можуть привести до хибних спрацьовувань СВВ, а

також спричинити збої в роботі системи. Низьке помилково позитивне значення є показником точної моделі.

3. Істинно негативний результат: Значення, що були визначені як негативні, тобто помилкові, і після аналізу вони виявились негативними. На прикладі СВВ – модель визначила пакет як негативний і він насправді негативний. Високе істинно негативне значення також є показником ефективної і точної моделі.

4. Помилково негативний результат: значення, що були визначені як негативні, але насправді були позитивними. На прикладі СВВ – модель визначила пакет як позитивний, а після аналізу він виявився негативним. Цей показник є найважливішим у процесі оцінки моделі, оскільки він прямо вказує на те скільки хибних прогнозів зробила модель. На рисунку 3.6 зображено приклад зразка матриці плутанини.

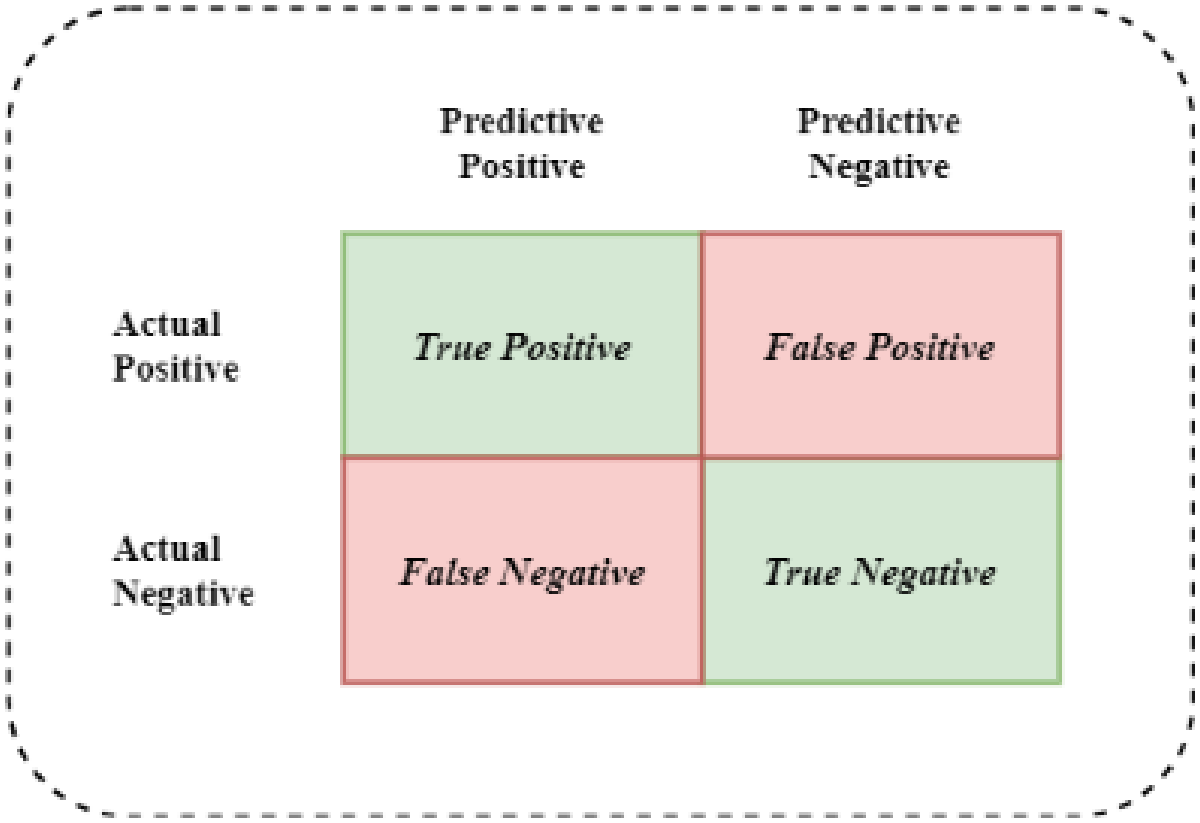


Рисунок 3.6 – Матриця плутанини

. З рисунку можна зрозуміти, що найбільший вплив на результат оцінки мають значення помилково позитивних та помилково негативних результатів.

### 3.3.2 Крива AUC-ROC

Крива AUC-ROC є ще одним застосованим метричним критерієм ефективності для моделі класифікації. Термін AUC скорочено від Area Under the Curve, який вимірює двовимірну площу під ROC, скорочено від Receiver Operating Characteristic Curve за різних порогових значень.

Щоб побудувати ROC, необхідно порівняти параметри, а саме коефіцієнт істинно позитивних результатів і коефіцієнт помилкових позитивних результатів, які можна підсумувати таким чином [13]:

А. Істинний позитивний показник також відомий як чутливість моделі, яка визначає частку значень, які є позитивними та були дійсно правильно визначені як позитивні.

Це можна виразити як:

$$\text{ІПП} = \frac{\text{Істинно позитивний}}{\text{Істинно позитивний} + \text{Помилково негативний}} \tag{3.3}$$

В. Рівень помилково позитивних результатів також відомий як специфічність моделі, яка визначає частку значень, які є від'ємними та були визначені моделлю як від'ємні.

Це можна виразити як:

$$\text{ППР} = \frac{\text{Помилково позитивний}}{\text{Помилково позитивний} + \text{Істинно негативний}} \tag{3.4}$$

Крива ROC відображає частоту істинних позитивних результатів моделі з частотою хибних позитивних результатів за різними пороговими значеннями класифікації, як показано на рисунку 3.7.

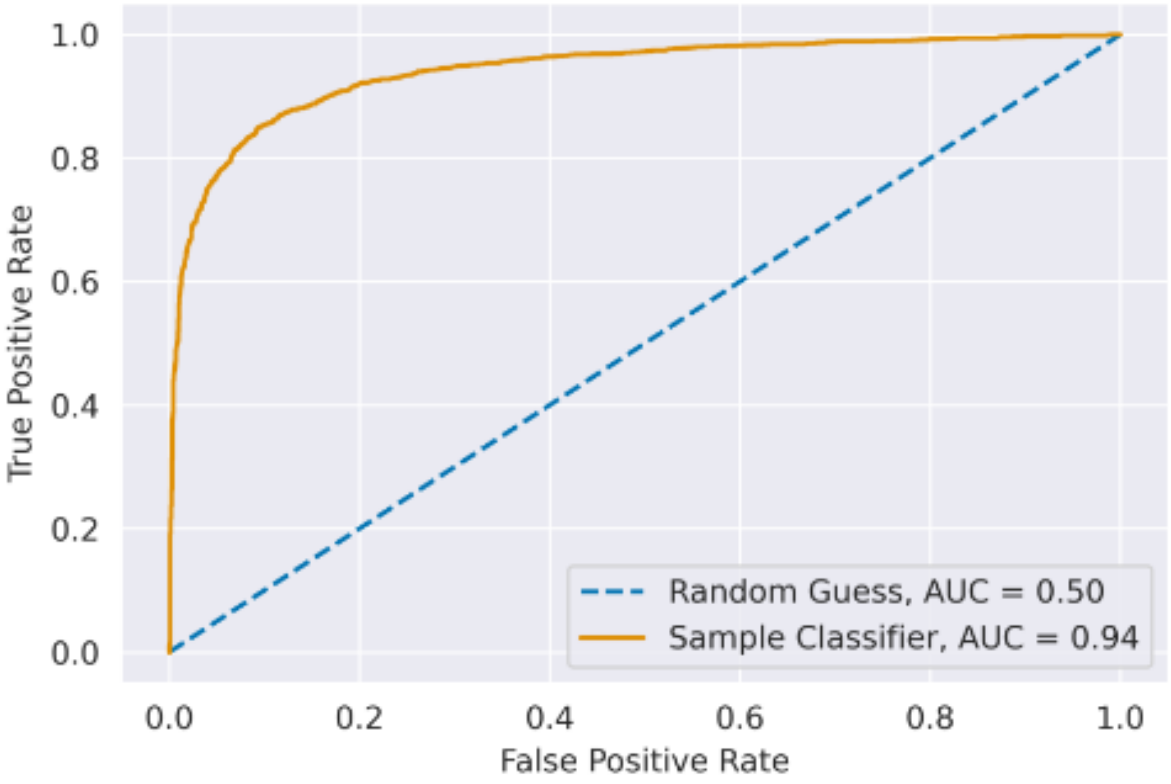


Рисунок 3.7 – Приклад кривої ROC

Значення AUC агрегує продуктивність моделі за всіма можливими пороговими значеннями класифікації

## 3.4 Вибір архітектури СВВ

### 3.4.1 Архітектури машинного навчання. Ефективність, швидкість роботи, точність класифікації

В цьому пункті роботи описано та продемонстровано експериментальні результати щодо застосування алгоритмів машинного навчання у процеси виявлення аномалій.

Як зазначалося раніше, машинне навчання в прикладному сенсі означає, що прогнозується характер даних на основі попереднього аналізу, який був задіяний під час процесу навчання. Для більшої ефективності СВВ перше що потрібно знати – це

походження пакета та чи можна віднести його до виділених класів, які модель може розпізнати.

При тестуванні та оцінці методів машинного навчання у процесах СВВ будемо використовувати методологію контрольованого навчання. Спочатку створимо модель із різноманітними прикладними, яка буде вміщати в собі як звичайні так і шкідливі пакети з основної підмножини. Це допоможе у відокремлені шаблонів, які в подальшому будуть використані для класифікації інших масивів даних. При такому методі навчання будь-які зловмисні пакети будуть ідентифікуватися як аномалія, а в подальшому їх характеристики будуть базовим рівнем для виявлення нових типів аномалій.

Така модель допоможе визначити оптимальний та адекватний стан (поведінку) мережі та пакетів, що транспортуються, тобто відокремити як нормальні характеристики нормальних пакетів, так і характеристики зловмисних пакетів.

Фактично, модель буде шукати будь-які відхилення від встановленого стану норми, який буде визначений під час процесу навчання, з метою відокремлення аномальних даних, пакетів у процесі майбутнього використання. Тобто відфільтрувати всі пакети із точки зори нормального використання та нормального стану мережі.

В процесі виконання роботи було відокремлено такі моделі машинного навчання, а саме:

- Логістична регресія.
- Метод найближчих сусідів.
- Дерево рішень.

На рисунку зображено гістограму для точності класифікації кожної моделі машинного навчання, застосованої для завдання виявлення вторгнень

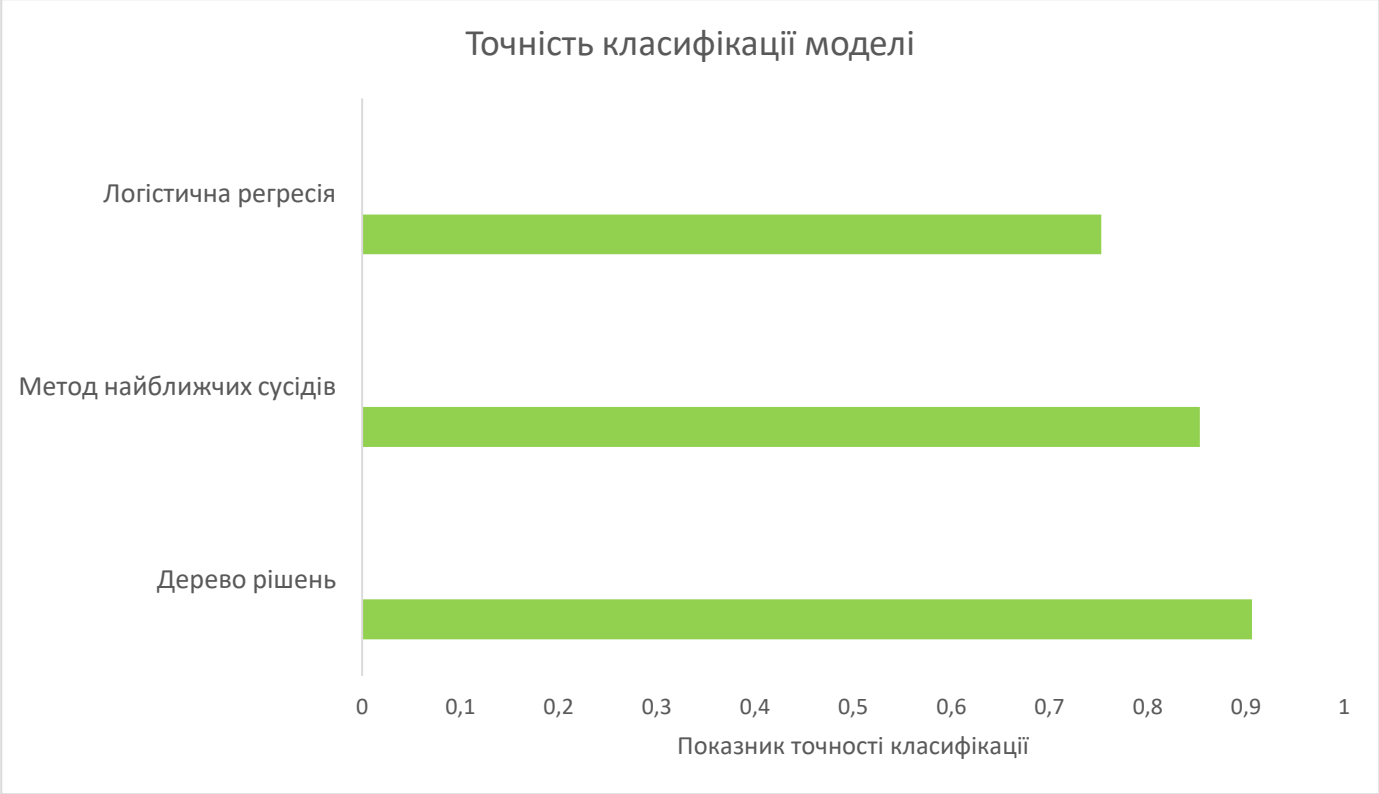


Рисунок 3.8 – Діаграма точності класифікації моделей машинного навчання

Під час проведення експериментів було визначено, що модель дерева рішень демонструє найкращі результати з точки зору оцінки точності - 90,64%. Метод найближчих сусідів продемонстрував точність класифікації 85,32%, а логістична регресія - 75,27%.

Оскільки окрім точності також важливим аспектом є швидкість роботи, також експериментально було досліджено і швидкість роботи моделей. Було досліджено за який період часу модель обробила весь набір даних обраної підмножини. Серед застосованих моделей «Дерево рішень» класифікувало дані за 6,33 секунди, метод найближчих сусідів за 31,6 секунди. Логістична регресія показала найкращий час тестування – 3,01 секунди.

На додаток до цього було використано криву ROC, щоб продемонструвати продуктивність кожної із застосованих моделей машинного навчання при різних порогових значеннях. Дані продемонстровано на рисунках 3.9 – 3.11 нижче:

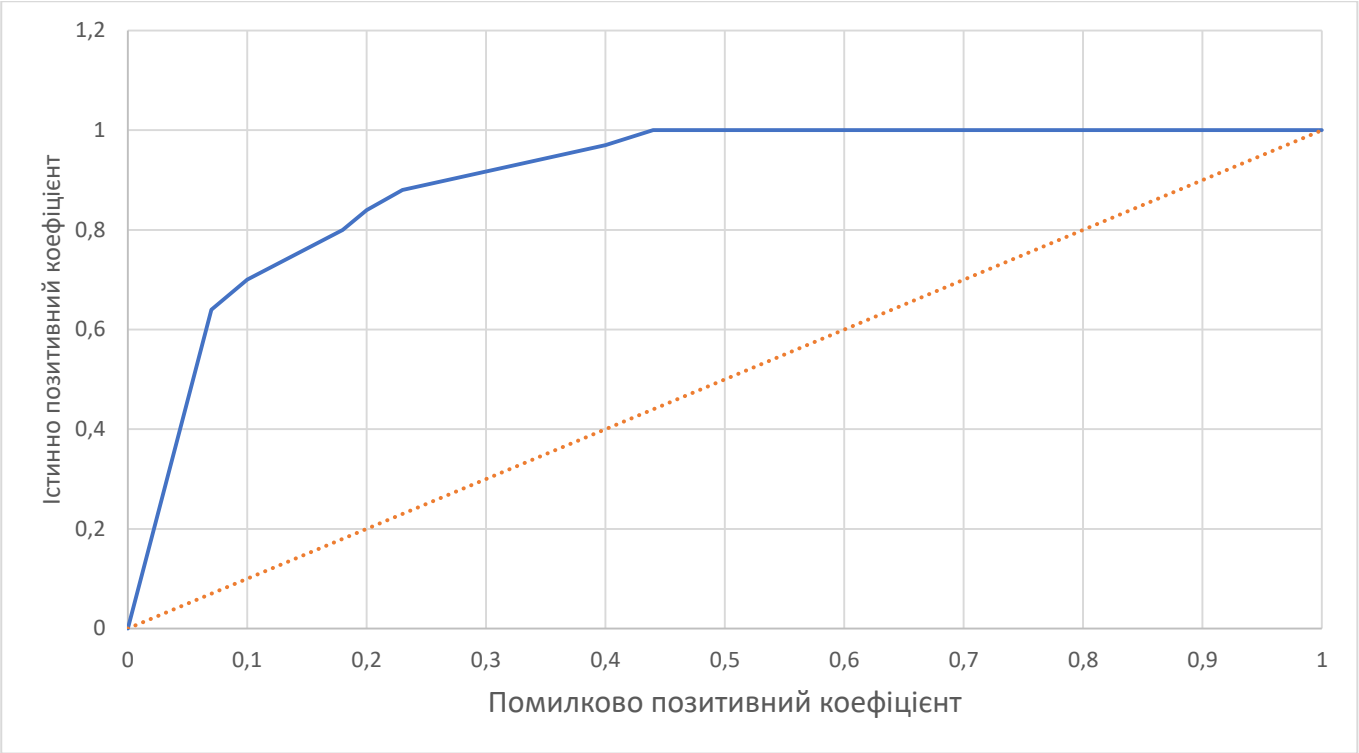


Рисунок 3.9 – Крива ROC для логістичної регресії

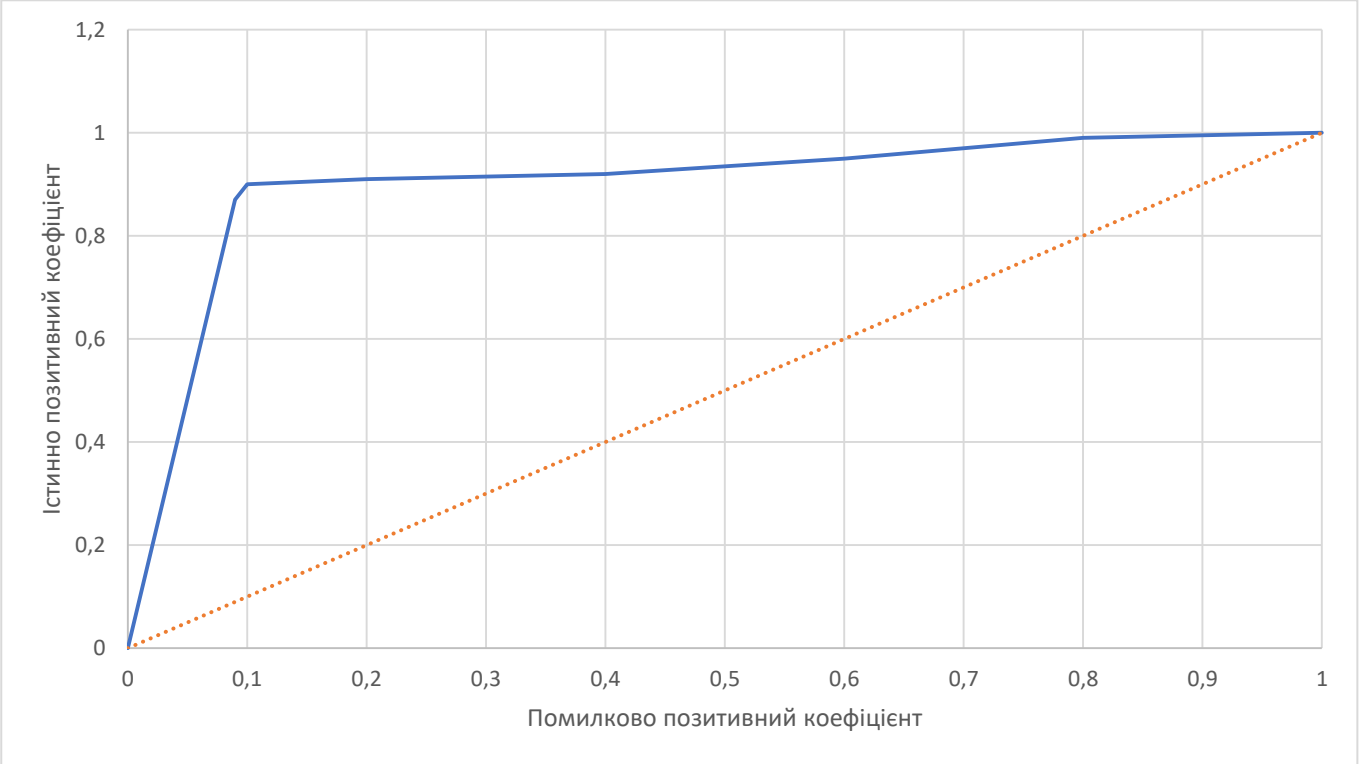


Рисунок 3.10 – Крива ROC для методу найближчих сусідів

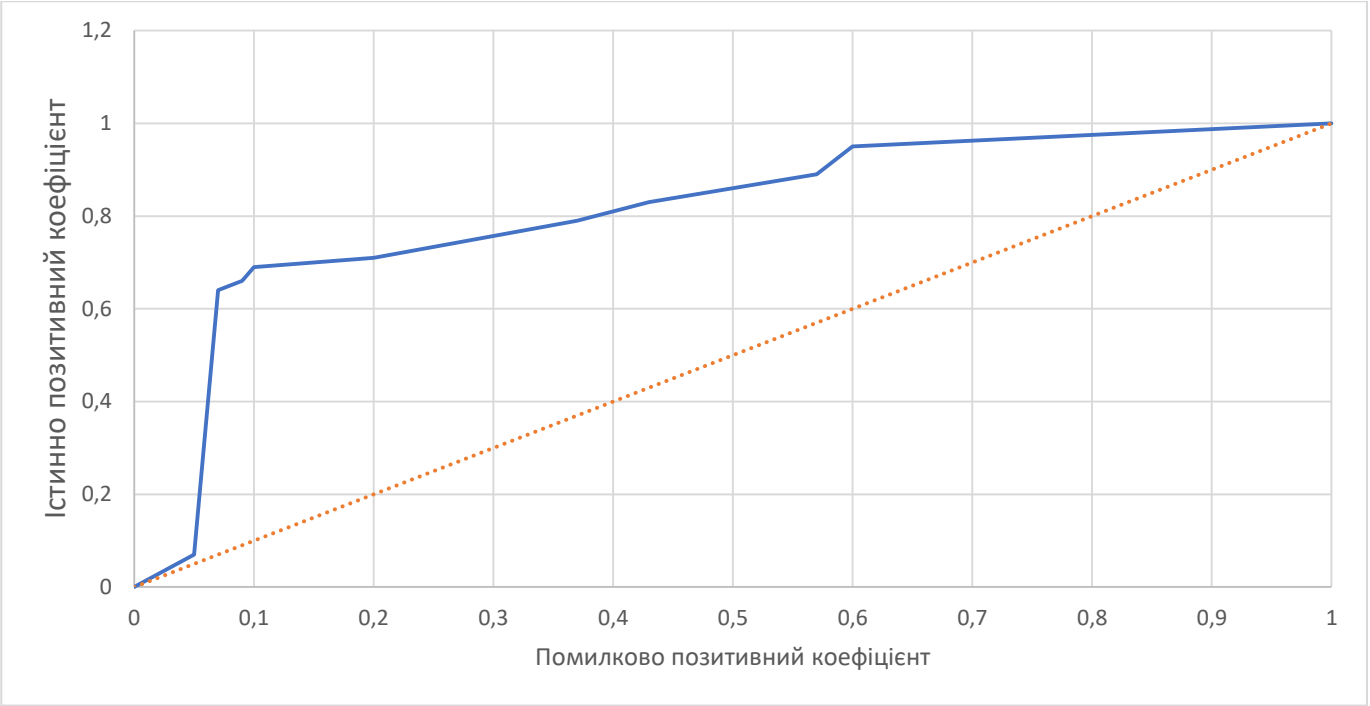


Рисунок 3.11 – Крива ROC для дерева рішень

Таблиця 3.2

Результати досліджень моделей машинного навчання

| МОДЕЛЬ | ТОЧНІСТЬ | ШВИДКІСТЬ | AUC |
|---|---|---|---|
| ЛОГІСТИЧНА РЕГРЕСІЯ | 75,27% | 3,01с | 0,84 |
| МЕТОД НАЙБЛИЖЧИХ СУСІДІВ | 85,32% | 31,6с | 0,93 |
| ДЕРЕВО РІШЕНЬ | 90,64% | 6,33с | 0,91 |

В таблиці 3.2 узагальнено результати експериментів для моделей машинного навчання. На основі проведених експериментів та отриманих результатів можна дійти до висновку, шо дерево рішень є достатньо ефективним компонентом для

використання у процесах СВВ. Також для повної оцінки цієї моделі було проведено його тестування з використанням матриці плутанини. Результати тестування проілюстровано на рисунку 3.12:

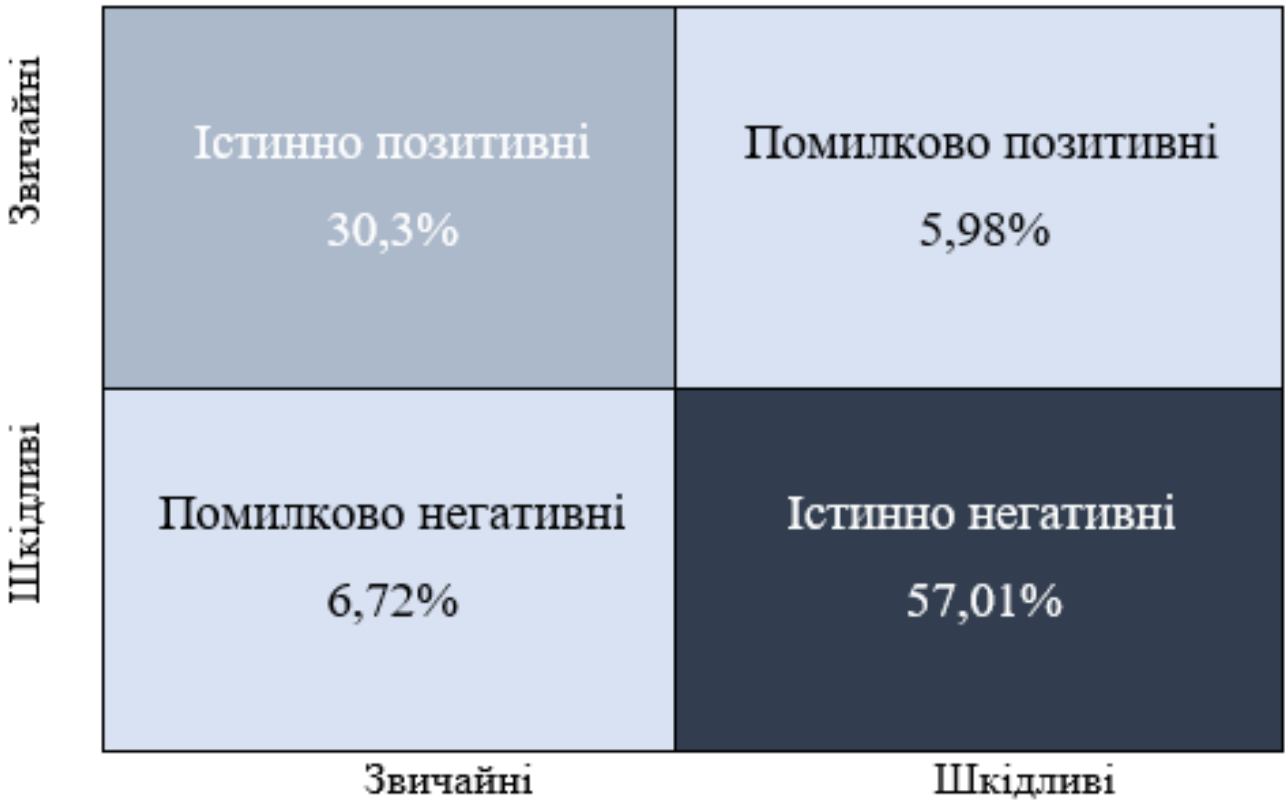


Рисунок 3.12 – Матриця плутанини

Згідно з матрицею плутанини, модель дерева рішень мала 5,98% помилково позитивних результаті і 6,27% помилково негативних результатів. Як зазначалося у попередній пунктах роботи, помилково позитивні результати визначають відсоток звичайних пакетів, які були визначені як шкідливі, а помилково негативні визначають відсоток шкідливих пакетів, які були визначені моделлю як звичайні.

Отже, це означає, що СВВ на основі дерев рішень можуть пропустити 6,72% зловмисних пакетів непоміченими, що в свою чергу може стимулювати більш складні атаки та вторгнення. Також було визначено 5,98% звичайних пакетів як шкідливі, це також можна вважати негативним результатом, оскільки модель витратила час на обробку непотрібної інформації.

Незважаючи на досить короткий термін виконання обробки даних моделлю дерева рішень, а також високу точність класифікації, можна прийти до висновку, що така модель не є ідеальною і не може видати гарні результати.

Для підвищення показника точності класифікації, а можливо і часу роботу, необхідно розглянути, проаналізувати і протестувати сферу глибокого навчання, для того щоб виявити найкращу модель прогнозування мережевої безпеки.

### 3.4.2 Архітектури глибокого навчання. Переваги над методами машинного навчання.

Виходячи з результатів попереднього пункту роботи, для підвищення ефективності роботи СВВ є необхідність додатково дослідити архітектури глибокого навчання для їх використання у процесах СВВ. За допомогою архітектур глибокого навчання (ГН) можна опрацьовувати більш складні аналітичні прогнози, виділяти шаблони більш складного типу, такі можливості пояснюються наявністю декількох рівнів складних нейромережевих зв’язків.

У попередньому розділі роботи було визначено, що на відміну від алгоритмів МН, алгоритми ГН не потребують втручання людини для отримання результату, тобто вони саморегулюються..

Серед моделей глибокого навчання було виділено найефективніші за оцінками користувачів моделі, а саме[14]:

- Глибока нейронна мережа.
- Згорткова нейронна мережа.
- Нейронна мережа з довгою короткочасною пам’яттю.

Для навчання та перевірки вибраних моделей буде використано два попередньо оброблені набори даних. Загалом як і при експериментах з машинним навчанням буде використано підмножину із 206 138 пакетів.

На рисунку 3.13 наведено діаграму точності класифікації для кожної з виділених для тестування архітектур глибокого навчання.

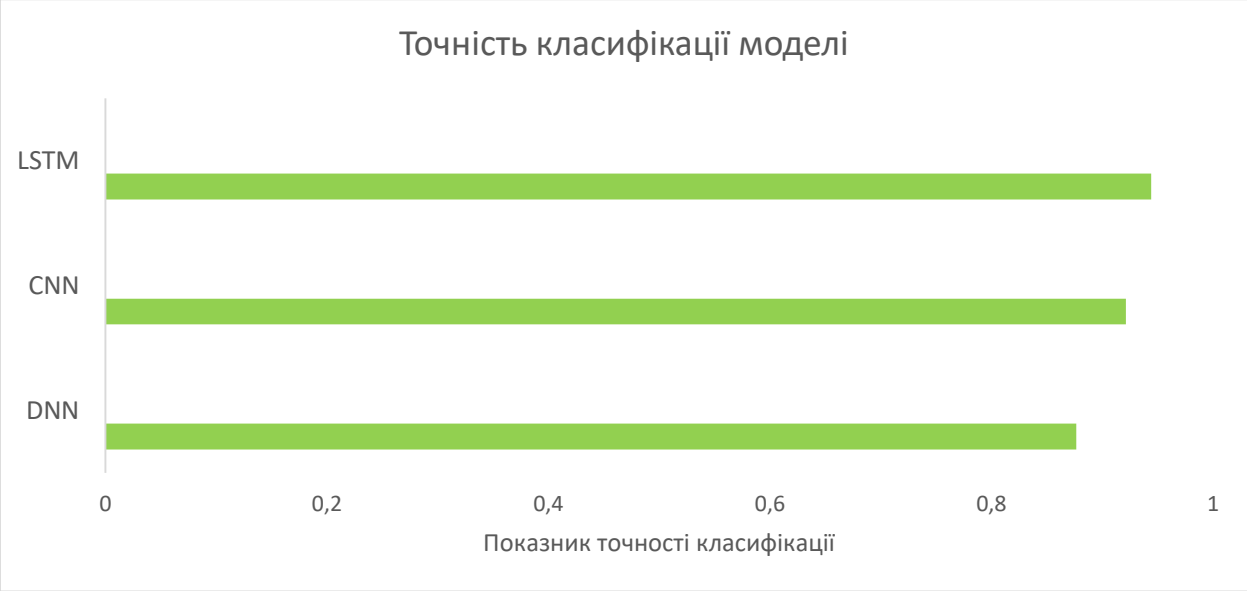


Рисунок 3.13 – Точність класифікації моделей глибокого навчання

Під час дослідження було визначено, що LSTM демонструє точність класифікації 94,42%, CNN - 92,16%, а DNN дає найменший показник точності класифікації - 87,66%.

Далі по аналогії з дослідженнями моделей машинного навчання, було використано криву ROC, щоб продемонструвати отримані у процесі дослідження результати при різних порогових значеннях.

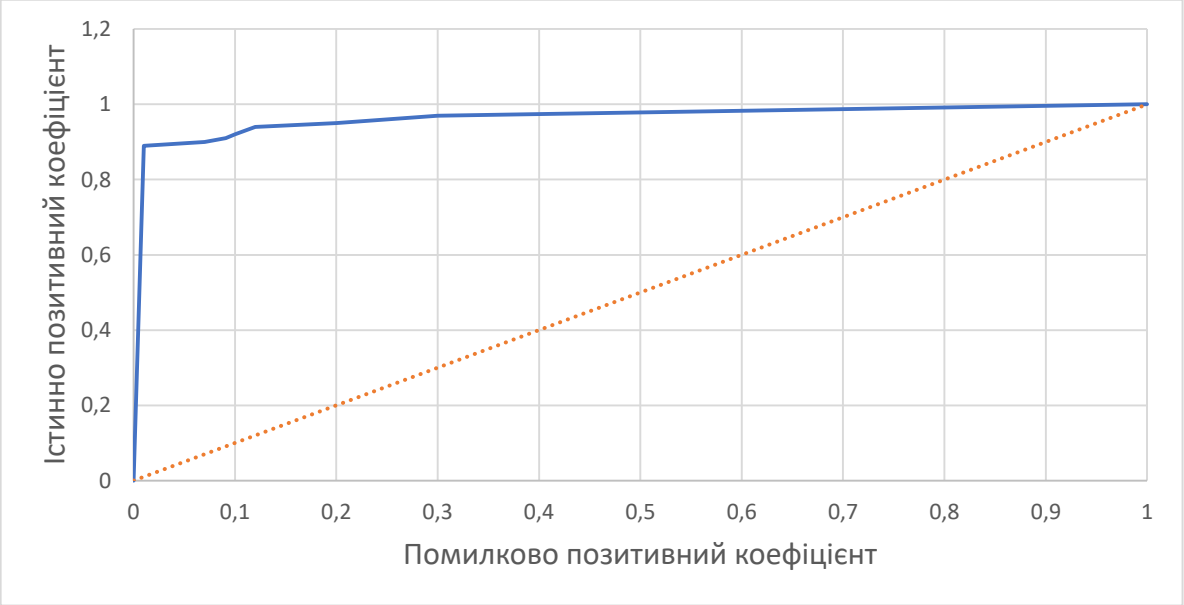


Рисунок 3.13 – Крива ROC для LSTM

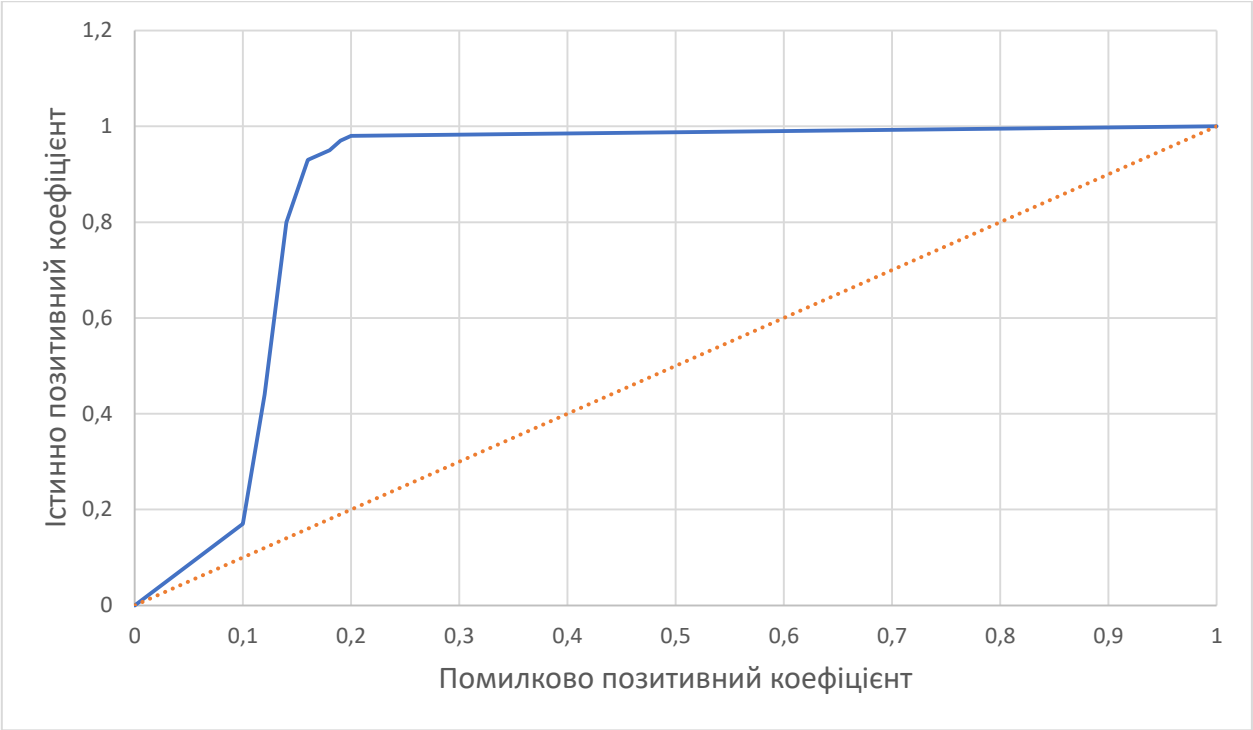


Рисунок 3.14 – Крива ROC для згорткової нейромережі

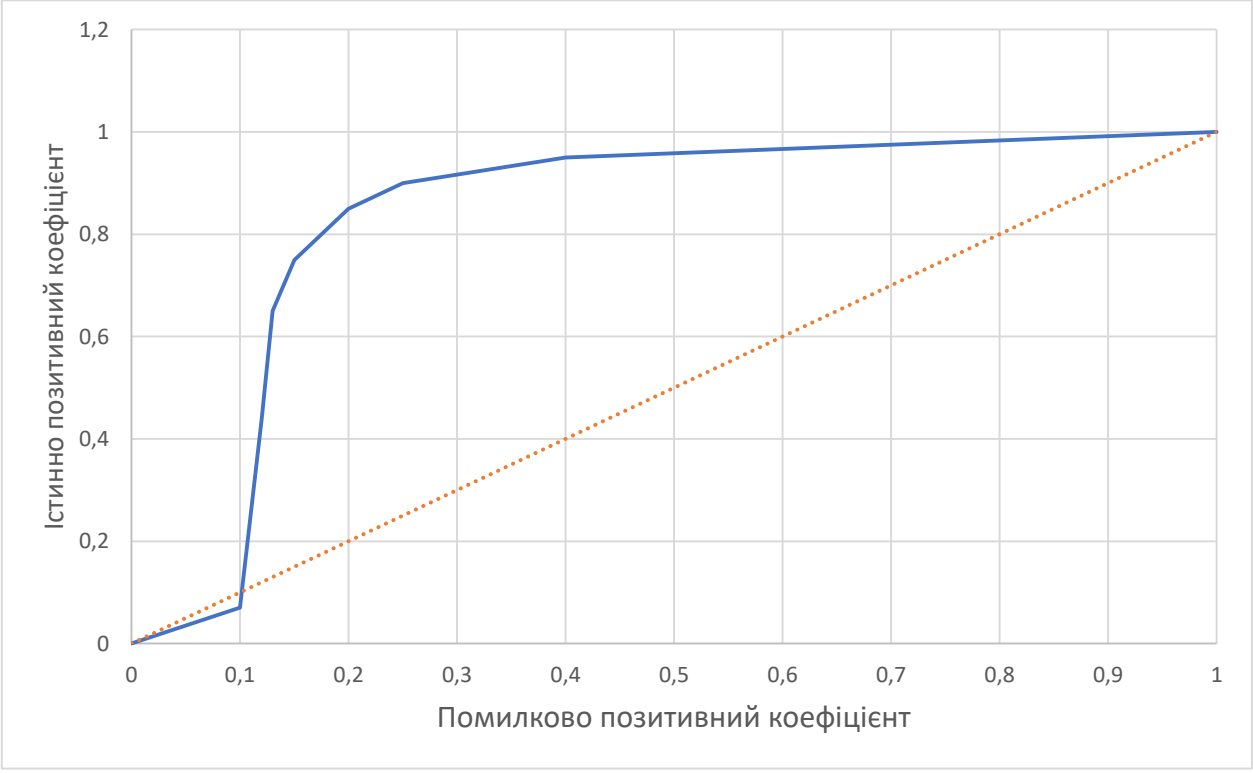


Рисунок 3.14 – Крива ROC для глибокої нейромережі

Після цього була протестована кожна з моделей на предмет швидкодії.

Час виконання моделі DNN займає лише 28 секунд, CNN - 2 хвилини 15 секунд, а LSTM - 3 хвилини 15 секунд.

Таблиця 3.3

Результати досліджень моделей глибокого навчання

| МОДЕЛЬ | ТОЧНІСТЬ | ШВИДКІСТЬ | AUC |
|---|---|---|---|
| **ГЛИБОКА НЕЙРОННА МЕРЕЖА** | 87,66% | 28с | 0,85 |
| **ЗГОРТКОВА НЕЙРОННА МЕРЕЖА** | 92,16% | 136с | 0,91 |
| **НЕЙРОННА МЕРЕЖА З ДОВГОЮ КОРОТКОЧАСНОЮ ПАМ'ЯТТЮ** | 94,42% | 194с | 0,94 |

У таблиці 3.3 продемонстровано результати досліджень моделей глибокого навчання. За отриманими показниками, можна зазначити, що згорткова нейронна мережа та НМ з довгою короткочасною пам'яттю видали достатньо вагомі результати, тому є потреба дослідити їх детальніше. Для початку використаємо дані на виході цих моделей для демонстрації у матриці плутанини, це допоможе визначити ефективність прогнозування обраних архітектур. На Рисунку зображено матрицю плутанини для прикладної моделі CNN. На рисунку 3.15 зображено матрицю плутанини для моделі LSTM:

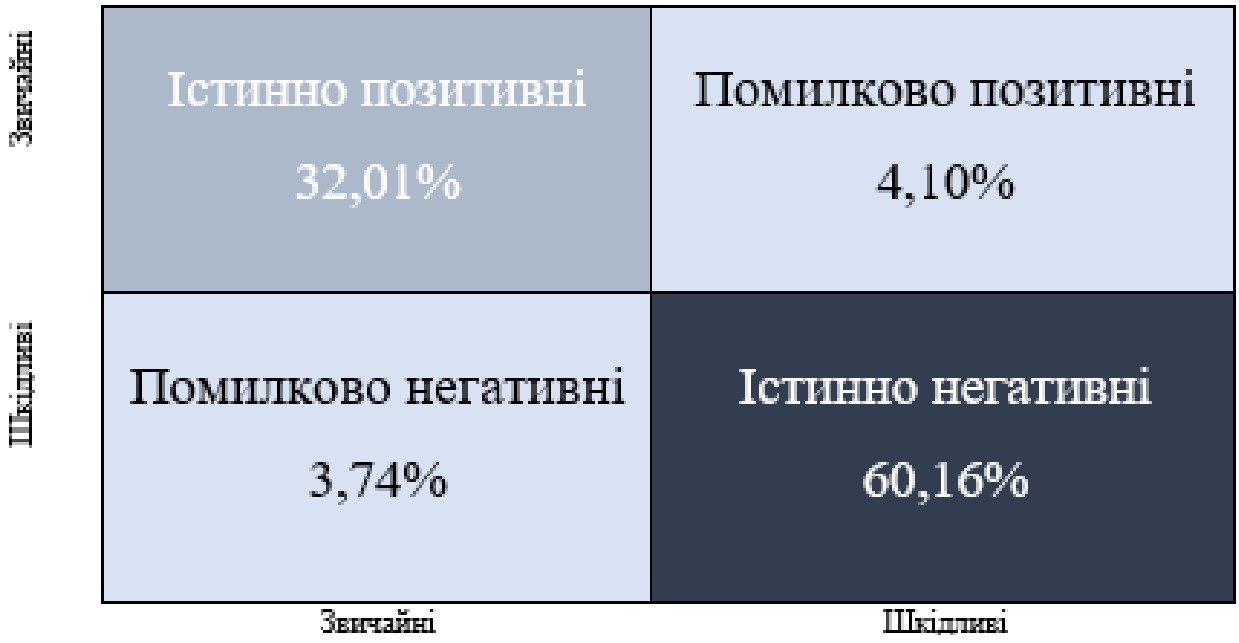


Рисунок 3.16 – Матриця плутанини для LSTM

Згідно результатів матриці плутанини модель СВВ на основі згорткової НМ видає 4,10% помилково позитивних і 3,74% помилково негативних результатів. Якщо порівнювати ці результати із неглибокими моделями, розглянутими у попередньому розділі, то це позитивний показник, але все одно недостатній.

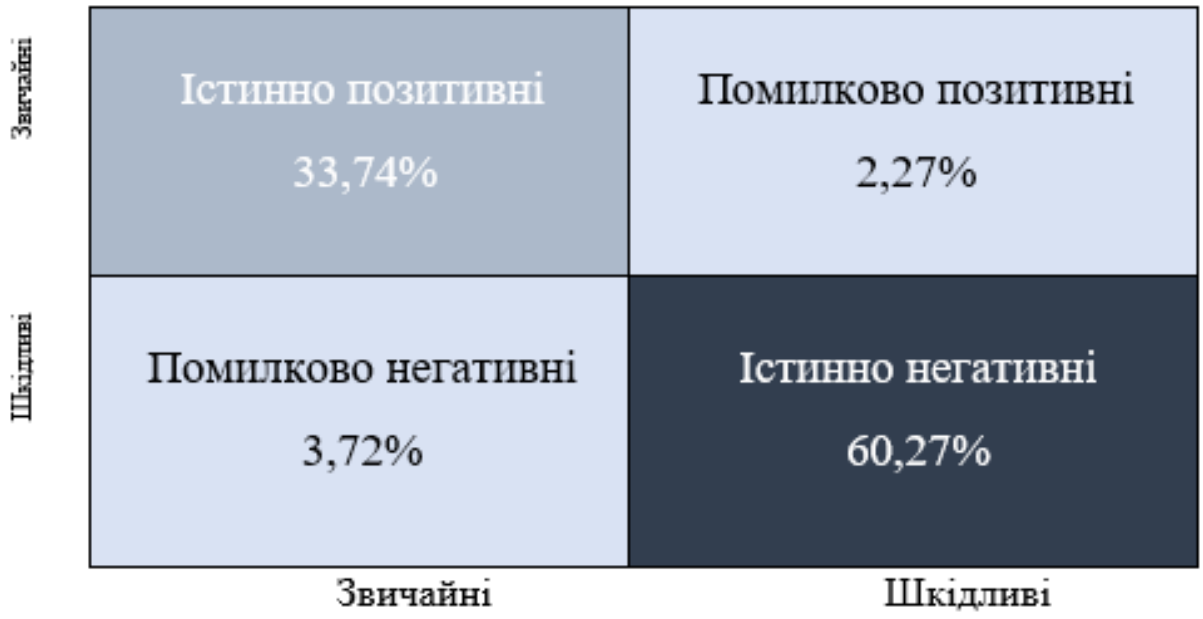


Рисунок 3.16 – Матриця плутанини для згорткової

Модель СВВ на основі LSTM демонструє ще кращі показники, проте все одно 3,72% помилково негативних значень – це багато. З іншого боку, помилково позитивне значення моделі також можна вважати високим, бо цілих 2,27% мережевих пакетів призведе до того, що атака не буде ідентифікована системою.

Як висновок, можна сказати, що моделі глибокого навчання показують кращі результати аніж звичайні архітектури МН, вони видають кращу точність, створюють більш чітке уявлення про пакет.

Проте, усі з досліджених моделей не дають показник точності достатній для того, щоб інтегрувати їх у реальні СВВ. Усі моделі пропускають достатньо високе значення помилково позитивних пакетів, що можуть привести до скритої атаки, а також витрачають зайві ресурси на оброблення виявлених помилково негативних пакетів.

Наступним кроком для дослідження буде аналіз моделі, яка комбінує декілька архітектур МН чи ГН. Можливо, поєднання декількох моделей в рази покращить результати класифікації, скоротить час обробки даних та видасть істотні показники за матрицею плутанини. В наступному пункті роботи запропоновано унікальну комбінацію архітектур для створення ефективнішої моделі виявлення аномалій.

### 3.4.3 Запропонована модель виявлення вторгнень

Як зазначалось у попередньому пункті, розумним рішенням буде поєднання декількох архітектур для створення кращої моделі. В цьому розділі пропонується зупинитися на поєднанні моделей ГН, оскільки вони в своєму стандартному вигляді видають кращі результати ніж моделі МН.

Для комбінації архітектур будемо використовувати моделі згорткової НМ та довгої короткочасної пам’яті.

Як показано на рисунку 3.17 запропонована модель СВВ буде використовувати модульний підхід до об’єднання архітектур ГН, щоб поєднати їх приховані можливості вилучення функцій, збереження пам’яті та класифікації, з метою

отримання більш високого показника точності у порівнянні з моделями, що застосовуються окремо.

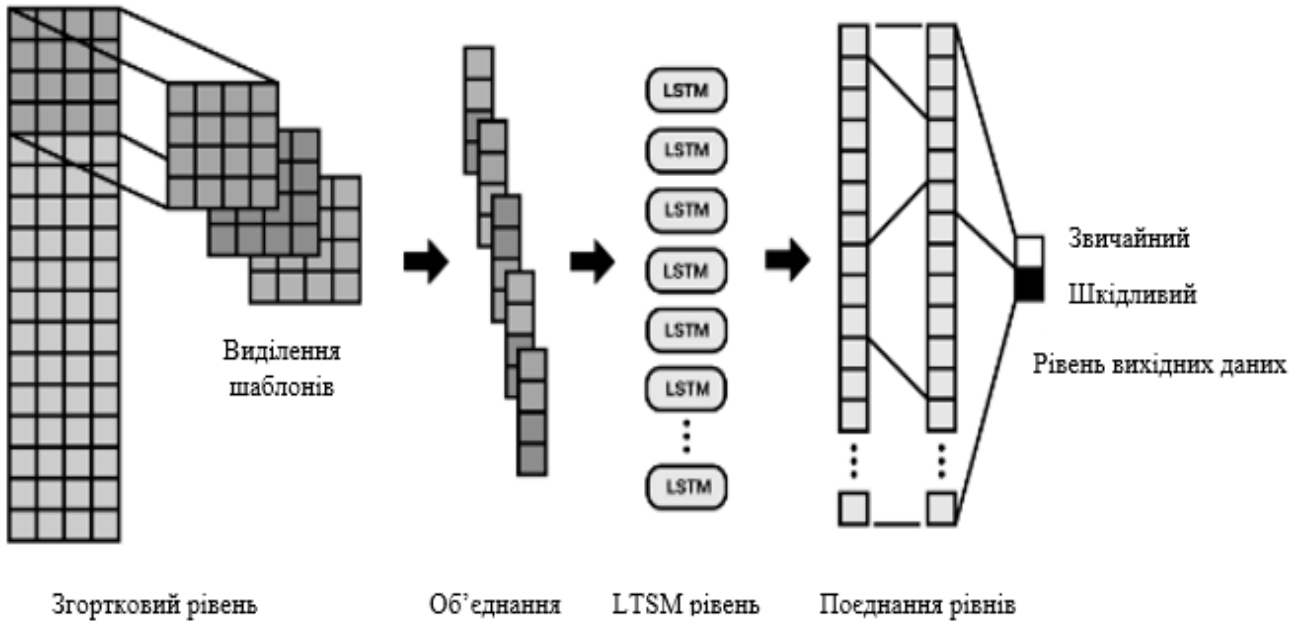


Рисунок 3.18 – Схема комбінованої моделі виявлення вторгнень на основі поєднання методів глибокого навчання

Згорткова НМ надає можливість вивчати та розпізнавати шаблони в просторі введення, а блоки НМ довгострокової короткочасної пам'яті будуть вивчати нові запропоновані дані та робити власний аналіз і висновок.

На рівні виходу також залучимо до використання глибоку НМ, яка буде вивчати вектор, що надходить до нього з попередній рівнів моделей, вивчати його, робити аналіз, і як висновок – видавати точні вихідні класи даних. Два з рівній цієї моделі, згорткова, глибок НМ – моделі прямого поширення, в яких дані можуть надходити лише в прямому напрямку.

Проте, згорткова мережа здатна отримувати функцію у двовимірному просторі та перетворювати його у вектор для подальшого представлення наступному рівню. НМ з довготривалою короткочасною пам'яттю приймає цей вектор з вже встановленими ознаками і додатково всередині своєї моделі опрацьовує його,

створює нові ваги цього вектора (ознаки), які постійно оновлюються поки мережа не опрацює всі вхідні дані. Під час процесу функціонування згортковий рівень витягує властиві функції, що знаходить у вхідних даних. LTSM рівень інтерпретує отримані на свій вхід дані через певні часові проміжки, виділяючи зв'язки, ознаки, роблячи модель більш ефективною.

Поєднання цих архітектур було досліджено в минулому, але там ці архітектури навчалися окремо, а їх результати поєднувались для отримання результату. Запропонована в цьому розділі комбінована модель навчається спільно, кожен з рівнів надає свої оброблені результати у вигляді функції на вхід наступного рівня. Така комбінована модель може перевершити інші прикладні моделі, оскільки вона виділяє та використовує сильні сторони кожної з окремо використаних архітектур.

У таблиці 3.4 наведено короткий виклад моделі поєднання згорткової нм та нм довготривалої короткочасної пам'яті.

Таблиця 3.4

Архітектура комбінованої. Детальний опис рівнів

| Тип рівня | Вихідна форма | Кількість пакетів даних |
|---|---|---|
| Conv1D | (None, 32,64) | 256 |
| Conv2D | (None, 32,64) | 12352 |
| Max_pooling1d_1 | (None, 16, 64) | 0 |
| Conv1d_3 | (None, 16, 128) | 24704 |
| Conv1d_4 | (None, 16, 128) | 49280 |
| Max_pooling1d_2 | (None, 8, 128) | 0 |
| Conv1d_5 | (None, 8, 256) | 98560 |
| Conv1d_6 | (None, 8, 256) | 196864 |

| | | |
|---|---|---|
| Max_poolin1d_3 | (None, 4, 256) | 0 |
| Lstm_1 | (None, 100) | 142800 |
| Dense_1 | (None, 256) | 25856 |
| Dropout_1 | (None, 256) | 0 |
| Dense_2 | (None, 128) | 32896 |
| Dropout_2 | (None, 128) | 0 |
| Dense_3 | (None, 1) | 129 |
| Загальна кількість пакетів | | 583,697 |

Вигода від використання згорткової мережі – зменшення дискретизації вхідних даних, при цьому зберігаючи основні функції під час процесу вилучення шаблонів. Тобто таким чином буде зменшено загальний розмір параметрів функцій. Вихідні на згортковому рівні дані на вході у рівень LSTM моделюють значення вектора відносно одиниці часу та при цьому створюється ваговий параметр за допомогою алгоритму зворотного поширення. На вході до кінцевого рівня дані передані до глибокої НМ вивчаються на предмет представлених ознак, також виділяються ознаки вищого порядку, що придатні для розподілення вихідних даних на різні мітки класів.

### 3.5 Метод навчання запропонованої моделі

Трансферне навчання — це концепція, за якої алгоритм навчання повторно використовує знання з минулих пов’язаних завдань, щоб полегшити процес навчання для виконання нового. Здатність передавати знання, отримані від попередніх завдань, має широкий спектр реальних застосувань, включаючи створення систем виявлення вторгнень у реальному часі, які можуть працювати оптимально навіть за дефіциту даних і обчислювальних ресурсів.

На рисунку 3.19 продемонстровано процес трансферного навчання а також його порівняння із традиційними методами навчання.

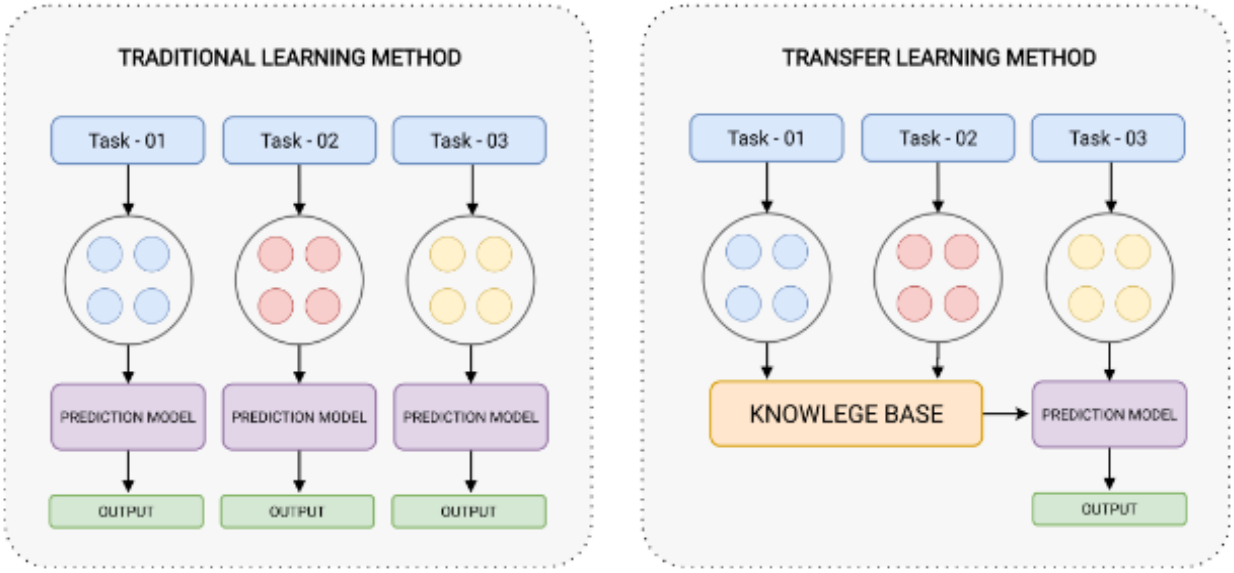


Рисунок 3.19 – Процес трансферного навчання

Традиційний підхід до навчання заснований на використанні однієї моделі НМ для виконання різних завдань подібного характеру, після чого отримуємо вихідні дані. Така архітектура оптимально працює, якщо дані, які класифікуються мають базову структуру, яку НМ навчилась ідентифікувати на етапі навчання, проте, як зазначалося у попередніх пунктах, результати будуть хибними, якщо дані які подані на вхід НМ будуть нові та ненормалізовані, тобто були відсутні на етапі навчання.

Розглядаючи методологію трансферного навчання, будуть витягнуті знання, отримані з НМ в одному або декількох попередніх завданнях. Це робиться для побудови унікальної бази знань у формі зручній для представлення НМ, щоб полегшити процес вивчення ваг векторів та їх подальшого застосування у майбутніх завданнях.

Основна перевага, яку надає така методологія полягає в можливості запуску симуляції в лабораторних умовах. Це вважається корисним, оскільки можна розвивати мережу, покращувати її продуктивність незалежно від того чи була вона інтегрована у реальне середовище чи ні.

Модель вивчає базові закономірності в іншому сегменті даних із подібним розподілом у кожній симуляції, та оптимізує характеристики для врахування всіх знань, отриманих із попередніх завдань, з метою їх застосування в майбутніх завданнях. Ефективність такої моделі не обмежена показником точності. Дана архітектура вже має первинну структуру, статичну, незмінну від попередніх завдань, тому вона не потребує більше часу, щоб починати весь алгоритм з самого початку, згідно цього – прискорює роботу системи в цілому. В роботі буде використано методологію трансферного навчання для розширення можливостей мереж, що беруть участь у дослідженнях.

Основна мета використання трансферного навчання – передати знання з первинної області визначення в цільову, пом'якшуючи припущення, що навчальні та тестові дані мають бути незалежними та однаково розподіленими. На рис. 3.20 показано процес передачі мережевої архітектури моделі та вивчених ваг з вихідної області з великим набором даних і більшими обчислювальними ресурсами в цільову область з меншим набором даних і обмеженими обчислювальними ресурсами.

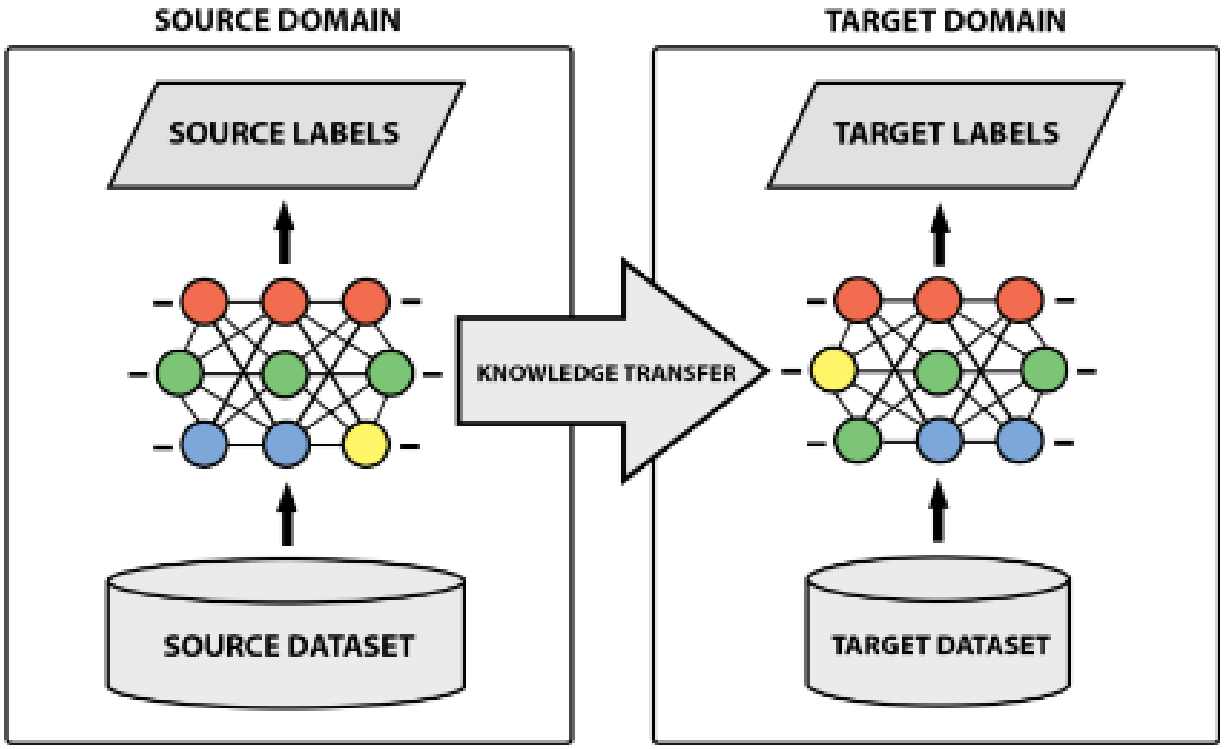


Рисунок 3.20 – Процес передачі знань з вхідної до цільової області визначення

Область визначення можна оголосити у вигляді [15] :

$$D = \{X, P(X)\} \tag{3.5}$$

що складається з двох частин: простору ознак x і розподілу P(X),

де

$$X = \{x1, x2, \ldots . xn\}, x_i \in X \tag{3.6}$$

Навчальне завдання можна представити у вигляді:

$$T = \{Y, P(Y|X)\} = \{Y, \eta\},\ Y = \{y1, y2, \ldots, y_n\}, y_i \in Y \tag{3.7}$$

де $Y$ – простір міток;

$\eta$ – функція прогнозування, яку можна дізнатися з навчальних даних, включаючи пари $\{xi, yi\}$, де $xi \in X$, $yi \in Y$.

Для кожного вектора ознак у області $\eta$ передбачено його відповідну мітку як

$$\eta(x_i) = y_i \tag{3.8}$$

Для подальших досліджень оголосимо вхідну область визначення як $DS$, а цільову область – $DT$. Дані вхідної області будемо позначати як

$$DS = \{(xS_1, yS_1), \ldots, (xS_n, yS_n)\} \tag{3.9}$$

де $xSi \in XS$ - екземпляр даних;

$ySi \in YS$ - відповідною міткою класу.

У СВВ $DS$ – це набір векторів у поєднанні з пов'язаними мітками загрози. Тому будемо позначати дані цільової області як

$$DT = \{(xT_1, yT_1), \dots, (xT_n, yT_n)\} \quad (3.10)$$

де вхід $xTi$ знаходиться в $XT$, а $yTi \in YT$ є відповідним виходом.

Враховуючи вхідну область визначення - $DS$, навчальне завдання - $TS$; цільову область визначення - $DT$ і навчальне завдання $TT$, трансферне навчання має на меті покращення процесу навчання цільової функції прогнозування $\eta t$ шляхом використання знань у вхідний області $DS$ і навчального завдання $TS$, де $DT \neq DS$ та $TS \neq TT$.

Якщо вдасться виявити зв'язок (наявний чи прихований) між просторами функцій двох областей, то можна стверджувати, що вхідна і цільова область пов'язані. Завдання трансферного навчання, визначене через $DS$, $TS$, $DT$, $TT$, $\eta t$, стає завданням глибокого трансферного навчання, якщо $\eta t$ є нелінійною функцією, представленою глибокою нейронною мережею.

### 3.6 Оцінка ефективності запропонованої моделі

Прагнучі кращого, попередньо розглянувши моделі машинного та глибокого навчання, було зроблено висновок, що потенційне поєднання декількох моделей в одну комбіновану зможе покращити показники точності класифікації. Згідно цього в попередніх пунктах роботи було запропоновано поєднання згорткової НМ, НМ довгострокової короткочастної пам'яті та глибокої НМ.

Для визначення ефективності і взагалі релевантності існування такої моделі у цьому пункті роботи надано експериментальні результати досліджень щодо застосування запропонованої архітектури в процесах СВВ, а також співставлено отримані нові результати із результатами попередньо розглянутих моделей.

На рисунку 3.21 проілюстровано результати точності класифікації запропонованої моделі у порівнянні з кращими моделями з попередніх досліджень.

Комбінована модель продемонструвала точність класифікації 98,30%, що є найвищим результатом у порівнянні з іншими прикладними моделями. Наступним кроком було використання матриці плутанини для детального вивчення процесу класифікації запропонованої моделі.

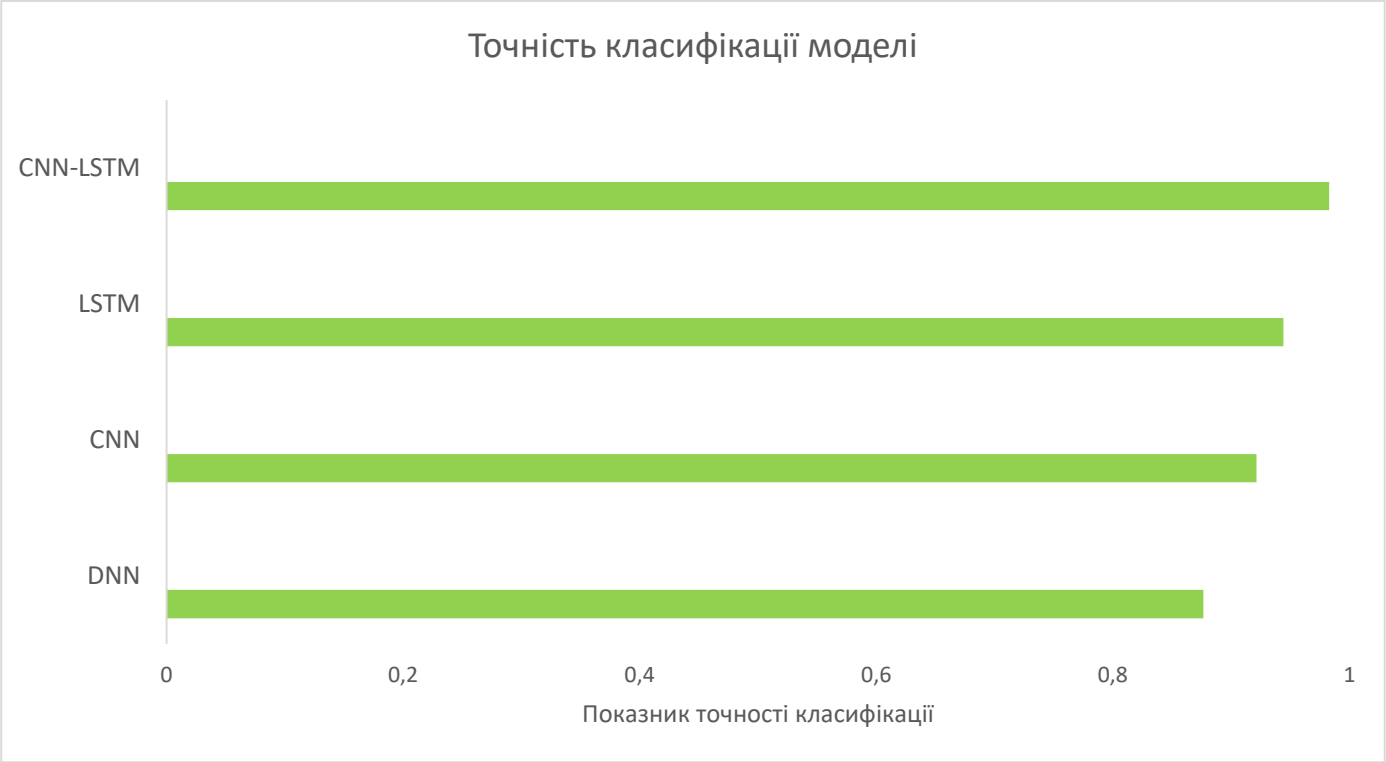


Рисунок 3.21 – Точність класифікації моделей

На основі показників матриці плутанини рисунок 3.22, комбінована архітектура демонструє покращені результати загальної класифікації помилково негативних і помилково позитивних пакетів. Модель продемонструвала 1,03% помилково позитивних результатів і 0,67% помилково позитивних результатів.

Виходячи з отриманих показників, можна сказати, що така модель працює набагато ефективніше в питанні класифікації та прогнозуванні пакетів порівняно із іншими архітектурами, а саме з найкращою з розглянутих раніше - LSTM, яка продемонструвала 2,27% помилково позитивних результатів і 3,72% помилково негативних значень.

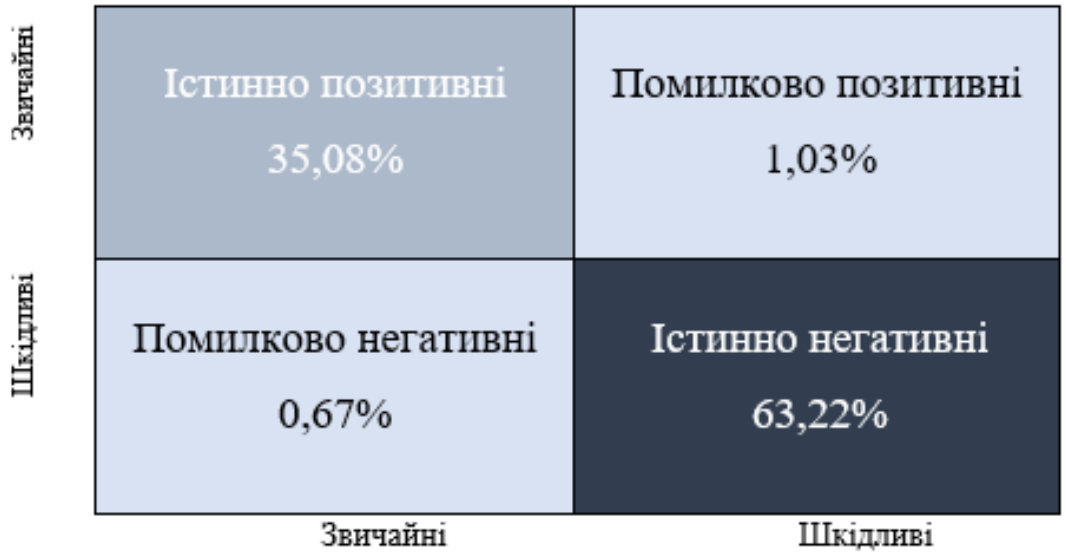


Рисунок 3.22 – Матриця плутанини для комбінованої моделі виявлення аномалій

По аналогії з попередніми дослідженнями було побудовано криву ROC для комбінованої моделі, як показано на рисунку 3.23 Крива ROC комбінованої моделі охоплює найбільшу область графіку, що демонструє здатність правильно виявляти більшу кількість наданих пакетів даних.

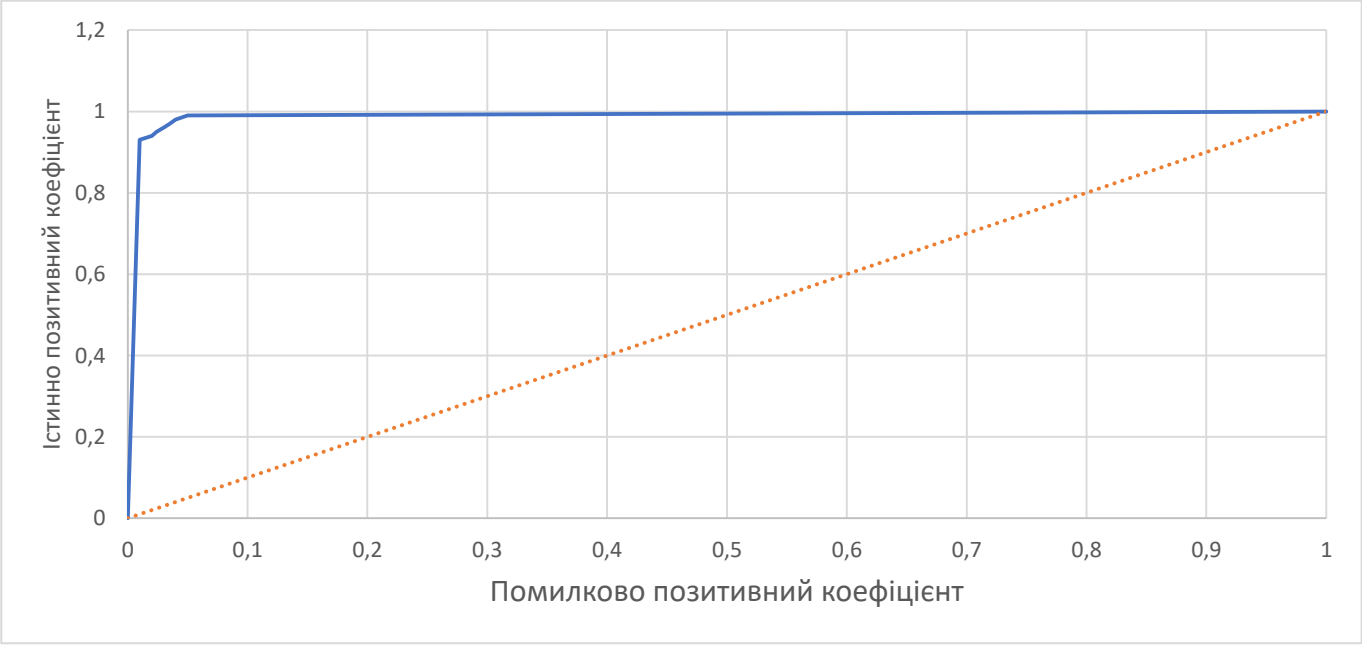


Рисунок 3.22 - Крива ROC для комбінованої моделі

Оскільки, крім точності класифікації важливим є також швидкість роботи, то потрібно розглянути дану модель на основі часу що знадобився для обробки вхідних даних.

На повну обробку даних моделі знадобилося 3 хвилини 56 секунд. Показник нижчий ніж у інших моделей, але це пояснюється складністю, кількість шарів.

Результати всіх моделей глибокого навчання, застосованих у результатах вихідної області, підсумовані в таблиці 3.5.

Таблиця 3.5

Результати моделей виявлення аномалій

| МОДЕЛЬ | ТОЧНІСТЬ | ШВИДКІСТЬ | AUC |
|---|---|---|---|
| **ГЛИБОКА НЕЙРОННА МЕРЕЖА** | 87,66% | 28с | 0,85 |
| **ЗГОРТКОВА НЕЙРОННА МЕРЕЖА** | 92,16% | 136с | 0,91 |
| **НЕЙРОННА МЕРЕЖА З ДОВГОЮ КОРОТКОЧАСНОЮ ПАМ'ЯТТЮ** | 94,42% | 194с | 0,94 |
| **CNN-LSTM НЕЙРОННА МЕРЕЖА** | 98,3% | 236с | 0,98 |

Таким чином можна зробити висновок, що в результаті дослідження запропонованої моделі можна зробити висновок, що комбінація моделей глибокого навчання перевершила окремі архітектури як машинного так і глибокого навчання. Показник точності який продемонструвала модель – 98,3%, що є найбільшим з розглянутих у роботі.

Але як вже зазначалось, мінусом даної архітектури є час роботи, який не може бути суттєво скорочений через складність комбінованої моделі.

## Висновки за розділом 3

У цьому розділі магістерської роботи проведено дослідження а також продемонстровано результати експериментів щодо питання інтеграції алгоритмів машинного, глибокого навчання у процеси СВВ.

Проведено вичерпне тестування кожної з розглянутих моделей, визначено їх показники ефективності, які дали можливість зрозуміти, що окремі моделі не можуть бути достатньо ефективними для використання у реальному середовищі, тому було прийнято рішення запропонувати унікальну модель, яка поєднує в собі три рівня, які в свою чергу є окремими моделями архітектур глибокого навчання.

Після проведення тестувань виявлено, що запропонована модель перевершила показники інших. Також для того щоб ще більше розширити можливості виявлення шкідливих пакетів використано трансферне навчання для передачі даних з вхідної області визначення в цільову. Цільова область визначення була змодельована як реальне середовище з низьким обчислюваним ресурсом та доступністю даних.

Виходячи з експериментальних результатів отриманих в даному розділі можна зробити припущення, що комплексний підхід до створення СВВ зможе покращити загальну здатність СВВ до виявлення потенційних вторгнень за відносно невеликий проміжок часу.

# ВИСНОВКИ

В роботі було виконано поставлене на початку завдання, а саме – створення рекомендацій для покращення процесу прогнозування вторгнень. Досліджено системні СВВ, виявлено їх недоліки, після чого розглянуто концепцію машинного та глибокого навчання, детально проаналізовано їх архітектури, які теоретично можуть покращити процес виявлення вторгнень.

В ході дослідження різних архітектур як машинного так і глибокого навчання, шляхом експериментальних досліджень виявлено недостатню ефективність окремих методів для їх включення у процеси СВВ, згідно чого було прийнято рішення комбінування архітектур для виведення найефективнішої моделі.

Запропонована модель може стати початковим рівнем для розробки власних архітектур, які допоможуть у виявленні шкідливих пакетів, аномалій, а також вона може бути використана як автономна СВВ, оскільки має відносно високий показник точності і може охоплювати велику кількість необроблених даних.

Крім того, в процесі досліджень, визначено ряд основних чинників які впливають на правильність рішення обраною моделлю, що можна використати як базові рекомендації для налаштування та інтеграції моделей глибокого навчання у процеси СВВ організації.

Запропоновано різноманітні методи оцінки моделі як рекомендація щодо визначення проценту точності класифікації та прогнозування трафіку. На основі наведених результатів досліджень можна адекватно оцінити та відкинути ряд моделей машинного навчання, окрім цього також не слід забувати і про системні методи СВВ, які ще до проведення основних досліджень і тестування показали себе як ненадійні.

Таким чином всі поставлені в роботі завдання виконані, мета роботи досягнута.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кавун С.В., Носов В.В., Мажай О.В. Інформаційна безпека: навчальний посібник. Ч.1. – Харків: ХНЕУ, 2008. – 352с.

2. A.Adeyemo. Design of an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS) [Електронний ресурс] / EIU Cybersecurity Laboratory. – 2018. - Режим доступу до ресурсу: https://thekeep.eiu.edu/theses/2482/

3. Khraisat, A., Gondal, I., Vamplew, P. Survey of intrusion detection systems: techniques, datasets and challenges: науково-практичний посібник - 2019. – 1-22с

4. Dinakara K. Anomaly based Network Intrusion Detection System [Електронний ресурс]. – 2016. - Режим доступу до ресурсу: http://www.facweb.iitkgp.ac.in/~pabitra/facad/06CS6026t.pdf

5. Intrusion Detection in networks and servers [Електронний ресурс]. – 2015. - Режим доступу до ресурсу: https://www.leadingindia.ai/downloads/projects/CS/cs_1.pdf

6. H.N.Aludhilu. An Intrusion Detection System using recurrent neural network with a real-world dataset [Електронний ресурс]. – 2021. - Режим доступу до ресурсу: https://repository.unam.edu.na/bitstream/handle/11070/3225/Aludhilu_2020.pdf?sequence=1&isAllowed=y

7. Ahmed M, A survey of network anomaly detection techniques [Електронний ресурс] / Journal of Network and Computer Applications. – 2015. – Режим доступу до ресурсу: http://dx.doi.org/10.1016/j.jnca.2015.11.016i

8. M. Mazini, B. Shirazi. Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. [Електронний ресурс]. – 2012. - Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S1319157817304287

9. Івановіч І.С., Гайін С.В. Recommendations for Network Traffic Analysis Using the NetFlow Protocol: науково-практичний посібник – 2016. - 40-42с.

10. M. Almgren. Introduction to Intrusion detection [Електронний ресурс]. - 2004. - Режим доступу до ресурсу: https://indico.cern.ch/event/417514/attachments/862479/1206510/kth-2004-intro-to-ids-JRA3.pdf

11. Komoldinovich, A. J. (2022). Intelligent System for Information Security Management: Architecture and Design Problems. European Multidisciplinary Journal of Modern Science. - 383-397c.

12. Feedforward neural network [Електронний ресурс]. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Feedforward_neural_network

13. Dumas, M., Schwartz L. Principles Of Computer Networks And Communications: науково-практичний посібник – 2009.

14. Hawkins S., He H., Williams G., Baxter R. Outlier detection using replicator neural networks: лекція по комп'ютерних науках видання 244 – Berlin: Springer, 2002. - 170–80c.

15. Jurafsky D., Martin J. Speech and Language Processing. Spotlight [Електронний ресурс] / Stanford. – 2016. – Режим доступу до ресурсу: https://web.stanford.edu/~jurafsky/slp3

16. The Global Risks Report. [Електронний ресурс] / World Economic Forum. – 2019. – Режим доступу до ресурсу: https://www.securitysales.com/access/covid-19-access-control-spotlight/.

17. CIRA Cybersecyrity Report. [Електронний ресурс] / Canadian Internet Registration Authority. – 2020. – Режим доступу до ресурсу: https://www.cira.ca/resources/cybersecurity/report/2019-cira-cybersecurity-survey

18. Word Threat Assessement Report [Електронний ресурс] / US Intelligence Community. – 2017. – Режим доступу до ресурсу: https://www.dni.gov/files/documents/Newsroom/Testimonies/SSCIUnclassifiedSFR-Final.pdf

19. Laptev N., Xiang D. Information-theoretic measures for anomaly detection. IEE symposium on security and privacy – 2001. – 43-130c.

20. H.Hindy. Intrusion Detection Systems using Deep Learning techniques [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://rke.abertay.ac.uk/ws/portalfiles/portal/33845351/Hindy_IntrusionDetectionSystemsUsingMachineLearning_PhD_2021.pdf

21. Zhou C., Sun C., Liu Z., Lau F. A C-LSTM neural network for text classification [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://arxiv.org/abs/1511.08630

22. Estevez-Tapiador J.M., Garcia-Teodoro P., Diaz J.E. Anomaly detection methods in wired networks [Електронний ресурс] / Comput Commun.. – 2004. – Режим доступу до ресурсу: https://www.researchgate.net/publication/222396636_Anomaly_detection_methods_in_wired_networks_A_survey_and_taxonomy

23. Noble CC, Cook DJ. Graph-based anomaly detection. Ninth ACM SIGKDD international conference on knowledge discovery and data mining. – New York, 2003. – 6-631c.

24. Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. Gradient based learning adpplied to document recognition. IEE vol. 86, no 11 – 1998. – 2278-2324c.

25. Eugenio Culurciello. Архітектури нейронних мереж [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://habr.com/ru/company/nix/blog/430524/

26. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrussion detection systems (UNSW-NB15 network dataset) Military communications and information systems conference. – 2015. – 1-6c.

27. Krizhevsky A., Sutskever I., Hilton G. E. Imagenet classification with deep convolutional neural networks: навчальний посібник. – 2012

28. Injadat M., Moubayed A., Nassif A., Shami A. Multi-stage optimized machine learning framework for network intrussion detection [Електронний ресурс] / IEEE Transations on network and service management.. – 2020. – Режим доступу до ресурсу: https://arxiv.org/abs/2008.03297

# ДОДАТОК А

# ЛІСТИНГ ІМПОРТУ ТА НОРМАЛІЗАЦІЇ ВХІДНИХ ДАНИХ

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
import glod
import matplotlib.pyplor as plt
from keras.models import Sequential
from keras.layer import Dense,Reshape,Conv2D,Flatten,MaxPoolingID,ConvID,LSTM
from keras import optimizers
from keras.utils import to categorical
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall
score, f1_score



#view of a sample of data
from sklearn import preprocessing
import glob
path = r`/content/drive/My Drive/dataset/AIBenchmark`
all_files=glod.glob(path+"/*csv")

dataset_conc=[]
for filename in all files:
        df=pd.read_csv(filename,index_col=None,header=0)
        df=df.replace(0,np,nan)
        df=df.dropna(axis=0, how='any', subset=['value']
        df.value = preprocessing.normalize([df.value]).T
        dataset_conc.append(convert2d(df))
frame=pd.concat(dataset_conc,axis=0,ignore_index=True)
```

# ДОДАТОК Б

## СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ

**Тези наукових доповідей:**

1. Кирило Золотарьов, Володимир Наконечний. Нейронні мережі як засіб для виявлення аномалій трафіку. Проблеми кібербезпеки інформаційно-телекомунікаційних систем. Modern biometric technologies in security access control systems. VI Міжнародна науково-практична конференція. 2023.,