Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«МОДЕЛЬ І МЕТОД ВИЯВЛЕННЯ КІБЕРАТАК В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ НА БАЗІ АЛГОРИТМІВ ШТУЧНОГО ІНТЕЛЕКТУ»

Виконав: студент 2 курсу, групи 1БС-22м
спеціальності 125 Кібербезпека
_____ Анатолій ДИМОВ
Керівник: к. т. н., доцент каф. ЗІ
_____ Вадим МАЛІНОВСЬКИЙ
«18» 12 2023 р.
Опонент: к. т. н. доцент каф. ОТ
_____ Людмила ЛІЩИНСЬКА
«18» 12 2023 р.

Допущено до захисту

Завідувач кафедри ЗІ

д. т. н., проф.

_____ Володимир ЛУЖЕЦЬКИЙ

«18» 12 2023 р.

Вінниця 2023

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Рівень вищої освіти ІІ-й (магістерський)
Галузь знань 12 Інформаційні технології
Спеціальність 125 Кібербезпека
Освітня програма – Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д. т. н., проф.
В. А. Лужецький
«19» 09 2023 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Димову Анатолію Олександровичу

1. Тема роботи: «Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту» керівник роботи: Маліновський Вадим Ігорович, к. т. н., доцент кафедри ЗІ, затверджені наказом ректора ВНТУ №247 від 18.09.2023р.
2. Строк подання студентом роботи: 13 грудня 2023р.
3. Вихідні дані до роботи:
   - тип засобу і моделі виявлення кібератак в ІМ/ІС: нейромережевий.
   - тип і швидкість виявлення кібератак: у режимі реального часу.
4. Зміст текстової частини: Вступ. 1. Аналіз моделей і методів виявлення кібератакна базі алгоритмів штучного інтелекту. 2. Розробка удосконаленого методу виявлення кібератак на основі моделей нейронних мереж та штучного інтелекту 3. Програмна реалізація моделі та методу аналізатора трафіку в ІКС. 4. Тестування та оцінка ефективності функціонування розробленої моделі та методу. 5. Економічна частина. Висновки. Перелік використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу: Алгоритм роботи аналізатора трафіку і виявлення кібератак(плакат А4). Реалізація циклу по розміру порції(плакат А4). Модель побудови технології аналізу трафіку на базі нейропідходів(плакат А4). Базова модель безпеки ІКС на базі ШІ(плакат А4). Функція створення глибокої архітектури нейронної мережі(плакат А4). Функція створення шуму(плакат А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Вадим МАЛІНОВСЬКИЙ, к. т. н., доц. каф. ЗІ	19.09	30.10
2	Вадим МАЛІНОВСЬКИЙ, к. т. н., доц. каф. ЗІ	19.09	20.11
3	Вадим МАЛІНОВСЬКИЙ, к. т. н., доц. каф. ЗІ	19.09	1.12
4	Вадим МАЛІНОВСЬКИЙ, к. т. н., доц. каф. ЗІ	19.09	6.12
5	Ольга РАТУШНЯК, к.т.н., доц.каф ЕПВМ	19.09	12.12

7. Дата видачі завдання 1 вересня 2023року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2023 – 10.09.2023	
2	Аналіз літературних джерел за напрямком магістерської роботи	10.09.2023 – 15.09.2023	
3	Розробка рішень	16.09.2023 – 22.09.2023	
4	Розробка модуля програмного засобу	30.09.2023 – 12.10.2023	
5	Практична реалізація, моделювання, експериментування, результати	14.10.2023 – 10.11.2023	
6	Розробка розділу тестування і обґрунтування доцільності розробки	11.11.2023 – 17.11.2023	
7	Аналіз виконання, висновки	18.11.2023 – 24.11.2023	
8	Оформлення пояснювальної записки	25.11.2023 – 30.11.2023	
9	Попередній захист та доопрацювання МКР	28.11.2023 – 10.12.2023	
10	Представлення МКР до захисту	11.12.2023 – 14.12.2023	
11	Захист МКР	14.12.2023 – 21.12.2023	

Студент_____Анатолій ДИМОВ

Керівник роботи_____Вадим МАЛІНОВСЬКИЙ

АНОТАЦІЯ

УДК 004.056

Димов А.О. Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту. Магістерська кваліфікаційна робота зі спеціальності 125 – Кібербезпека, освітня програма – Безпека інформаційних і комунікаційних систем. – Вінниця: ВНТУ, 2023. – 85 с.

На укр. мові. Бібліогр: 17 назв; рис.: 20; табл. 9.

В магістерській кваліфікаційній роботі представлено розробку методу і моделі виявлення кібератак для підвищення рівня захисту в інформаційно-комунікаційних систем на базі алгоритмів штучного інтелекту. Запропоновані в роботі метод і модель є інноваційними і передбачають вдосконалення вже існуючих підходів підвищення безпеки, а також реалізацію програмним шляхом. Для успішної розробки програмного засобу проведено дослідження наявних аналогів моделей та програмних реалізацій систем збирання та аналізування. Під час роботи обґрунтовано вибір власних методів, розроблено ряд схем і алгоритмів, здійснено програмну реалізацію моделі. Проведено тестування программного засобу на коректність роботи.

Ілюстративна частина складається з 6 плакатів.

В економічному розділі оцінено витрати на роботу.

Ключові слова: кібератаки, алгоритми штучного інтелекту, система виявлення, нейронні мережі.

ABSTRACT

Dymov A.O. Model and method of detecting cyberattacks in information and communication systems based on artificial intelligence algorithms. Master's qualification work in specialty 125 - Cybersecurity, educational program - Security of information and communication systems: VNTU, 2023. – 85 p.

In Ukrainian. Bibliogr: 17 titles; Figures: 20; Table 9.

The master's thesis presents the development of a method and model for detecting cyberattacks to improve the level of protection in information and communication systems based on artificial intelligence algorithms. The method and model proposed in the work are innovative and provide for the improvement of existing approaches to improving security, as well as implementation by software. For the successful development of the software tool, a study of existing analogues of models and software implementations of collection and analysis systems was conducted. In the course of the work, the choice of our own methods was substantiated, a number of schemes and algorithms were developed, and the model was implemented in software. The software tool was tested for correct operation.

The illustrative part consists of 6 posters.

The economic section estimates the costs of the work.

Key words: cyberattacks, artificial intelligence algorithms, detection system, neural networks.

ЗМІСТ

ВСТУП

В останні роки у світі суспільний розвиток характеризується формуванням інформаційного суспільства. Нова епоха інформаційного суспільства принесла із собою не лише позитивні трансформації, але й сприяла зростанню негативних наслідків. Основними недоліками стали ризики кібератак, які набувають все більших масштабів. Так, за один день у будь-якій інформаційно-комунікаційній системі виникає велика кількість подій, частина з яких є кібератаками. Наявність кіберінцидентів в інформаційно-комунікаційній системі свідчить про розвиток кібератак або вже їх здійснення. Актуальність теми полягає в тому, що використання штучного інтелекту є одним із вагомих факторів, що сприяє інформаційній безпеці. Адже штучний інтелект – це один із трендових напрямків, який охоплює всі розвинені країни світу. Системи штучного інтелекту допомагають посилити кібербезпеку: розпізнають аномалії, сповіщають про загрози та захищають дані. Під штучним інтелектом розуміють сукупність технологічних рішень, які дозволяють імітувати когнітивні функції людини і одержувати під час виконання конкретних завдань результати, рівні результатам інтелектуальної діяльності [1]. Активний розвиток інформаційних технологій визначає актуальність дослідження проблем інформаційної безпеки, а саме: загрози інформаційним ресурсам, різноманітним засобам та заходам захисту; бар'єри проникнення; вразливість у системі захисту інформації.

Загалом під інформаційною безпекою слід розуміти сукупність засобів, моделей та методів, які забезпечують захист інформаційних даних і, як наслідок, гарантують безпеку, як інформаційних систем, так і відомостей, які в таких системах зберігаються й обробляються.

Отже, використання штучного інтелекту в системі забезпечення безпеки інформаційно-комунікаційних систем, оцінки та аналізу інформаційних загроз та практичне застосування штучного інтелекту з метою захисту інформації, що зберігається та обробляється в інформаційно- комунікаційних системах, набуває

особливого значення та актуальності у спектрі відносин в інформаційному суспільстві.

Магістерська кваліфікаційна робота присвячена темі виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту, та спрямована на аналіз і вдосконалення моделей та методів забезпечення виявлення та усунення інформаційних загроз із використанням штучного інтелекту. У дослідженні розглянуто сучасні нейромережеві технології, а також проведено їх аналіз на прикладі мережі Кохонена, Персептрона та типу нейромереж Трансформер тощо [2]. У роботі відзначено сучасні технології, завдяки яким були створені різні алгоритми роботи штучного інтелекту, які використовуються задля виявлення інформаційних вкидів та кібератак.

У роботі відзначено сучасні технології, завдяки яким були створені різні алгоритми роботи штучного інтелекту, які використовуються задля виявлення інформаційних інцидентів: кіберзагроз та кібератак по їх характерним ознакам.

Об'єктом дослідження є процес інформаційного захисту в інформаційно-комунікаційних систем за допомогою методу і моделі виявлення кіберзагроз на базі штучного інтелекту та нейромереж.

Предметом дослідження є модель і метод виявлення кіберзагроз на базі підходів і алгоритмів штучного інтелекту і нейромережевих моделей і алгоритмів.

Метою роботи є підвищення рівня інформаційної захищеності інформаційно-комунікаційних систем із використанням моделі і методу виявлення кіберзагроз на базі алгоритмів штучного інтелекту та нейромережевих підходів.

Для досягнення мети передбачається вдосконалити метод і модель виявлення кіберзагроз по характерним ознакам у трафіку даних інформаційно-комунікаційних систем (ІКС) для забезпечення підвищення інформаційного захисту, а також демонстрації важливості і можливості реалізації цих напрямків.

Наукова новизна магістерської роботи полягає в тому, що:

Удосконалено метод і модель виявлення кіберзагроз у трафіку даних інформаційно-комунікаційних систем по характерним ознакам на базі алгоритмів і

принципів штучного інтелекту, що дозволило підвищити рівень та діапазон виявлення кіберзагроз, та на відміну від відомих включає додаткові етапи аналізу трафіку і двохстадійний процес аналізу трафіку.

Практичне значення одержаних результатів:

- удосконалено алгоритм та загальну структуру роботи аналізатора трафіку і засобу виявлення кіберзагроз у трафіку даних в інформаційно-комунікаційних систем, що працює на базі моделі нейромережі із ознаками і алгоритмами штучного інтелекту , що на відміну від відомих має окремі блоки аналізу за допомогою нейромережевих підходів із ознаками штучного інтелекту для задач виявлення кібератак та кіберзагроз на інформаційний ресурс;

- розроблено програмну реалізацію, яка дозволяє моделювати процес виявлення кіберзагроз за допомогою програмних методів на базі запропонованих підходів і методу виявлення кіберзагроз на базі алгоритмів штучного інтелекту. Це дозволило проводити прості моделювання процесу виявлення кіберзагроз нейромережевими методами та проілюструвати сам процес ефективності та актуальності запропонованих підходів.

Апробація результатів магістерської роботи автором наведено у роботі [3].

1 АНАЛІЗ МОДЕЛЕЙ І МЕТОДІВ ВИЯВЛЕННЯ КІБЕРАТАК НА БАЗІ АЛГОРИТМІВ ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Аналіз моделей виявлення кібератак

Системи штучного інтелекту допомагають посилити кібербезпеку, виявляючи аномалії та нове шкідливе програмне забезпечення, сповіщаючи про загрози та захищаючи наднизькі дані. Ці системи побудовані на основі алгоритмів машинного навчання та лінгвістичних нейронних мереж і вже давно є частиною сучасних засобів захисту даних.

Штучний інтелект допомагає зміцнити кібербезпеку, виявляючи аномалії та нове шкідливе програмне забезпечення, сповіщаючи про загрози та захищаючи наднизькі дані.

Класичні системи штучного інтелекту працюють шляхом виявлення відомих загроз. Але нові методи злому з'являються чи не щодня. Тому важливо мати можливість передбачити зловмисну поведінку та реагувати на неї до того, як уразливість стане активною. Інструменти, засновані на генеративних моделях штучного інтелекту, можуть впоратися з цим завданням, не тільки вимірюючи певні шаблони, але й генеруючи контент. Це перспективний напрям для створення антивірусних та комерційних продуктів кібербезпеки [4].

У задачах штучного інтелекту часто необхідно зменшити розмірність моделей машинного навчання. Це робиться за допомогою:

- метод головних компонент (PCA) (метод факторного аналізу);
- сингулярне розкладання (SVD);
- T-Distributed Stochastic Neighbor Nesting (T-SNE) (для візуалізації великовимірних даних у 2D та 3D просторах) [4];
- лінійний дискримінантний аналіз (LDA) (пошук лінійних комбінацій ознак, які відрізняють різні класи. Пов’язаний з регресійним аналізом і PCA, працює

за умови, що вимірювання незалежної змінної для кожного спостереження є безперервним);

– прихований семантичний аналіз (LSA) (направлення тексту та аналіз вмісту);
– факторний аналіз (FA) (методи: метод головних компонент, кореляційний аналіз, метод максимальної правдоподібності, нечіткий факторний аналіз);
– незалежний компонентний аналіз (ICA) (вибір певного типу корисного сигналу в задачі перехоплення витоків через технічні канали, наприклад, прослуховування в шумному приміщенні)[1];
– Integral Matrix Factorization (NMF) (обробка звукових сигналів, кластеризація, комп'ютерне бачення). Ключові переваги та можливості генеративних систем штучного інтелекту у сфері кіберзахисту зображено на рисунку 1.1.

Рисунок 1.1 – Генеративна модель штучного інтелекту

– виявляти атипову (аномальну) поведінку. У класичних системах безпеки ми встановлюємо певні критерії, за якими поведінка користувача буде вважатися зловмисною. Якщо кілька з цих факторів збігаються, виявляється вторгнення. У той же час генеративний ШІ може помічати стандарти, про які ми самі не знаємо і з якими ніколи раніше не стикалися. Таким чином можна поєднати найкращу прогностичну модель [5].

- автоматичні дії. Просто ідентифікувати загрози недостатньо, надзвичайно важливо своєчасне реагування та впровадження систем захисту. Моделі, створені генеративними алгоритмами ШІ, можуть зробити це якомога швидше.
- використання автоматичних атак. Здебільшого великі компанії атакують не люди, а алгоритми. Часто люди не можуть належним чином вирішити цю проблему, оскільки зловмисне програмне забезпечення постійно змінюється. Однак одного робота можна і потрібно порівнювати з іншим.
- боротьба з помилково негативними та помилково позитивними сигналами. У світі корпоративної кібербезпеки системи повинні відстежувати багато подій одночасно: Інтернет-трафік, відвідування веб-сайтів тощо. У цьому «шумі» легко посилити та витлумачити нормальну поведінку як зловмисну і навпаки. Зараз це серйозна проблема кібербезпеки, і створення штучного інтелекту може зменшити кількість неправильно витлумачених сигналів [6].

Таким чином, генеративні моделі представляють окремий клас моделей штучного інтелекту, призначених для моделювання фактичних даних на основі попередніх рішень і даних.

Метою регресійної моделі є прогнозування значення однієї або кількох безперервних цільових змінних t з урахуванням значень D-вимірного вектора вхідних змінних x (лінійна регресія). Найпростішою формою моделі лінійної регресії також є лінійна функція вхідних змінних [7].

Дано навчальний набір даних, що містить N спостережень $\{xn\}$, де $n = 1,. . , N$ разом із відповідними значеннями $\{tn\}$ мета полягає в тому, щоб передбачити значення t для нового x.

У найпростішому підході це можна зробити шляхом безпосередньої побудови відповідної функції $y(x)$, значення якої для нових вхідних даних x робиться шляхом передбачень для відповідних значень t.

Загалом, з ймовірнісної точки зору, ми прагнемо змоделювати прогнозний розподіл $p(t|x)$, оскільки це виражає нашу невизначеність щодо значення t для

кожного значення *x*. З цього умовного розподілу ми можемо робити прогнози *t* для будь-якого нового значення x таким чином, щоб мінімізувати очікуване значення відповідно до обраної функції втрат.

Регресійну модель можна представити у вигляді [7]

$$y(x, w) = w_0 + w_1 x_1 + \cdots + w_D x_D. \quad (1.1)$$

Також можна представити клас моделей виду:

$$y(x, w) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(x). \quad (1.2)$$

Серед прикладів використання такої моделі можна виділити:

- прогнози;
- аналітика дієвості політики безпеки;
- аналітика ризиків;
- задачі класифікації.

1.2 Аналіз методів виявлення кібератак

Використання алгоритмів штучного інтелекту для виявлення мережевих атак в інформаційно-комунікаційних системах є важливим напрямком забезпечення безпеки мережі. Для виявлення аномалій, атак та інших загроз у мережах і системах можна використовувати різні методи та техніки. Ось деякі ключові аспекти та технології в цій галузі:

- Машинне навчання. Трененоване навчання моделі машинного навчання можна навчити на історичних даних для визначення нормальної та ненормальної поведінки. Наприклад, для виявлення відхилень від нормального трафіку можна використовувати такі алгоритми класифікації, як опорні векторні

машини (SVM) або нейронні мережі. Наприклад, для виявлення відхилень від нормального трафіку можна використовувати такі алгоритми класифікації, як опорні векторні машини (SVM) або нейронні мережі.

- Навчання без учителя. Методи кластеризації та алгоритми виявлення аномалій (такі як Isolation Forest) можуть ефективно виявляти ненормальні моделі в даних без попереднього маркування.
- Штучні нейронні мережі. Глибокі нейронні мережі, включаючи згорткові та рекурентні мережі, можна використовувати для аналізу трафіку, виявлення атак і прийняття рішень у реальному часі [8].
- Статистичні методи. використовуйте статистичні методи, такі як тест Ширяєва-Роберта, щоб виявити аномалії в даних трафіку.

Використання моделей часових рядів та здатність виявляти аномалії на основі змін у часі, наприклад відхилення від типових часових патернів, технологія споживає відчутно менше ресурсів під час роботи, але потрібний досвід і глибоке розуміння специфіки конкретної мережної інфраструктури.

- Системи виявлення вторгнень (IDS). Поділяються на два типи такі як Мережеві IDS та Хост-засновані HIDS. Мережевим IDS присутній саме моніторинг мережного трафіку для виявлення аномалій та атак. Щодо Хост-засновані HIDS, то задля виявлення потрібно забезпечувати моніторинг активності на конкретних комп'ютерах чи пристроях. Хостові системи, на відміну від мережевих, встановлюються «точково» на кожен окремо взятий хост в рамках мережі, дозволяючи забезпечити вибірковий захист вузлів, що піддаються атаці. HIDS теж здатна аналізувати вхідний та вихідний трафік, але робить це локальніше, для одного пристрою.

  Установку HIDS рекомендують здійснювати на критично важливих машинах у мережі для запобігання загрозам [9]. Загальна структура роботи порівняно з IPS зображена на рисунку 1.2. [9].

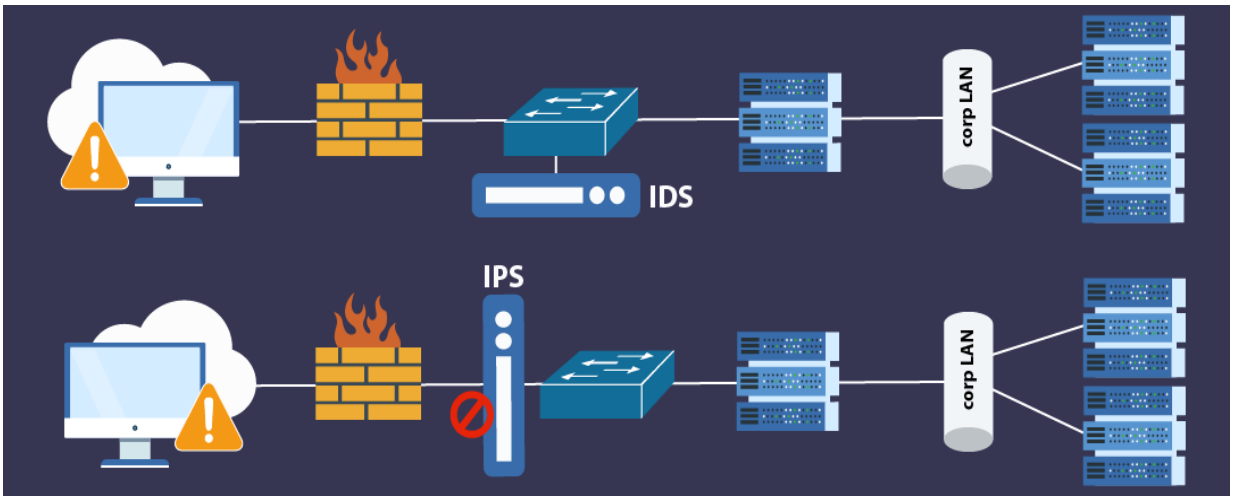


Рисунок 1.2 – Загальна структура роботи порівняно з IPS

Дана технологія споживає відчутно менше ресурсів під час роботи, але потрібний досвід і глибоке розуміння специфіки конкретної мережної інфраструктури, щоб правильно вибрати цільові хости для HIDS.

– Обробка природної мови (NLP). Аналіз логів та текстів. Використання методів обробки природної мови для аналізу логів та текстових даних для виявлення потенційних загроз.
– Системи управління подіями та інформацією (SIEM). Інтеграція даних. Використання SIEM-систем для збирання, агрегації та аналізу даних із різних джерел для виявлення підозрілих активностей.
– Аналіз поведінки користувачів. Профілювання користувачів. Відстежування поведінки користувачів і виявляйте відхилення від їх типової поведінки.

Важливо розуміти, що поєднання різних методів і підходів може бути найефективнішою стратегією виявлення кібератак. Крім того, системи необхідно регулярно оновлювати та адаптувати до нових загроз, використовуючи останні дані для навчання моделей і вдосконалення методів виявлення.

1.3 Виявлення кібератак за допомогою нейромереж

Основним методом обробки експериментальних даних і встановлення правил прийняття рішень для багатопараметричного моніторингу композитних виробів є

використання штучних нейронних мереж. Існує багато наукових робіт, що описують і підтверджують ефективність використання нейронних мереж для первинної обробки інформаційних сигналів ідентифікації сигналів або розділення сигналів на фоні шуму. Однак це не єдиний напрямок їх можливого застосування в задачах: навчені нейронні мережі не тільки здатні розпізнавати (класифікувати) сигнали, отримані від датчиків під час процесів керування, але й здатні зберігати важливу інформацію про закономірності та зв’язки між формами. інформація. Керуйте сигналами та станами об'єктів. Це дозволяє нейронній мережі правильно класифікувати нові сигнали та можливі дефекти, які раніше були невідомі та не виникали під час навчання, дозволяючи їй динамічно розширювати власну базу знань [10].

Тому однією із сфер застосування нейронних мереж є багатопараметричний моніторинг для прийняття рішень.

Використання штучних нейронних мереж для виявлення мережевих атак в інформаційно-комунікаційних системах є важливим напрямком у сфері мережевої безпеки. Взагалі кажучи, штучні нейронні мережі здатні виявляти аномалії або зловмисні моделі в трафіку або поведінці системи, що робить їх потужними інструментами проти сучасних кіберзагроз.

Залежно від конкретного завдання та характеристик даних для виявлення кібератак в інформаційно-комунікаційних системах можуть використовуватися різні типи нейронних мереж. Нижче наведені декілька типів нейронних мереж, які часто застосовуються в галузі кібербезпеки:

- Рекурентні нейронні мережі (RNN). RNN підходять для аналізу послідовних даних і можуть використовуватися для моделювання тимчасових залежностей потоку мережного трафіку або послідовності подій. Вони можуть бути ефективними при виявленні аномалій у динаміці поведінки системи [10].
- Згорткові нейронні мережі (CNN). CNN зазвичай застосовуються для аналізу зображень, але вони також можуть бути використані для виявлення аномалій

даних, наприклад, в мережевому трафіку. Застосування фільтрів і шарів згортки допоможе виявити шаблони, які можуть бути пов'язані з кібератаками [10].

- Глибокі автоенкодери (Deep Autoencoders). Автоенкодери можуть використовуватися для виявлення аномалій у даних. Глибокі автоенкодери, включаючи варіанти, такі як варіаційні автоенкодери (VAE), дозволяють вивчати складні нелінійні залежності та виявляти відхилення від звичайного патерну.
- Генеративні змагальні мережі (GAN). GAN можуть використовуватися для створення моделей, які генерують "нормальні" дані, а потім відхилення від цих даних можуть вважатися потенційними аномаліями. Це дозволяє моделі більш гнучко адаптуватися до характеристик кіберзагроз, що змінюються.
- Логічні нейронні мережі (Logical Neural Networks). Цей тип нейронних мереж розробляється спеціально для виявлення аномалій в мережевому трафіку. Вони намагаються аналізувати логічні залежності між подіями та виявляти аномальні комбінації подій.
- Мережі довгої короткострокової пам'яті (LSTM). LSTMs, підтип рекурентних нейронних мереж, добре підходять для аналізу послідовних даних із довгостроковими залежностями. Їх можна використовуватиме виявлення аномалій у поведінці системи.

Часто для конкретної задачі виявлення кібератак використовується комбінація різних типів нейронних мереж або їх модифікацій для досягнення найкращих результатів. Точний вибір залежить від характеру даних, типу кібератаки, яку потрібно виявити, та інших факторів.

Нижче наведено кілька кроків, які можна виконати, щоб запровадити систему виявлення кібератак за допомогою штучних нейронних мереж.

- Збір даних. Необхідно збирати дані про нормальну роботу системи, щоб навчити нейронну мережу розпізнавати нормальну поведінку. Ці дані можуть

містити інформацію про мережевий трафік, журнали подій та інші параметри ІКТ.

- Вибір архітектури нейронної мережі. Для обробки різних типів даних можна використовувати рекурентні нейронні мережі (RNN), згорткові нейронні мережі (CNN) або комбінацію цих типів мереж.
- Навчання моделі. Зібрані дані для навчання нейронної мережі. Навчіть модель розпізнавати звичайну поведінку системи та виявляти відхилення, які можуть свідчити про кібератаку.
- Визначення ознак кібератак. Визначте характерні ознаки кібератак, які можуть бути виявлені за допомогою нейронної мережі. Це можуть бути аномалії у мережевому трафіку, незвичайні запити до системи, зміни у зразку використання ресурсів тощо.
- Інтеграція із системою моніторингу. Інтегруйте розроблену модель із системою моніторингу ІКТ для безперервного аналізу даних у реальному часі.
- Налаштування параметрів та оновлення моделі. Регулярно налаштовуйте параметри моделі та оновлюйте її, враховуючи нові види кібератак та зміни в обстановці.
- Керування помилковими спрацьовуваннями. Передбачте механізми для зниження кількості помилкових спрацьовувань, таких як аналіз та фільтрація результатів.
- Моніторинг результатів. Постійний моніторинг ефективності системи виявлення та вносити корективи у разі потреби.

Ефективність системи виявлення кібератак за допомогою штучних нейронних мереж залежить саме від якості даних, правильного вибору та конфігурації моделі та здатності системи швидко адаптуватися до нових загроз [10].

Щоб вирішити такі проблеми, як кластерний аналіз, визначення стану об’єктів керування, класифікація їхніх дефектів, прогнозування розвитку дефектів під час роботи та ідентифікація шаблонів дефектів, були вивчені архітектури

нейронних мереж, такі як Кохонен та інші рівневі персептрони, мережа радіальних базисних функцій, гібридна нейронна мережа, мережа теорії адаптивного резонансу. Перераховані архітектури нейронних мереж відрізняються алгоритмами роботи та навчання, що визначає коло їх завдань.

Нейронна мережа Кохонена призначена для вирішення задач кластерного аналізу та об'єднання множин діагностичних ознак у групи, формування класів дефектів об'єкту контролю, визначення його стану тощо [12].

Мережа складається з деякої кількості *M* адаптивних лінійних суматорів, що діють паралельно (лінійних формальних нейронів). Всі вони мають однакову кількість входів *N* і отримують на свої входи один і той же вектор вхідних сигналів $x = (x1, \ldots, xN)$.

Дані, що подаються на входи нейронної мережі Кохонена представляються у вигляді вектора діагностичних ознак в *N*-вимірному евклідовому просторі, а також мають бути правильно промасштабовані для подальшої їх обробки. В процесі функціонування нейрони мережі Кохонена визначають функцію відстані $\rho k(X, Wk)$ між вхідним вектором та власними центрами, де *X* — вхідний вектор, *Wk* — вагові коефіцієнти нейрона *k*.

На підставі отриманого значення функції відстані мережею приймається рішення про приналежність даного вхідного вектору до певної групи (кластеру). Структуру мережі Кохонена зображено на рисунку 1.3.

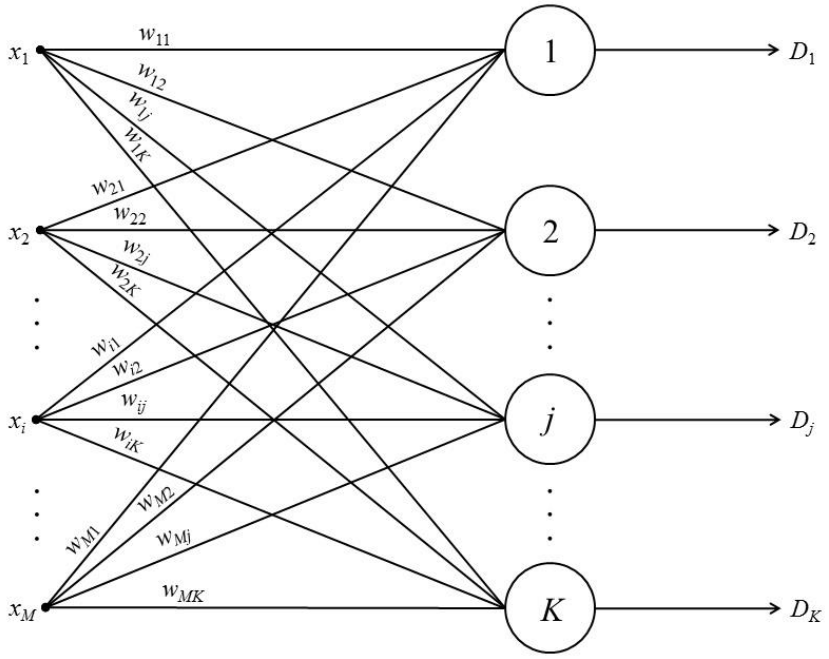


Рисунок 1.3 – Структура мережі Кохонена [4]

Нейронні мережі Кохонена працюють, відображаючи набір високовимірних вхідних векторів на карту менших вимірних кластерів, і таким чином тісні кластери на карті відповідають кластерам у початковому просторі. Отже, в результаті функції (самоорганізації) на виході мережі Кохонена утворюються кластери (група активних нейронів певної розмірності), які характеризують певні категорії вхідних векторів, що відповідають одній і тій же вхідній ситуації (наприклад, об’єкти контролю, наприклад) можливі дефекти певного типу). Навчання такої мережі відбувається за алгоритмом навчання без учителя одним із наступних правил конкуруючого навчання: WTA (winner takes all), CWTA (conscience WTA), WTM (winner takes most) тощо. Алгоритм роботи та формування класів дефектів об’єкту контролю за допомогою мережі Кохонена можна представити наступним чином:

1) ініціалізувати вагові коефіцієнти випадковими значеннями. Задати величину швидкості ε та часу навчання tmax;
2) подати значення вхідних сигналів $X = (x1, ..., xp)$ на вхід мережі;
3) визначити відстань ρk від вхідного сигналу X до кожного нейрона k мережі;
4) знайти нейрон-переможець, тобто знайти нейрон k, для якого відстань ρk є найменшою;

5) адаптувати вагові коефіцієнти нейрона-переможця;
6) оновити величину швидкості навчання *ε(t),* якщо цього передбачає алгоритм навчання;
7) якщо *(t < tmax),* то перейти до пункту 2, якщо інше — СТОП.

Оскільки алгоритм неконтрольованого навчання використовується для налаштування параметрів нейронної мережі Кохонена, мережу можна використовувати для розв'язування діагностичних проблем без тестування, оскільки вона не вимагає початкової інформації про об'єкт керування. Однак у випадку складних лінійно-нероздільних просторів даних достовірність формування класів нейронною мережею Кохонена не перевищує показника 0,88…0,93 [13].

Іншим поширеним типом нейронної мережі, який можна використовувати, є багатошаровий персептрон. Завдання, що вирішуються цим типом нейронної мережі, включають визначення стану об'єкта керування, класифікацію можливих дефектів, прогнозування розвитку дефектів у часі, виявлення шаблонів дефектів, інтерполяцію даних (визначення характеру залежностей між вхідними даними та станами), мету керування.

Багатошаровий персептрон є мережею з *L* шарами нейронів і *Jl* нейронами на кожному шарі, l - номер шару (l ∈ l ... L) і входами нейронів першого шару. Нейрони кожного шару з'єднуються з нейронами попереднього і подальшого шарів за принципом «кожен з кожним».

Кожен шар виконує нелінійне перетворення на основі лінійної комбінації вихідних сигналів попереднього шару. Таким чином, багатошаровий персептрон формує довільну багатовимірну функцію на вихідному кінці шляхом відповідного вибору кількості шарів, діапазону зміни сигналу та параметрів нейрона. Завдяки поперемінному обчисленню лінійних комбінацій і нелінійних перетворень можна досягти апроксимації будь-якої багатовимірної функції шляхом відповідного вибору параметрів мережі. Вихідний сигнал нейронів першого шару обмежує початкову область сигналу наближеного розпізнавання і надходить у мережу другого рівня. Нейрони другого шару додають ще одну

площину, що розділяє інформаційний простір. Нормаль даної площини є лінійною комбінацією нормалей першого шару нейронів. Таким чином, кожен нейрон другого шару виділяє фрагмент інформаційного простору. Потім сигнал досягає третього рівня, де простір діагностичних ознак розбивається більш детально. За цією схемою фрагментація інформаційного простору відбувається рівномірно. Вихідний нейрон об'єднує просторові сегменти, вибрані на попередньому етапі. У результаті нейронна мережа формує набір площин, які розділяють інформаційний простір, і здатна виокремлювати області простору зі складною конфігурацією, залежно від взаємного розташування та граничного порядку площин поділу. Запропонована мережа виконує нелінійне розділення та класифікацію об’єктів на основі набору діагностичних ознак, які часто мають великий розмір. Багатошарові класифікатори на основі персептронів є загальним методом апроксимації функції, що дозволяє використовувати його для вирішення завдань класифікації різної складності. На відміну від мережі Кохонена, багатошаровий персептрон навчається вчителем, тобто для нього потрібна навчальна вибірка, яка містить набір можливих вхідних сигналів (векторів даних) і відповідних їм мережевих вихідних сигналів. Тому його не можна використовувати для безстандартної діагностики контрольних суб’єктів. Але індекс надійності класифікації багатошарового персептрона значно перевищує показник мережі Кохонена, дозволяючи класифікувати об’єкти, навіть якщо лінійно не класифікуються, а також дозволяє використовувати вектори діагностичних ознак великої розмірності та класифікувати сигнали на тлі перешкод.[14] Для навчання багатошаровий персептрон використовуються градієнтні методи, що враховують помилку нейронів кожного шару та виконують корекцію вагових коефіцієнтів нейронів в залежності від їхньої помилки. В процесі навчання мережа змінює свої параметри і вчиться давати потрібне відображення множини вхідних векторів X у множину потрібних значень виходів мережі Y. За рахунок здібності до узагальнення мережею можуть бути отримані правильні результати, якщо подати на вхід вектор, який не зустрічався при навчанні. При вирішенні більшості задач для навчання багатошаровий

персептрон використовується метод зворотного розповсюдження помилки (back propagation error, згідно з яким наступний крок направлений в сторону антиградієнта функції помилки. Відповідно до цього методу функція помилки E представляється у вигляді складної функції і послідовно розраховуються частинні похідні за формулою для складної функції [15].

Проведений аналіз показує, що нейронна мережа Кохонена та багатошаровий персептрон не можуть бути використані окремо для діагностики, оскільки багатошаровий персептрон обов'язково потребує початкових навчальних зразків (попередньої інформації про об'єкти контролю та їх можливу номенклатуру) дефектів (досить складна у випадку композитних матеріалів), і мережа Кохонена не може надійно класифікувати дефекти за наявності лінійно нероздільного простору діагностичних ознак. У цьому випадку може бути використана спеціально створена гібридна нейронна мережа, що складається з шару Кохонена та багатошарового персептрона або радіальної базисної мережі.

Подібні архітектури гібридних нейронних мереж можуть ідентифікувати та класифікувати дефекти в композитних продуктах з високою надійністю, будувати нелінійні гіперплощини поділу, кластерний аналіз і неконтрольоване навчання.

Саме гібридна мережа має всі переваги її складових (мережі Кохонена та багатошарового персептрону), а саме можливість виконувати кластерний аналіз даних, високу достовірність контролю, здатність будувати складні нелінійні розділяючі гіперплощини, а також визначати нові об'єкти та розширювати власну базу знань (базу запам'ятованих класів). Гібридну нейронну мережу Кохонена та БП зображено на рисунку 1.4 [15].

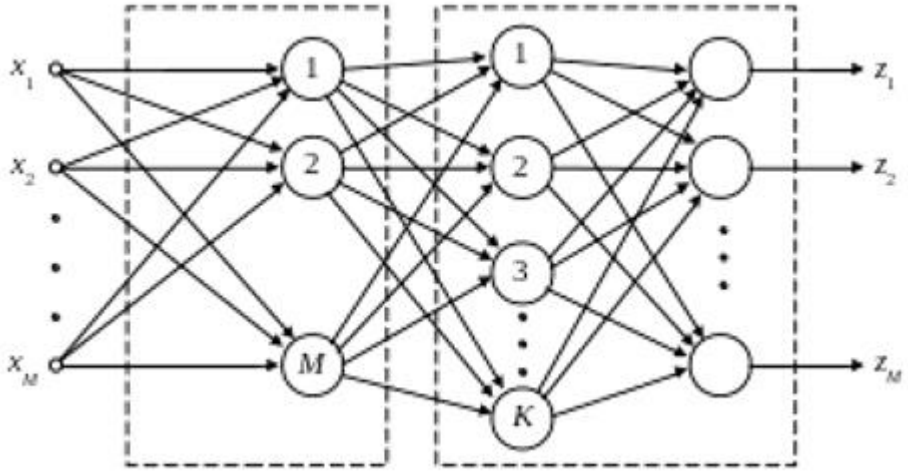


Рисунок 1.4 – Гібридна нейронна мережа Кохонена та БП [15]

Однак ця мережа має певні недоліки: вона потребує більше часу на навчання, якщо з'являються нові об'єкти, і додатковий час на перенавчання (що відбувається у випадку формування нових класів у процесі контролю (класів), це повне перенавчання, де інформація, отримана раніше, втрачається. Крім того, для роботи цієї нейронної мережі необхідний основний вибір діагностичних характеристик (амплітуда, тривалість, частота, фаза, спектральна щільність імпульсів тощо), які використовуються для контролю та формування діагностичного рішення.

1.4 Висновки до розділу

Завершуючи розділ, присвячений аналізу моделей і методів виявлення кібератак на основі алгоритмів штучного інтелекту, ми робимо висновок, що структури в усьому світі перебувають на переломному етапі через використання більш сучасного обладнання та програмного забезпечення для інформаційної протидії. У контексті інформаційної безпеки штучний інтелект – це програмне забезпечення, здатне інтерпретувати стан навколишнього середовища, ідентифікувати певні події та самостійно вживати необхідних заходів. Технології ШІ ефективні в розшифровці шаблонів і аномалій, тому вони можуть стати інструментами моніторингу загроз. Надійна стратегія інформаційної безпеки також допомагає захистити особисті дані громадськості, а також державні дані та

алгоритми, що стає ще важливішим із розгортанням нових моделей штучного інтелекту.

Розглянуто найвідоміші архітектури та типи нейронних мереж, а саме багатошарові персептрони, мережі Кохонена, гібридні нейронні мережі тощо, а також можливості їх застосування в системах. Описано їх структуру, алгоритми роботи та навчання. Нейронні мережі мають особливі властивості, такі як самоорганізація, здатність до навчання на роботі, узагальнення, моделювання процесів і явищ (в тому числі нелінійних), формування складних залежностей у просторі діагностичних ознак і поза ним. класу, ефективність, з якою обробляються великорозмірні елементи, визначає зручність їх використання для вирішення діагностичних задач, особливо з композиційними матеріалами.

2 РОЗРОБКА УДОСКОНАЛЕНОГО МЕТОДУ ВИЯВЛЕННЯ КІБЕРАТАК НА ОСНОВІ МОДЕЛЕЙ НЕЙРОННИХ МЕРЕЖ ТА ШТУЧНОГО ІНТЕЛЕКТУ

2.1 Обґрунтування вибору методу виявлення кібератак на базі алгоритмів штучного інтелекту

На сьогоднішній день відомо багато різних методів виявлення кібератак. Перевагами являється саме застосування методів машинного навчання в порівнянні з сигнатурним аналізом. Вища точність виявлення та менша кількість хибних спрацювань, також можливість виявлення аномалій та нових різних ознак атак. Але ці методи мають і певні недоліки.

Серед них саме необхідність в додаткових апаратних ресурсах та більш низька швидкість обробки даних. В розділі представлено огляд сучасних методів, спрямованих на виявлення кібератак та аномалій в інформаційно-комунікаційних мережах із застосуванням методів машинного навчання.

Основними недоліками відомих методів є неспроможність виявлення та адаптивного реагування на атаки. Недолік є найбільш критичним, саме про це свідчить постійне зростання кількості кібератак.

Обмеженням для більшості відомих підходів є потреба в значних обсягах обчислювальних ресурсів та значний час відгуку систем виявлення кібератак.

Виявлення кібератак є важливою складовою кібербезпеки. Існує кілька методів і інструментів для виявлення кібератак [16].

Такі як системи виявлення вторгнень IDS (Intrusion Detection System) та IPS (Intrusion Prevention System) використовуються для виявлення та запобігання неправомірним спробам доступу до комп'ютерної системи чи мережі. IDS аналізує мережевий трафік та сповіщає про можливі атаки, тоді як IPS може автоматично вживати заходи для блокування атак [16]. Модель з використанням методу системи виявлення вторгнень IDS зображена на рисунку 2.1.

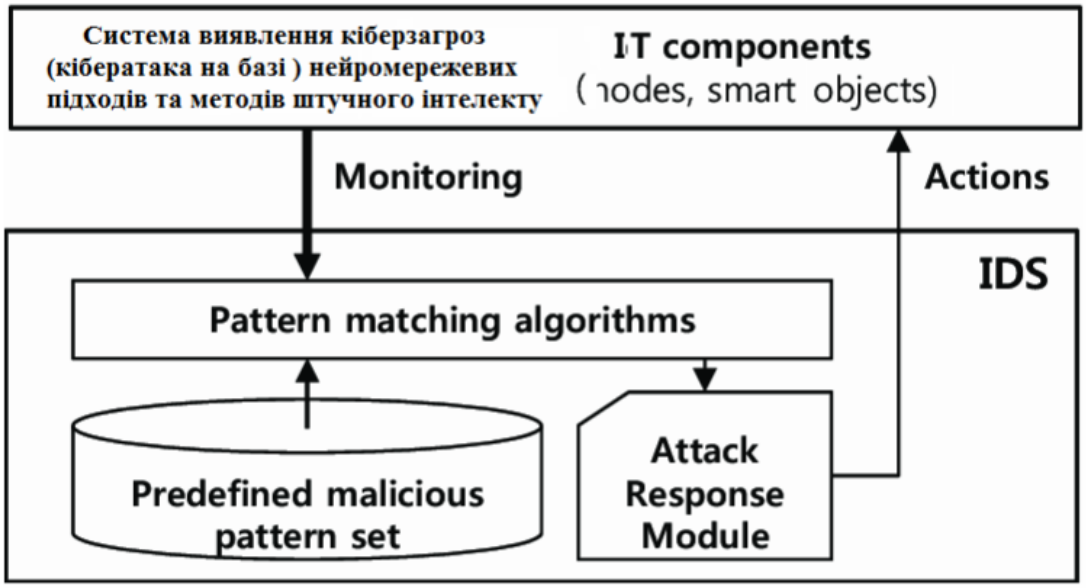


Рисунок 2.1 – Модель з використанням методу системи виявлення вторгнень IDS

Також системи логування та аудиту, які записують подій та аудит системних ресурсів може допомагати виявити аномальні дії або неправомірний доступ до системи [4]. Щодо аналізу вразливостей, то системи можуть використовувати інструменти для пошуку вразливостей у програмному забезпеченні та виправлення їх до того, як їх можуть використовувати зловмисники.

Системи аналізу аномалій, аналізуючи звичайні патерни поведінки системи чи користувачів, системи аналізу аномалій можуть виявити неправомірні дії.

Моніторинг мережі - системи моніторингу мережі дозволяють виявляти незвичайний трафік або аномалії в мережі, що можуть бути показниками кібератак.

Інтелектуальні системи та машинне навчання застосування інтелектуальних систем, таких як системи машинного навчання, може допомогти виявляти нові, раніше невідомі види загроз на основі аналізу великих обсягів даних [6].

Хонепоти (Honeypots) та хоненети (Honeynets) – це система, яка призначена для симуляції служби чи ресурсу, який може бути цільовим для атаки. Вона слугує приманкою для зловмисників. Хоненет – це мережа злитих хонепотів, що дозволяє спостерігати за атаками в реальному часі.

Ці методи та інструменти часто використовуються разом для максимально ефективного виявлення кібератак та забезпечення безпеки інформаційних систем.

2.2 Удосконалення методу виявлення кібератак на базі архітектури нейронних мереж із алгоритмами штучного інтелекту

Удосконалення методу виявлення кібератак нейронної мережі включає кілька ключових етапів, які варто враховувати під час проектування та вибору оптимальної моделі. Етапи реалізації методу нейронної виявлення кібератак мережі зображено на рисунку 2.2.

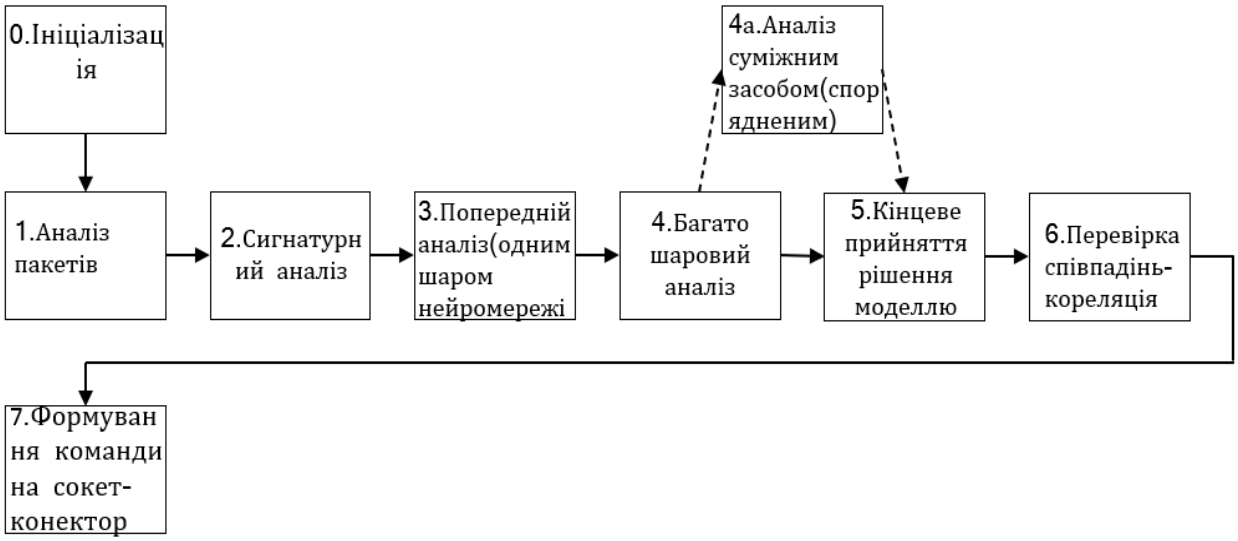


Рисунок 2.2 – Етапи реалізації методу

Ініціалізація в нейронних мережах — це процес присвоєння початкових значень вагам нейронів під час створення або перезапуску нейронної мережі перед початком навчання. Ефективна ініціалізація ваг є важливою, оскільки вона може впливати на процес навчання та результати моделі.

Мережевий аналіз пакетів - це процес вивчення та інтерпретації мережевого трафіку, щоб здобути різноманітну інформацію, таку як стан мережі, аномалії, безпекові проблеми та ефективність мережі. Класична схема організації мережі зображена на рисунку 2.3 [13].

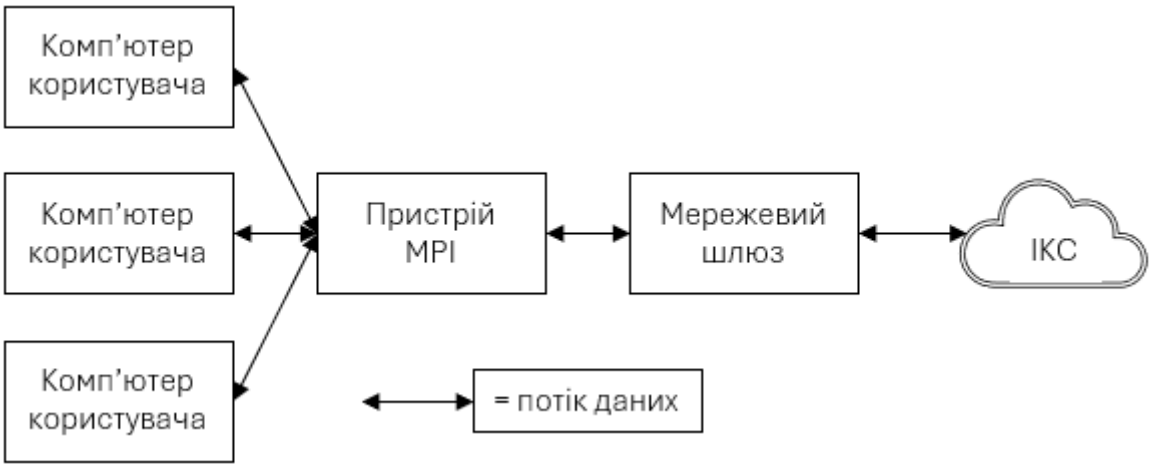


Рисунок 2.3 – Класична схема організації мережі

Для проведення мережевого аналізу пакетів використовуються спеціальні інструменти, які дозволяють перехоплювати, аналізувати та відображати пакети даних, що пересилаються по мережі.

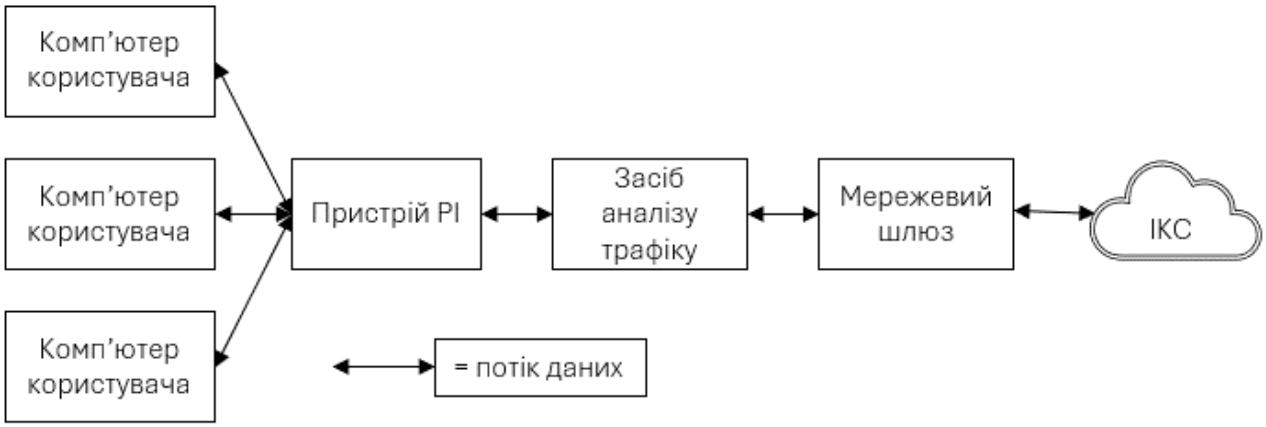


Рисунок 2.4 – Схема організації мережі на базі AI моделі аналізу трафіку в інформаційно-комунікаційних системах

Сигнатурний аналіз - це метод виявлення аномалій в даних на основі сигнатур, тобто унікальних властивостей або позначень, що характеризують певний тип подій чи аномалій. Він має свої переваги, такі як ефективність у виявленні відомих атак, але він може бути обмежений у виявленні нових атак або аномалій, які не входять в існуючі сигнатури [8]. Також існує ризик використання фальшивих позитивів або обхід сигнатур шляхом кількох змін у атаках. Сучасні системи

безпеки зазвичай комбінують сигнатурний аналіз з іншими методами, такими як аномальний аналіз та машинне навчання, для більш ефективного виявлення загроз.

Аналіз одного шару нейромережі включає в себе дослідження структури та властивостей конкретного шару в нейронній мережі. Нейронні мережі складаються з різних типів шарів, таких як вхідний шар, приховані шари і вихідний шар. Кожен шар виконує свою функцію в обчисленнях та вивченні залежностей у наборі даних. Багатошарова модель забезпечує високий рівень гнучкості, масштабованості, надійності, захищеності та еластичності [2].

Багатошаровий аналіз нейронної мережі включає вивчення кожного шару і їх взаємодії в контексті вирішення конкретної задачі. Багатошарова нейронна мережа складається з трьох основних типів шарів: вхідного шару, прихованих шарів і вихідного шару. Вивчення кожного типу шару допомагає зрозуміти, як мережа здійснює свою роботу та як вона може бути оптимізована [4].

Кінцеве прийняття рішення моделлю нейромережі включає в себе ряд етапів, де модель видає прогнози чи класифікації, і ці результати використовуються для прийняття рішення. Загальний опис цього процесу:

- Введення даних. Починається з подачі вхідних даних моделі. Це може бути новий вхідний об'єкт, для якого потрібно зробити прогноз, або новий набір даних.
- Прогнозування моделлю. Модель нейромережі оброблює вхідні дані і генерує прогнози чи класифікації відповідно до завдання.
- Вивчення прогнозів. Аналіз результатів прогнозів, їхній інтерпретації та вивчення впливу вхідних факторів на вихідні результати.
- Прийняття рішення. На основі прогнозів моделі та контексту завдання видається остаточне рішення. Це може включати в себе прийняття конкретного рішення або визначення ймовірності вірності прогнозів загрози.
- Додатковий аналіз. Залежно від конкретного випадку може виконуватися додатковий аналіз результатів, робота з вибірковими групами, порівняння з очікуваннями чи інші дії для вдосконалення процесу прийняття рішення.
- Застосування рішення. Прийняте рішення застосовується у практиці.

– Моніторинг та оцінка. Після впровадження рішення слід здійснювати моніторинг його ефективності та оцінку результатів. Це може включати аналіз того, наскільки добре рішення відповідає на практиці та, за потреби, коригування методу або моделі.

2.3 Загальна структура шарів нейронної мережі засобу виявлення кібератак

Шари в нейромережі – це групи нейронів, які працюють разом і відповідають за різні етапи обробки інформації. Вони забезпечують нейромережі можливість адаптуватися до різних задач. Шари з'єднані між собою, як сходи між поверхами. Кожен шар має свою роль у процесі обробки інформації. Є три основні типи шарів у нейромережі. Вхідний шар: Це "перший поверх" будинку, де ми починаємо. Вхідний шар приймає дані ззовні, наприклад, зображення або текст, і передає їх у наступні шари. Вхідний шар не змінює дані, а лише служить точкою входу для них. Приховані шари: Це "середні поверхи" будинку. Приховані шари забезпечують обробку вхідних даних і передачу інформації між шарами. Вони називаються "прихованими", оскільки їхні результати не відображаються напряму на виході мережі. Кількість прихованих шарів та нейронів у них може варіюватися в залежності від складності задачі та архітектури мережі. Вихідний шар: Це "останній поверх" будинку. Вихідний шар формує результат, який нейромережа передбачає на основі вхідних даних [5].

Результат може бути класифікацією, числовим значенням або іншою інформацією, залежно від типу задачі. Усі нейрони в шарах з'єднані між собою через ваги зв'язків. Ваги відіграють важливу роль у навчанні нейромережі, оскільки вони визначають силу впливу одного нейрона на інший. В процесі навчання, ваги оптимізуються, щоб мінімізувати помилку передбачення мережі. Зміщення допомагає нейромережі легше адаптуватися до різних даних та виконувати більш гнучкі перетворення на вхідних даних. Ще одним ключовим компонентом нейромережі є функція активації. Вона застосовується до кожного нейрона у

прихованих та вихідних шарах, щоб визначити його активність на основі суми вхідних сигналів, помножених на відповідні ваги та додавання зміщення. Функція активації може бути лінійною або не лінійною, в залежності від типу задачі та архітектури мережі.

Загальна структура нейронної мережі включає в себе різні шари, що працюють разом для вирішення конкретної задачі. Основні компоненти включають вхідний шар, приховані шари і вихідний шар. На рисунку 2.5 зображено моделювання структури шарів і параметрів розробленої моделі.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 66)]              0

 hl1 (Dense)                 (None, 1000)              67000

 hl2 (Dense)                 (None, 250)               250250

 hl3 (Dense)                 (None, 20)                5020

 score (Dense)               (None, 1)                 21

=================================================================
Total params: 322291 (1.23 MB)
Trainable params: 322291 (1.23 MB)
Non-trainable params: 0 (0.00 Byte)
```

Рисунок 2.5 – Моделювання структури шарів і параметрів розробленої моделі

Перший шар *input_1* отримує дані датасету та перетворює їх в тензор, його розмірність (*None, 66*), перше число (в даному випадку *None*) визначає розмір порції даних для тренування, *None* визначає що вона не вказана, тому модель приймає порцію будь-якого розміру, друге число(в даному випадку 66) визначає кількість характеристик об’єкту, за якими вже будується мережа зв’язків з наступним шаром.

Dense-шар (*hl1*) перший прихований шар нейронної мережі на 1000 нейронів, що з’єднаний з кожною характеристикою об’єкту.

hl2 і *hl3* також *Dense*-шари, що зменшують кількість нейронів для узагальнення шаблонів вхідних даних.

Dense шар *score*, з лінійною активацією, підраховує загальну оцінку аномальності, чим більша оцінка тим більше окремий об’єкт відрізняється від норми.

Отже, базова структура та компоненти нейромережі включають вхідний, прихований та вихідний шари, нейрони з вагами зв’язків та зміщеннями, а також функції активації. Разом вони сприяють адаптації нейромережі до вхідних даних та вирішенню складних задач. Зрозуміння цих компонентів та їх взаємодії допоможе новачкам краще розібратися в основах нейромереж та їх застосуваннях.

2.4 Загальна структура розроблених моделі та методу

Більшість традиційних методів виявлення аномалій, наприклад, метод, що базується на відстані, та метод, що базується на щільності, є неефективними у вирішенні проблеми нерелевантних ознак чи нелінійно роздільних класів через розмірності та недоліки у виявленні нелінійних зв'язків.

На даний час, ансамблеві методи (наприклад, iForest та багато інших) показали значне покращення порівняно з цими підходами, працюючи з обраними підпросторами ознак, але ефективний спосіб ідентифікації відповідних підпросторів та моделювання складних взаємозв'язків залишається відкритою проблемою виявлення аномалій.

Для вирішення цієї проблеми створена нова структура, що поєднує нейромережі, апріорний розподіл ймовірностей аномальних оцінок та нову функцію втрат для навчання глибокого детектора аномалій з метою призначення статистично значущих більших оцінок аномалій, ніж звичайним об'єктам.

Очікується, що отримана модель буде давати оцінки аномалій та буде більш ефективною за обчислювальною потужністю, ніж двоетапний підхід.

Спочатку використовується мережа оцінювання аномалій, зліва, для отримання скалярної оцінки аномалії для кожного вхідного об’єкта.

Для керування навчанням оцінок аномалій ми використовуємо генератор опорної ймовірності, з якої буде отримана середня ймовірність.

Для випадково вибраних звичайних об'єктів, опорна ймовірність може бути як вивчена моделлю, так і визначена апріорною ймовірністю рисунок 1, праворуч.

Згодом, стандартне відхилення апріорної ймовірності вводиться в контрастову функцію втрат для керування оптимізацією, в якій здійснюється оптимізація оцінок аномалій так, щоб оцінки аномалій сильно відхилялися від середньої оцінки нормальних об’єктів, при цьому оцінки звичайних об'єктів були якомога ближче до середнього по апріорній функції.

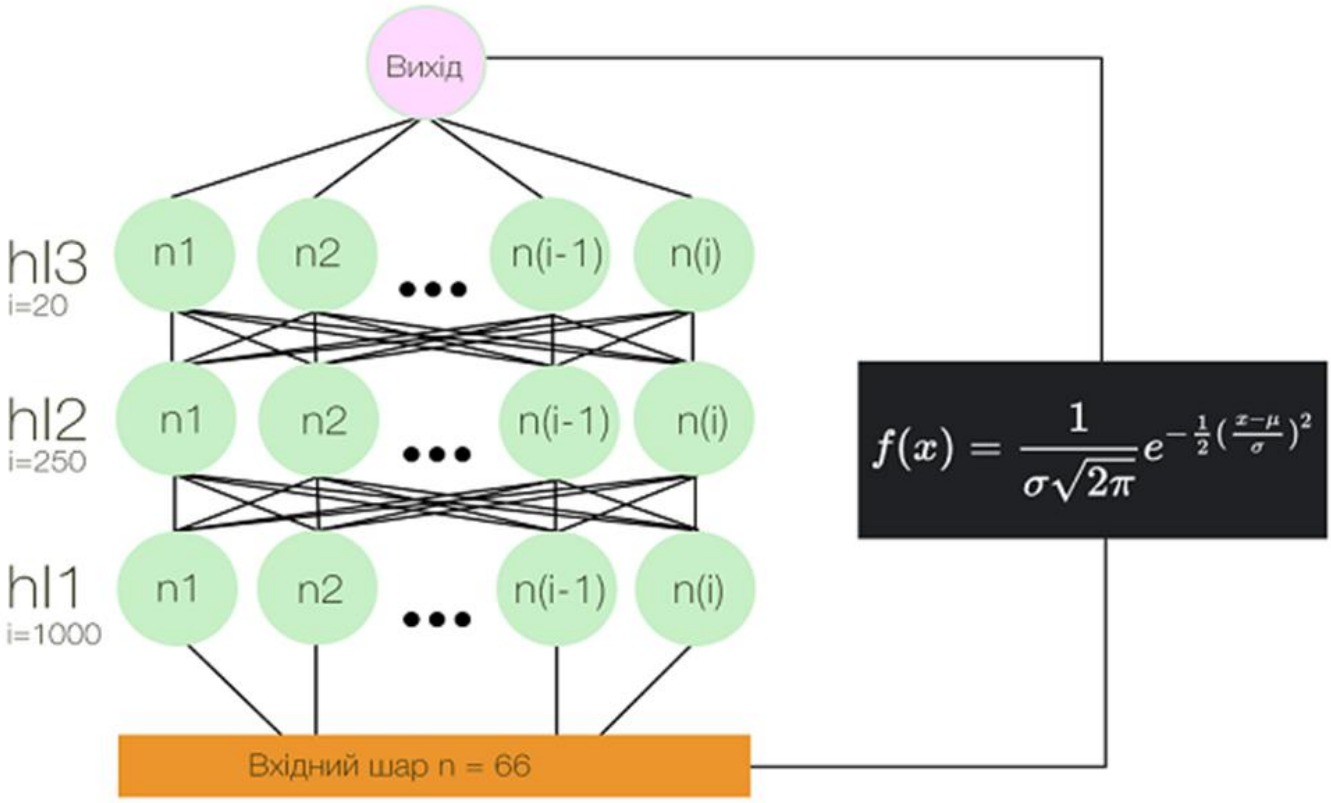


Рисунок 2.6 – Загальна структура моделі нейромережі для задач виявлення кібератаки

Формула для ймовірнісної щільності нормального розподілу (також відомого як гаусівський розподіл) має вигляд:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}, \quad (2.1)$$

де *f(x|μ,σ)* - ймовірнісна щільність розподілу для значення *x* при середньому μ та стандартному відхилень *σ*.

μ - середнє (середнє значення) розподілу.

σ - стандартне відхилення розподілу.

Ця формула представляє собою колоколоподібну криву, де середнє значення відповідає піку, а стандартне відхилення контролює ширину кривої. Формула дозволяє обчислити ймовірність того, що випадкова змінна, взята з нормального розподілу, прийме конкретне значення.

Розрахунок інформаційної надійності і стійкості визначення кіберзагроз програмними засобами штучного інтелекту і нейромереж в процесі оброблення-передавання інформаційних пакетв даних в інфраструктурі інформаційної мережі.

Розраховуємо середній час стійкого визначення кіберзагроз (робота із стійким визначенням ознак кібезагроз) в умовах підтримки повного функціоналу:

$$T_{CS} = 1/loss_{hings}(x, y, \omega),\ T_{CS} = \frac{1}{loss_{hings}(x,y,\omega)}\ . \qquad (2.2)$$

Визначаємо коефіцієнт визначення кіберзагрози/кібератаки моделлю:

$$H_{\lambda i} = \mu/(T_{cs}+\lambda_i), \qquad (2.3)$$

де μ - інтенсивність надходження запитів інформаційних птотоків у програмних компонентів/модулів ІС: $\mu = 0..1,2*10^{-1}$; λ_i – середня інтенсивність надходження запитів самих кіберзагроз в інформаційних системах (ІС). Яка формується моделлю сумарна інтенсивність складає:

$$H_S=SUM(H_{\lambda i})_{i=1..N}. \qquad (2.4)$$

Для захисту від інформаційних втручань та впливу, кібератак і кіберзагроз в ІС автоматизованих інформаційних систем, а також у ІМ і пристроях в контексті

сучасних підходів інформаційних систем, все більш актуальними стають комплексні підходи захисту, в із застосуванням нейромереж та систем штучного інтелекту і захистом на базі апаратного підключення до ІС і втручання в роботу інформаційних процесів і ПЗ. Основні зусилля направляються на захист інформаційних систем старт інтелектуальними методами штучного інтелекту. Що дозволяють точно я якісно виділяти і класифікувати сучасні атаки і кіберзагрози та ефективно їм протидіяти.

2.5 Висновки до розділу

У даному розділі було розглянуто різні етапи реалізації архітектури нейронної мережі, а саме: ініціалізацію, мережевий аналіз пакетів, сигнатурний аналіз, аналіз одним шаром нейромережі, багатошаровий аналіз нейромережі, кінцеве прийняття рішення моделлю нейромережі та інші.Також загальну структуру шарів нейронної мережі, моделі та методів. Ефективний спосіб ідентифікації відповідних підпросторів та моделювання складних взаємозв'язків залишається відкритою проблемою виявлення аномалій. Для вирішення цієї проблеми створена нова структура, що поєднує нейромережі, апріорний розподіл ймовірностей аномальних оцінок та нову функцію втрат для навчання глибокого детектора аномалій з метою призначення статистично значущих більших оцінок аномалій, ніж звичайним об'єктам.

Очікується, що отримана модель буде давати оцінки аномалій та буде більш ефективною за обчислювальною потужністю, ніж двоетапний підхід.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛІ ТА МЕТОДУ АНАЛІЗАТОРА ТРАФІКУ В ІКС

3.1 Формулювання вимог до програмного засобу

Метою є розробка моделі і методу виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту.

Розроблений програмний засіб повинен бути у вигляді модулю, що може бути підключений до іншої системи, наприклад веб-застосунку.

Додаток повинен відповідати наступним вимогам:

- Наявність можливості підключення до веб-застосунку
- Можливість перевірки запитів з метою виявлення потенційно небезпечних запитів
- Наявність можливості роботи з інформацією у .json, а також інших форматах
- Розробка супроводжувальної документації
- Застосування алгоритмів машинного навчання

Обґрунтування засобів для реалізації

Python - це популярна мова програмування, яка має загальне призначення, що може бути використане для найрізноманітніших програм. Вона включає динамічне введення тексту та прив'язування, високорівневі структури даних та багато інших функцій, які роблять його настільки ж корисним для складної розробки додатків, як і для створення сценаріїв. Це універсальна мова, що зустрічається в безлічі різних застосувань.

Python є однією з найпопулярніших мов програмування для машинного навчання завдяки своїй простоті, великому співтовариству розробників та багатству бібліотек для машинного навчання. Ось кілька ключових бібліотек та інструментів Python для машинного навчання:

NumPy бібліотека для роботи з великими масивами та матрицями чисел. Вона надає базові функції для виконання операцій лінійної алгебри, що є важливим для багатьох алгоритмів машинного навчання.

Pandas бібліотека для обробки та аналізу даних. Вона дозволяє легко завантажувати дані, виконувати операції з фільтрацією та групуванням, а також готує дані для використання в моделях машинного навчання.

Scikit-Learn це високорівнева бібліотека для машинного навчання, яка містить реалізації багатьох алгоритмів класифікації, регресії, кластеризації та інших. Scikit-Learn надає простий та єдиночний інтерфейс для роботи з різними моделями.

TensorFlow та PyTorch ці бібліотеки надають інструменти для реалізації та навчання нейронних мереж. Обидві бібліотеки є потужними та використовуються в глибокому навчанні.

Keras високорівневий інтерфейс для роботи з TensorFlow та іншими бібліотеками глибокого навчання. Keras спрощує процес побудови та тренування нейронних мереж [18].

Основними перевагами Python є:

- простий і зрозумілий синтаксис;
- відсутність дужок;
- автоматичний розподіл пам'яті;
- динамічний набір тексту;
- велика підтримка в інтернеті;
- багато підтримуваних бібліотек.

Можливі мінуси Python:

- повільний і може стати громіздким для великих і складних програм;
- мова високого рівня, яка не підходить для написання програмних програм;
- для деяких завдань неявне виділення пам’яті може бути недоліком.

Серед таких середовищ, як Atom, PyCharm, Sublime Text, Visual Studio Code, в межах робити обрано PyCharm. Він є безкоштовним, має режим налагодження коду. Підтримує мову програмування Python, автоматично виділяє синтаксичні конструкції, має підказки та довідку.

3.3 Підготовка даних для навчання моделі

Підготовка даних є дуже важливим етапом процесу машинного навчання. Якщо коротко, то підготовка даних - це набір процедур, що допомагає зробити масив даних більш придатним для машинного навчання. У ширшому сенсі під підготовкою даних також розуміють створення відповідного механізму збору даних. І ці процедури споживають переважну більшість часу, що витрачається на машинне навчання. Завдяки бібліотекам і інструментам, які були описані у розділі, де було обґрунтовано засоби для реалізації, ці інструменти дозволяють розробникам швидко та ефективно використовувати Python для вирішення завдань машинного навчання. На рисунку 3.1 зображено підключення бібліотек та інструментів.

```
import pandas as pd
import numpy as np
from keras.layers import Input, Dense
from sklearn.model_selection import train_test_split
from keras import regularizers, Model
from keras.optimizers import RMSprop
from keras import backend as K
import tensorflow as tf
import random
import sys
```

Рисунок 3.1 – Підключення бібліотек для навчання моделі

Для навчання моделі був обраний датасет з відкритого доступу Multi-Step Cyber-Attack Dataset (MSCAD) [19].

MSCAD.xlsx представляє розмічену версію набору даних. Шість файлів PCAP були оброблені за допомогою Wireshark. Під час обробки аналізувавши часові мітки мережевого трафіку (зловмисного та нормального), щоб позначити мережевий трафік [20]. Після обробки цих PCAP-файлів згенерований набір даних

(MSCAD) містить 77 характеристик (мережевих параметрів) з мітками. Фунцкію лейблингу до датафрейму зображенно на рисунку 3.2.

```
# Читаємо набір даних з csv-файлу та конвертації його у табличний об'єкт pandas.DataFrame
data = pd.read_csv("/kaggle/input/mscad/MSCAD.csv")

# Функція лейблингу датасету, Brute_Force у даному датасеті означає запит що був частиною DDOS або фішингової атаки
def remapping_func(el):
    if el == "Brute_Force":
        return 1
    else:
        return 0

# Застосовуємо функцію лейблингу до датафрейму
data['Label'] = data['Label'].map(remapping_func)

# Відокремлюємо лейбли від основної частини даних та конвертуємо у масив
# labels - масив лейблів, x - масив даних
labels = data['Label'].tolist()
labels = np.array(labels, dtype=np.float32)
x = data.drop('Label', axis=1)
x = x.to_numpy(dtype=np.float32)


MAX_INT = np.iinfo(np.int32).max
```

Рисунок 3.2 – Функція лейблингу до датафрейму

MSCAD включає два сценарії багатоетапних кібератак. Два сценарії багатокрокових атак були виконані наступним чином:

- Сценарій багатокрокових атак A: У цьому сценарії зловмисник намагається здійснити атаку перебором паролів (Brute force) на будь-який хост в мережі жертви.

- Багатокроковий сценарій атаки B: У сценарії зловмисник намагається виконати об'ємну DDoS-атаку на будь-який хост в мережі жертви [20].

3.4 Тренування нейронної мережі за допомогою фреймворку Tensorflow

TensorFlow — відкрита програмна бібліотека для машинного навчання цілій низці задач, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі для виявляння та розшифровування образів та кореляцій, аналогічно до навчання й розуміння, які застосовують люди [21]. Процес навчання моделі зображено на рисунку 3.3.

```
Epoch 15/200
4/4 [==============================] - 0s 9ms/step - loss: 2.1313
Epoch 16/200
4/4 [==============================] - 0s 11ms/step - loss: 1.9434
Epoch 17/200
4/4 [==============================] - 0s 9ms/step - loss: 1.7812
Epoch 18/200
4/4 [==============================] - 0s 9ms/step - loss: 1.6424
Epoch 19/200
4/4 [==============================] - 0s 9ms/step - loss: 1.5246
Epoch 20/200
4/4 [==============================] - 0s 9ms/step - loss: 1.4258
Epoch 21/200
4/4 [==============================] - 0s 10ms/step - loss: 1.3437
Epoch 22/200
4/4 [==============================] - 0s 10ms/step - loss: 1.2762
Epoch 23/200
4/4 [==============================] - 0s 8ms/step - loss: 1.2213
Epoch 24/200
```

Рисунок 3.3 – Результат процесу навчання моделі

На рисунку 3.3 вище зображено процес тренування моделі. Також зображено скільки епох тренування пройшло, скільки загалом, час на епоху, та найголовніше показник функції втрат під час тренування, що і визначає наскільки добре модель справляється з запам’ятовуванням та передбаченням результатів.

Демонстрація реалізації процесу тренування у виді зображень представлена нижче та є доказом, що ефективність цього процесу має позитивний результат на модель за обчислювальною потужністю, ніж двоетапний підхід.

```
def inject_noise(seed, n_out, random_seed):
    rng = np.random.RandomState(random_seed)
    n_sample, dim = seed.shape
    swap_ratio = 0.05
    n_swap_feat = int(swap_ratio * dim)
    noise = np.empty((n_out, dim))
    for i in np.arange(n_out):
        outlier_idx = rng.choice(n_sample, 2, replace=False)
        o1 = seed[outlier_idx[0]]
        o2 = seed[outlier_idx[1]]
        swap_feats = rng.choice(dim, n_swap_feat, replace=False)
        noise[i] = o1.copy()
        noise[i, swap_feats] = o2[swap_feats]
    return noise
```

Рисунок 3.4 – Функція створення шуму

На цьому етапі, який зображений на рисунку 3.4 вище представлено функцію створення шуму для додаткового розширення датасету та більш ефективного тренування. Функція обирає 5% записів, та заміняє їх на шум.

Для виконання у подальшому нескінченного генератору порцій для тренування, формуємо функції генерації порцій даних для тренування що містять і нормальні значення і аномалії.

Далі досліджуємо форму масиву вхідних даних, тобто кількість ознак у кожному записі та створюємо пустий масив бажаної форми (розмірність порції, кількість ознак запису). Наступним кроком є читання довжини масиву нормальних значень та довжина масиву аномальних значень, а також й цикл, що проходить по розміру порції. для кожного парного значення додаємо у вихідний масив нормальний запис вхідних даних, а для кожного непарного значення додаємо у вихідний масив аномальний запис вхідних даних. Останнім етапом повертаємо порцію значень та масив лейблів

Наступним кроком є створення нескінченного генератора опцій. Метод, яким було реалізовано зображено на рисунку 3.5 нижче.

```
def batch_generator_sup(x, outlier_indices, inlier_indices, batch_size, nb_batch, rng):
    rng = np.random.RandomState(rng.randint(MAX_INT, size=1))
    counter = 0
    while 1:
        ref, training_labels = input_batch_generation_sup(x, outlier_indices, inlier_indices, batch_size, rng)
        counter += 1
        yield (ref, training_labels)
        if (counter > nb_batch):
            counter = 0
```

Рисунок 3.5 – Створення нескінченного генератору порцій

Основна функція тренування моделі штучного інтелекту (ШІ) включає в себе навчання моделі на основі вхідних даних з метою вивчення зв'язків та побудови внутрішнього представлення для здійснення певного завдання. Нижче на рисунку 3.6 наведено основну функцію навчання:

```
def run_devnet(mode, epochs, batch_size, nb_batch):
    random_seed = 42

    outlier_indices = np.where(labels == 1)
    outliers = x[outlier_indices]
    x_train, x_test, y_train, y_test = train_test_split(x, labels, test_size=0.2, random_state=42, stratify=labels)
    outlier_indices = np.where(y_train == 1)[0]
    n_outliers = len(outlier_indices)
    n_noise = len(np.where(y_train == 0)[0]) * 0.02 / (1. - 0.02)
    n_noise = int(n_noise)
    rng = np.random.RandomState(random_seed)
    if n_outliers > 30:
        mn = n_outliers - 30
        remove_idx = rng.choice(outlier_indices, mn, replace=False)
        x_train = np.delete(x_train, remove_idx, axis=0)
        y_train = np.delete(y_train, remove_idx, axis=0)
    noises = inject_noise(outliers, n_noise, random_seed)
    x_train = np.append(x_train, noises, axis=0)
    y_train = np.append(y_train, np.zeros((noises.shape[0], 1)))
    outlier_indices = np.where(y_train == 1)[0]
    inlier_indices = np.where(y_train == 0)[0]
    n_outliers = len(outlier_indices)
    input_shape = x_train.shape[1:]
    model = deviation_network(input_shape, mode)
```

Рисунок 3.6 – Основна функція навчання

Основні кроки у функції тренування включають в себе ряд етапів які описані нижче.

- Компіляція моделі. Визначення оптимізатора, функції втрат та метрик, які будуть використовуватися під час тренування.
- Тренування моделі. Передача тренувальних даних (ознак і міток) моделі для навчання. Тривалість тренування залежить від кількості епох (повторень навчання на всіх даних) та інших гіперпараметрів.
- Повернення навченої моделі. Функція повертає навчену модель, яку можна використовувати для здійснення передбачень на нових даних.

Повернення навченої моделі зображена на рисунку 3.7.

```
model.fit_generator(batch_generator_sup(x_train, outlier_indices, inlier_indices, batch_size, nb_batch, rng),
                    steps_per_epoch=nb_batch,
                    epochs=epochs)
return model
```

Рисунок 3.7 – Повернення навченої моделі

Функція тренування на створеному нескінченному генераторі, тренування проводиться epochs раз, та максимальна, кількість кроків за епоху, так як генератор нескінченний - *nb_batch.*

3.5 Висновки до розділу

У розділі під заголовком «Програмна реалізація моделі та методу аналізатора трафіку в ІКС» було сформульовано вимоги до програмного засобу, які в себе включають те, що розроблене ПЗ повинно бути у вигляді модулю, який може бути підключений до іншої системи, наприклад, такої як веб-застосунок. Також додаток повинен відповідати деяким вимогам, як можливість перевірки запитів з метою виявлення кіберзагрози. Можливість роботи з інформацією в інших форматах та застосування алгоритмів машинного навчання.

Було обґрунтовано засоби для реалізації, а саме мову програмування та середовище розробки. Мовою програмування було обрано саме Python - це універсальна мова, що зустрічається в безлічі різних застосувань, а у машинному

навчанні вона займає передуюче місце, завдяки своїм ключовим бібліотекам та інструментам для машинного навчання такі як: Pandas, Keras, TensorFlow, Scikit-Learn, NumPy.

На етапі тренування нейронної мережі за допомогою фреймворку TensorFlow було досліджено скільки епох тренування пройшло загалом та їх час, основним показником функції втрат під час тренування це й визначає, що модель більш ніж достатньо справляється з запам’ятовуванням та передбаченням результатів процесу.

Надалі для більш ефективного тренування було реалізовано функцію створення шуму, також й формування генератору нескінченних порцій та багато інших етапів. Наприкінці результати є максимально позитивними, як для машинного навчання нейронної мережі, після цього модель була повернена та її можна використовувати для здійснення передбачень нових даних.

4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ФУНКЦІОНУВАННЯ РОЗРОБЛЕННОЇ МОДЕЛІ ТА МЕТОДУ

4.1 Відповідність роботи моделі до реальних значень

З метою перевірки на працездатність моделі у різних умовах, наприклад таких як: перевірка працездатності моделі на підмножині даних, які не використовувались при тренуванні чи відповідність її до реальних значень, було сформовано процес рандомізації масиву даних, згодом усі оброблені у відповідності до вірного узгодження з показниками кінцевого результату моделі та методу аналізатора трафіку для виявлення кібератак в інформаційно-комунікаційних системах.

Процес перемішування (рандомізації) масиву даних є важливим етапом підготовки даних для машинного навчання. Це допомагає уникнути впливу порядку даних на процес навчання моделі та поліпшує її універсальність.

Після перемішування масиву даних, який зображений на рисунку 4.1, перевіряємо роботу моделі та її відповідність до реальних значень.

```
model = run_devnet('deep', 200, 32, 4)
c = list(zip(x, labels))
random.shuffle(c)
x, labels = zip(*c)
```

Рисунок 4.1 – Функція перемішування масиву даних

Створюємо порцію даних для моделі та отримуємо результати роботи моделі. Якщо результат моделі більше нуля це означає, що вона позначила ці записи як аномальні, далі виводимо реальний лейбл запису, нормальний чи злоякісний в іншому випадку модель позначила запис як нормальний. Перевірку роботи моделі та її відповідність до реальних значень зображено на рисунку 4.2.

```
for res, truth in zip(x[:100], labels[:100]):
    res = np.expand_dims(res, axis=0)
    prediction = model.predict(res, verbose=0)
    if prediction > 0:
        print(f"Warning, this request may relate to DDOS or password fishing attack")
        print(f"In reality, request is {'Normal' if truth == 0 else 'Malignant'}\n")
    else:
        print(f"This request seems normal")
        print(f"In reality, request is {'Normal' if truth == 0 else 'Malignant'}\n")
```

Рисунок 4.2 – Перевірка роботи моделі та її відповідність до реальних значень

Отже, згідно з результатами роботи моделі та значень, які позначають записи по відношенню нормальні чи злоякісні, дозволяється зробити заключення, що розроблена модель повністю відповідає її роботи до реальних значень. Завдяки цьому процесу перевірки, можна переходити до іншого етапу та перевіряти модель на працездатність на підмножині даних, які не використовувались при тренуванні моделі.

4.2 Перевірка працездатності моделі на підмножині даних, які не використовувались при тренуванні моделі

Перевірка працездатності моделі на підмножині даних, які не використовувались при тренуванні (так звана тестова вибірка), є критичним етапом в оцінці ефективності моделі. Цей процес дозволяє визначити, наскільки добре модель генералізує свої знання на нових даних, які вона раніше не бачила. Перевірку працездатності моделі на підможині даних, які не використовувались при тренуванні моделі зображено на рисунку 4.3.

```
Predicted Output: This request seems normal, score = [[-0.02930745]]
In reality, request is Normal

Predicted Output: This request seems normal, score = [[-0.02930745]]
In reality, request is Normal

Predicted output: Warning, this request may relate to DDOS or password fishing attack, score = [[2.8368928]]
In reality, request is Malignant

Predicted Output: This request seems normal, score = [[-0.02930745]]
In reality, request is Normal

Predicted output: Warning, this request may relate to DDOS or password fishing attack, score = [[2.8108845]]
In reality, request is Malignant

Predicted output: Warning, this request may relate to DDOS or password fishing attack, score = [[2.7920513]]
In reality, request is Malignant
```

Рисунок 4.3 – Перевірка працездатності моделі на підмножині даних, які не використовувались при тренуванні моделі

Процес перевірки працездатності моделі, закладається в тому, що модель перевіряється на підмножині даних яка не використовувалась при тренуванні моделі.

Кожний запис містить 2 частини, перша частина - передбачення моделі, з загальною оцінкою об’єкту, в даному випадку веб-запиту. Передбачення моделі, зокрема в контексті обробки веб-запитів, може включати в себе оцінку об'єкта (наприклад, чи є цей веб-запит підозрілим з точки зору кібербезпеки). Оцінка може бути представлена числовим значенням (ймовірність, що запит є підозрілим) або міткою класу (позитивний або негативний).

Наприклад, якщо ми маємо модель для виявлення зловмисного веб-трафіку або атак на веб-додатки, оцінка може вказувати на те, наскільки ймовірно, що даний веб-запит є атакою. Також треба додати, що саме розроблена модель після процесу перемішування масиву даних може повертати ймовірність або бінарну мітку (підозрілий/не підозрілий).

Друга частина даного процесу на перевірку працездантності моделі на підмножині даних, які не використовувались при тренуванні моделі містить реальний лейбл даного запиту, тобто він нормальний (Normal), звичайний, чи відноситься до кібератаки (Malignant).

4.3 Результати тестування

Модель показала чіткі та позитивні результати з точності виявлення кібератак в інформаційно-комунікаційних системах та порівняно низькі показники хибних спрацювань.

Загальний процент коректності та розпізнавання аномальних активностей та ідентифікувати потенційні загрози роботи системи склав 91%, що вважається високим результатом для систем такого рівня. Порівняльна статистика моделі з аналогами на точність виявлення кіберзагроз представлена у виді таблиці, та продемонстрована на таблиці 4.1.

Таблиця 4.1 – Порівняльна статистика моделі з аналогами на точність виявлення кіберзагроз

Модель				
Датасет	Запропонована модель	REPEN	FSNET	iForest
MSCAD	0.887±0.002	0.794±0.005	0.808±0.027	0.874±0.015
FRAUD	0.910±0.001	0.912±0.001	0.732±0.020	0.624±0.020
Backdoor	0.900±0.004	0.842±0.006	0.734±0.046	0.752±0.021

Отже, розроблено програмний засіб повинен бути у вигляді модулю, що може бути підключений до іншої системи, наприклад веб-застосунку. За результатами тестування модель показала невеликі затрати по часу на виявлення кібератаки та високий рівень точності її виявлення.

4.4 Висновки до розділу

У даному розділі було розглянуто тестування розробленого програмного засобу. Також було пройдена відповідність на коректність розробленого

програмного засобу на такі перевірки працездатності моделі як відповідність роботи моделі до реальних значень та перевірка працездатності моделі на підмножині даних, які не використовувались при тренуванні (так звана тестова вибірка). Було сформовано процес рандомізації масиву даних, згодом усі оброблені у відповідності до вірного узгодження з показниками кінцевого результату моделі та методу аналізатора трафіку для виявлення кібератак в інформаційно-комунікаційних системах. Щодо процесу перевірки працездатності моделі на підмножині даних, які не використовувались при тренуванні, то цей процес дозволяє визначити, наскільки добре модель генералізує свої знання на нових даних, які вона раніше не бачила.

Розроблена модель продемонструвала чіткі результати з точності виявлення кібератак в інформаційно-комунікаційних системах та порівняно низькі показники хибних спрацювань.

Також треба додати, що загальний процент коректності та розпізнавання аномалій та ідентифікація потенційних загроз роботи системи склав 91%, що вважається високим результатом для систем такого рівня.

5 ЕКОНОМІЧНА ЧАСТИНА

Для успішного впровадження науково-технічної розробки надзвичайно важливо, щоб вона відповідала поточним вимогам науково-технічного прогресу і враховувала економічні аспекти. Оцінка економічної ефективності результатів науково-дослідної роботи є ключовою частиною цього процесу. Дослідження, яке представлене у магістерській роботі і присвячене розробці та вивченню "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту", віднесено до науково-технічних проектів, спрямованих на введення на ринок. Рішення про комерціалізацію розробки може бути прийняте протягом виконання самої роботи, відкриваючи можливості для подальшого введення на ринок. Цей напрямок визначається як пріоритетний, оскільки розроблені результати можуть бути корисними для різних зацікавлених сторін і приносити економічні вигоди. Однак для успішної реалізації цього процесу вирішальним є залучення зацікавленого інвестора, який виявить інтерес до втілення даного проекту, і переконання його у доцільності інвестування у цю розробку. З метою досягнення цього завдання були визначені такі етапи виконання робіт:

1. Проведення комерційного аудиту науково-технічної розробки, включаючи визначення науково-технічного рівня та комерційного потенціалу.
2. Розрахунок витрат на реалізацію науково-технічної розробки.
3. Проведення розрахунку економічної ефективності впровадження та комерціалізації науково-технічної розробки для потенційного інвестора, а також обґрунтування економічної доцільності комерціалізації з точки зору інвестора.

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту" є підвищення ефективності в інформаційно-комунікаційних системах захисту, а також доведення важливості використання штучного інтелекту та вдосконалення його алгоритмів в інформаційно-комунікаційних системах.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1. [22].

Таблиця 5.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуютьс я у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці. Для оцінки науково-технічного рівня і комерційного потенціалу розробки експертами було запрошено трьох незалежних експертів: Вадим Ігорович Маліновський к.т.н. доцент кафедри захисту інформації Вінницького національного технічного університету, Дар'я Рузметова інженер штучного інтелекту ТОВ «ППНР», Андрій Гонца спеціаліст з налаштування інформаційних систем ТОВ «ПЛЕЙТИКА УКРАЇНА».

Таблиця 5.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	Вадим Маліновський	Дар'я Рузметова	Андрій Гонца
	Бали, виставлені експертами:		
1. Технічна здійсненність концепції	4	3	4
2. Ринкові переваги (наявність аналогів)	3	4	4
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	2	3	3
5. Ринкові переваги (експлуатаційні витрати)	4	3	4
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	1	2	2
8. Практична здійсненність (наявність фахівців)	4	3	4
9. Практична здійсненність (наявність фінансів)	3	4	3
10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	4	4	4
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	$СБ_1$=39	$СБ_2$=40	$СБ_3$=42
Середньоарифметична сума балів $СБ_c$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{39+40+42}{3} = 40.3$		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 5.3. [22].

Таблиця 5.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту" становить 40 балів, що, відповідно до таблиці 5.3 рівень комерційного потенціалу розробки високий, що свідчить про комерційну важливість проведення даних досліджень.

Магістерська кваліфікаційна робота "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту" відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок, тобто при цьому відбувається комерціалізація науково-технічної розробки. Цей напрямок є для нас пріоритетним, оскільки результатами розробки можуть користуватися не тільки самі розробники, а й інші споживачі, отримуючи при цьому суттєвий економічний ефект.

Результатом магістерської роботи є модель і метод аналізатора трафіку в інформаційно-комунікаційних системах для виявлення кібератак.

5.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

В якості аналога для розробки було обрано аналог iForest. До недоліків можна віднести точність виявлення кібератак.

У розробці дана проблема вирішується за допомогою попередньої перевірки методом семантичного аналізу, яка допомагає відсіяти інформацію, що точно не містить факторів кіберзагрози. Також система випереджає аналог за такими параметрами як швидкість роботи.

Одиничний параметричний індекс розраховуємо за формулою:

$$q_i = \frac{P_i}{P_{базі}}. \quad (5.1)$$

де q_i – одиничний параметричний індекс, розрахований за i-м параметром;

P_i – значення i-го параметра виробу;

$P_{базі}$ – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 5.4.

Таблиця 5.4 – Основні техніко-економічні показники аналога та розробки, що проектується

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
1	2	3	4	5
Доступність сервісу,%	90	100	1,11	10%
Час виявлення кіберзагрози, мкс	40	50	1,25	25%
Кількість виявлених вразливостей, шт	4	2	2	20%
Точність виявлення кіберзагрози, %	82	91	1,1	45%

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою:

$$I_{НП} = \prod_{i=1}^{n} q_i \ , \qquad (5.2)$$

де $I_{нп}$ – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i-м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому Інп = 1.

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра:

$$I_{ТП} = \sum_{i=1}^{n} q_i \cdot \alpha_i \ , \qquad (5.3)$$

де $I_{тп}$ – груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

q_i – одиничний параметричний показник *i*-го параметра;

α_i – вагомість *i*-го параметричного показника, $\sum_{i=1}^{n}\alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{mn} = 1{,}11 \cdot 0{,}1 + 1{,}25 \cdot 0{,}25 + 2 \cdot 0{,}2 + 1{,}1 \cdot 0{,}45 = 1{,}32.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою:

$$I_{ЕП} = \sum_{i=1}^{m} q_i \cdot \beta_i \,, \qquad (5.4)$$

де $I_{ЕП}$ – груповий параметричний індекс за економічними показниками;

q_i – економічний параметр *i*-го виду;

β_i – частка *i*-го економічного параметра, $\sum_{i=1}^{m}\beta_i = 1$;

m – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці .

$$I_{ЕП} = 0{,}75 \cdot 0{,}5 + 0{,}86 \cdot 0{,}5 = 0{,}80$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою:

$$K_{ІНТ} = I_{НП} \cdot \frac{I_{ТП}}{I_{ЕП}} \,, \qquad (5.5)$$

$$K_{ІНТ} = 1 \cdot 1{,}32 / 0{,}80 = 1{,}65.$$

Інтегральний показник конкурентоспроможності $К_{ІНТ} > 1$, отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту", під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Зо) розраховуємо у відповідності до посадових окладів працівників, за формулою:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \qquad (5.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 20000 \cdot 5 / 21 = 4545 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	20000	909,1	5	4545
Інженер	15000	681,8	35	23864
Всього				28409

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Зр) за відповідними найменуваннями робіт НДР на тему "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту" розраховуємо за формулою:

$$З_p = \sum_{i=1}^{n} C_i \cdot t_i \,, \tag{5.7}$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду Сі можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} \,, \tag{5.8}$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо M_M=6500 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 6700{,}00 \cdot 1 \cdot 1{,}65 / (21 \cdot 8) = 65{,}8$ грн.
$З_{р1} = 65{,}8 \cdot 12 = 789{,}6$ грн.

Таблиця 5.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
1.Підготовчі	12	1	65,8	789,6
2.Монтажні	37	3	88,8	3286,9
3.Складальні	35	5	111,9	3915,3
4.Налагоджувальні	47	2	72,4	3402,0
5.Випробувальні	28	4	59,8	1675,0
Всього				13068,9

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (5.9)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 11%.

$З_{дод}$ = (28409+13068,9) · 11 / 100% = 4562,58 грн.

5.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (5.10)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$З_н$ = (28409+13068,9+4562,58) · 22 / 100% = 10128,92 грн.

1.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту".

Витрати на матеріали *(М),* у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^{n} H_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} B_j \cdot Ц_{вj}, \qquad (5.11)$$

де H_j – норма витрат матеріалу *j*-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу *j*-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 … 1,15*);

B_j – маса відходів *j*-го найменування, кг;

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

Проведені розрахунки зведемо до таблиці.

Таблиця 5.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний (А4)	170	1	170
ручка	19	1	19
Флешка	250	1	250
Всього			439
З врахуванням коефіцієнта транспортування			482,9

1.3.4 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (5.12)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл} = (50000 \cdot 1) / (2 \cdot 12) = 2083{,}33$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 5.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Комп'ютер	50000	2	1	2083,33
Робоче місце розробника ПЗ	220000	20	2	1833,33
Всього				3916,67

1.3.5 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ($B_е$) розраховуємо за формулою:

$$B_е = \sum_{i=1}^{n} \frac{W_{уі} \cdot t_i \cdot Ц_е \cdot К_{впі}}{\eta_i}, \quad (5.13)$$

де $W_{уі}$ – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_е$ = 7,5 грн;

$К_{впі}$ – коефіцієнт, що враховує використання потужності, $К_{впі} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$В_е = 0,25 \cdot 295,0 \cdot 7,5 \cdot 0,5 / 0,8 = 345,70$ грн.

1.3.6 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту" належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \qquad (5.14)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 20%.

$В_{св} = (28409+13068,9) \cdot 20 / 100\% = 8295,6$ грн.

1.3.7 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (5.15)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$І_в$ = (28409+13068,9) · 50 / 100% = 20738,99 грн.

1.3.8 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (5.16)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$В_{нзв}$ = (28409+13068,9) · 100 / 100% = 41477,98 грн.

Витрати на проведення науково-дослідної роботи на тему "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту" розраховуємо як суму всіх попередніх статей витрат за формулою:

$$В_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}. \quad (5.17)$$

$В_{заг}$=28409+13068,9+4562,58+10128,92+482,9+3916,67+345,70+8295,6+20738,99 +41477,98 = 131427,31 грн.

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{В_{заг}}{\eta}, \quad (5.18)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,5.

ЗВ = 131427,31 / 0,5 = 262854,63 грн.

5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту" передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, приймемо 1 особа

$Ц_б$ – вартість послуги у році до впровадження інформаційної системи, приймемо 30000,00 грн;

$\pm\Delta Ц_о$ – зміна вартості послуги від впровадження результатів, приймемо зростання на 1000,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 3-х років, протягом яких очікується отримання позитивних результатів

від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою:

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (5.19)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо $\rho = 40\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (1 \cdot 1000 + 30000 \cdot 70) \cdot 0,83 \cdot 0,4 \cdot (1\text{-}0,18/100\%) = 370864,33$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (1 \cdot 1000 + 30000 \cdot (20+50)) \cdot 0,83 \cdot 0,4 \cdot (1\text{-}0,18/100\%) = 636474,58$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (1 \cdot 1000 + 30000 \cdot (70+50+60)) \cdot 0,83 \cdot 0,4 \cdot (1\text{-}0,18/100\%) = 954211,87$ грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (5.20)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=18%;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = 370864,33 /(1+0,18)^1+636474,58 /(1+0,18)^2+954211,87 /(1+0,18)^3=$$
$$= 1305824,19 \text{ грн.}$$

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \tag{5.21}$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 262854,63 грн.

$PV = k_{інв} \cdot ЗВ$ = 2 · 262854,63 = 525709,26 грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{5.22}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 1305824,19грн;

PV – теперішня вартість початкових інвестицій, 525709,26 грн.

$E_{абс} = ПП - PV$ = 1305824,19- 525709,26 = 780114,93 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Тж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.23)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 3 роки.

$$E_в = \sqrt[Тж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+780114{,}93 /525709{,}26)^{1/3} - 1 = 0{,}58.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (5.24)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,25.

$\tau_{мін}$ = 0,1+0,25 = 0,35 < 0,58 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія онтологічного моделювання бази знань з організації бібліотеки» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \quad (5.25)$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 0,58 = 1,7р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

5.5 Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту" становить 40 балів, що, свідчить про комерційну важливість проведення даних досліджень оскільки рівень комерційного потенціалу розробки високий.

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,65 рази.

Також термін окупності становить 1,7 роки, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою "Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту".

ВИСНОВКИ

Результатом виконання магістерської кваліфікаційної роботи є модель та метод виявлення кібератак в інформаційно-комунікаційних система на базі алгоритмів штучного інтелекту. За допомогою розробленого засобу у вигляді модулю, що може бути підключений до іншої системи, наприклад веб-застосунку, який відповідає наступним вимогам: наявність можливості підключення до веб-застосунку; можливість перевірки запитів з метою виявлення потенційно небезпечних запитів; наявність можливості роботи з інформацією у .json, а також інших форматах; застосування алгоритмів машинного навчання;

Модель показала чіткі та позитивні результати з точності виявлення кібератак в інформаційно-комунікаційних системах та порівняно низькі показники хибних спрацювань.

Загальний процент коректності та розпізнавання аномальних активностей та ідентифікувати потенційні загрози роботи системи склав 91%, що вважається високим результатом для систем такого рівня.

Крім того, за результатами тестування модель показала невеликі затрати по часу на виявлення кібератаки.

Можна зробити висновок, що розроблена модель та метод є перспективним рішенням у покращенні кібербезпеки в інформаційно-комунікаційних системах, магістерська кваліфікаційна робота демонструє самі перспективи засобів і моделей виявлення кіберзагроз на базі підходів і моделей штучного інтелекту та обумовлює актуальні тенденції їх подальшого розвитку.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Martti Lehto, Pekka Neittaanmäki Cyber Security. – Springer International Publishing, 2022. – 234p.
2. Єременко В. С. Застосування нейромережевих технологій у системах неруйнівного контролю / В. С. Єременко, А. В. Переїденко, О. В. Монченко // Технічна діагностика і неруйнівний контроль. – 2016. – № 1. – С.35-41.
3. Моделі і принципи захисту інформаційних даних в системах безпеки пристроїв Інтернету речей [Електронний ресурс]//В. І. Маліновський, А.О. Димов, Д. А. Василевський – Збірник матеріалів Міжнародної Інтернет-конференції “Світ наукових досліджень." (випуск 25)"- 14-15 грудня червня 2023. – Режим доступу: URL: https://www.economy-confer.com.ua/full-article/5172/ (дата звернення 13.12.2023р.).
4. K.P. Murphy, Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series) / Kevin Patrick Murphy, – 2012. – MIT Press. – 1104 p.
5. Черноусов, А. В. Метод автоматичного виявлення помилок безпеки в програмному забезпеченні на основі глибинного навчання: магістерська дис. : 125 Кібербезпека / Черноусов Артем Вікторович. – Київ, 2019. – 85 с.
6. Olexa Riznyk. Colored neural network.svg: Glosser.caderivative work [Електронний ресурс]: – / Olexa Riznyk // CC BY-SA 3.0. – Тип доступу URL: https://commons.wikimedia.org/w/index.php?curid=65466580 (дата звернення 09.12.23).
7. Переїденко А. В. Система класифікації дефектів на основі штучних нейронних мереж / А. В. Переїденко, В. С. Єременко, Ж. О. Павленко // Вісн. НТУУ «Київ. політехн. ін-т». — 2010. — № 40. – С. 72–80.
8. Гулак Г. М. Методологія захисту інформації. Аспекти кібербезпеки: підручник. – Київ: Видавництво НА СБ України. – 2020. –256 с.

9. Що таке IPS/IDS і де застосовується [Електронний ресурс]: – Тип доступу URL: https://www.hostzealot.com.ua/blog/about-solutions/shho-take-ipsids-i-de-zastosovujetsya#article-anchor-0 (дата звернення 29.11.23).
10. Martti Lehto Cyber Security. Critical Infrastructure Protection / Martti Lehto, Pekka Neittaanmäki. – Springer International Publishing, 2022. – 500p.
11. Chio K. Machine learning and safety / K. Chio, D. Freeman. — M.: DMK Press, 2019. — 388 p.
12. Kostas K. Anomaly Detection in Networks Using Machine Learning [Text]/ Kostas K. – 2018. – part 1. – 64p.
13. Anomaly Detection in High Dimensional Data //Dilini Talagala, Priyanga; Hyndman, Rob J.; Smith-Miles, Kate. – 2019. – pp.260-312.
14. Dr.Pramod Pandya "Network and System Security (Second Edition). – 2014. – Chapter 9. – pp.259-290.
15. Комп'ютерні мережі / О. Д. Азаров, С. М. Захарченко, О. В. Кадук [та ін.]. Вінниця : ВНТУ, 2020. – 378 с.
16. Закон України «Про основні засади забезпечення кібербезпеки України» [Електронний ресурс]: Закон України від 05.10.2017 р. № 2163-VIII : станом на 17 серп. 2022 р. - Тип доступу URL: https://zakon.rada.gov.ua/laws/show/2163-19#Text (дата звернення: 10.11.2023).
17. Розпорядження Кабінету міністрів України № 1556-р від 02.12.2020. «Концепція розвитку штучного інтелекту в Україні» [Електронний ресурс]: Київ. 2019р. – 25с. – Тип доступу URL: https://zakon.rada.gov.ua/laws/show/1556-2020-%D1%80#Text / (дата звернення 10.12.2023р.).
18. Keras 3. API documentation [Електронний ресурс]: – Тип доступу URL: https://keras.io/guides/ (дата звернення 09.11.23).
19. Jamil Al-Sawwa .Multi-Step Cyber-Attack Dataset (MSCAD) [Data set]: / Dr. Jamil Al-Sawwa, Dr. Mohammad Almseidin, Dr. Mouhammd Alkasassbeh. – 2022.- – Тип доступу URL: https://doi.org/10.34740/KAGGLE/DSV/3830715 (дата звернення 10.11.23)

20. Almseidin, Mohammad. Generating a Benchmark Cyber Multi-step Attacks Dataset for Intrusion Detection / Almseidin, Mohammad, Al-Sawwa, Jamil, and Alkasassbeh, Mouhammd.. – 1.01. 2022 : part 1 – 15p.
21. Word2vec Tutorial [Електронний ресурс]: – Тип доступу URL: https://www.tensorflow.org/text/tutorials/word2vec (дата звернення 11.11.23).
22. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

ДОДАТКИ

Додаток А
Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: Модель і метод виявлення кібератак в інформаційно-комунікаційних системах на базі алгоритмів штучного інтелекту

Автор роботи: Димов Анатолій Олександрович

Тип роботи: магістерська кваліфікаційна робота

Підрозділ кафедра захисту інформації ФІТКІ
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність – 95,05 %. Схожість – 4,95 %.

Аналіз звіту подібності (відмітити потрібне):

√ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

√ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

√ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ (підпис) Валентина КАПЛУН

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Анатолій ДИМОВ

Керівник роботи _____ (підпис) Вадим МАЛІНОВСЬКИЙ

Додаток Б
Текст програми

```
import pandas as pd
import numpy as np
from keras.layers import Input, Dense
from sklearn.model_selection import train_test_split
from keras import regularizers, Model
from keras.optimizers import RMSprop
from keras import backend as K
import tensorflow as tf
import random
import sys



pandas.DataFrame
data = pd.read_csv("/kaggle/input/mscad/MSCAD.csv")

def remapping_func(el):
    if el == "Brute_Force":
        return 1
    else:
        return 0

data['Label'] = data['Label'].map(remapping_func)

labels = data['Label'].tolist()
labels = np.array(labels, dtype=np.float32)
x = data.drop('Label', axis=1)
x = x.to_numpy(dtype=np.float32)

MAX_INT = np.iinfo(np.int32).max

def dev_network_d(input_shape):

    x_input = Input(shape=input_shape)
    intermediate = Dense(1000, activation='relu',
                         kernel_regularizer=regularizers.l2(0.01),
```

```
name='hl1')(x_input)
    intermediate = Dense(250, activation='relu',
                         kernel_regularizer=regularizers.l2(0.01),
name='hl2')(intermediate)
    intermediate = Dense(20, activation='relu',
                         kernel_regularizer=regularizers.l2(0.01),
name='hl3')(intermediate)
    intermediate = Dense(1, activation='linear', name='score')(intermediate)
    return Model(x_input, intermediate)

def dev_network_s(input_shape):
    x_input = Input(shape=input_shape)
    intermediate = Dense(20, activation='relu',
                         kernel_regularizer=regularizers.l2(0.01),
name='hl1')(x_input)
    intermediate = Dense(1, activation='linear', name='score')(intermediate)
    return Model(x_input, intermediate)
ref = K.variable(np.random.normal(loc=0., scale=1.0, size=2000), dtype='float32')

def deviation_loss(y_true, y_pred):
    global ref
    y_true = tf.cast(y_true, dtype='float32')
    y_pred = tf.cast(y_pred, dtype='float32')
    dev = (y_pred - K.mean(ref)) / K.std(ref)
    inlier_loss = K.abs(dev)
    outlier_loss = K.abs(K.maximum(confidence_margin - dev, 0.))
    return K.mean((1 - y_true) * inlier_loss + y_true * outlier_loss)

def deviation_network(input_shape, mode='shallow'):
    if mode == "deep":
        model = dev_network_d(input_shape)
    elif mode == "shallow":
        model = dev_network_s(input_shape)
    else:
        sys.exit("The network depth is not set properly")
    rms = RMSprop(clipnorm=1.)
    model.compile(loss=deviation_loss, optimizer=rms)
    return model
```

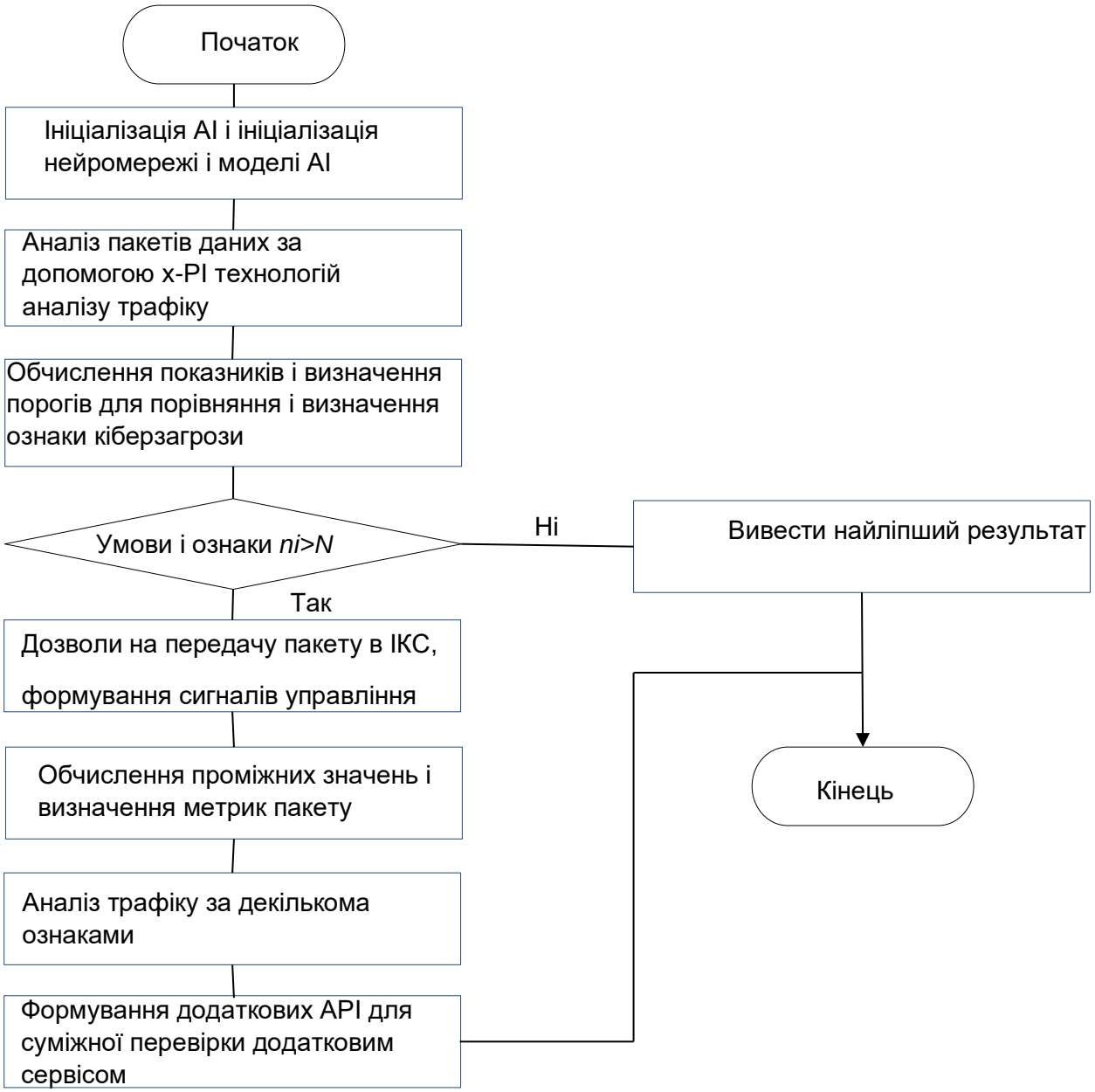
```
def inject_noise(seed, n_out, random_seed):
    rng = np.random.RandomState(random_seed)
    n_sample, dim = seed.shape
    swap_ratio = 0.05
    n_swap_feat = int(swap_ratio * dim)
    noise = np.empty((n_out, dim))
    for i in np.arange(n_out):
        outlier_idx = rng.choice(n_sample, 2, replace=False)
        o1 = seed[outlier_idx[0]]
        o2 = seed[outlier_idx[1]]
        swap_feats = rng.choice(dim, n_swap_feat, replace=False)
        noise[i] = o1.copy()
        noise[i, swap_feats] = o2[swap_feats]
    return noise

def input_batch_generation_sup(x_train, outlier_indices, inlier_indices, batch_size,
rng):
    dim = x_train.shape[1]
    ref = np.empty((batch_size, dim))
    training_labels = []
    n_inliers = len(inlier_indices)
    n_outliers = len(outlier_indices)
        if (i % 2 == 0):
            sid = rng.choice(n_inliers, 1)
            ref[i] = x_train[inlier_indices[sid]]
            training_labels += [0]
        else:
            sid = rng.choice(n_outliers, 1)
            ref[i] = x_train[outlier_indices[sid]]
            training_labels += [1]
    return np.array(ref), np.array(training_labels)

def  batch_generator_sup(x,  outlier_indices,  inlier_indices,  batch_size,  nb_batch,
rng):
    rng = np.random.RandomState(rng.randint(MAX_INT, size=1))
    counter = 0
    while 1:
```

```
        ref,   training_labels   =   input_batch_generation_sup(x,   outlier_indices,
inlier_indices, batch_size, rng)
        counter += 1
        yield (ref, training_labels)
        if (counter > nb_batch):
            counter = 0
def run_devnet(mode, epochs, batch_size, nb_batch):
    random_seed = 42
    outlier_indices = np.where(labels == 1)
    outliers = x[outlier_indices]
    x_train, x_test, y_train, y_test = train_test_split(x, labels, test_size=0.2,
random_state=42, stratify=labels)
    outlier_indices = np.where(y_train == 1)[0]
    # визначаємо кількість аномалій
    n_outliers = len(outlier_indices)
    n_noise = len(np.where(y_train == 0)[0]) * 0.02 / (1. - 0.02)
    n_noise = int(n_noise)
    rng = np.random.RandomState(random_seed)
    if n_outliers > 30:
        mn = n_outliers - 30
        remove_idx = rng.choice(outlier_indices, mn, replace=False)
        x_train = np.delete(x_train, remove_idx, axis=0)
        y_train = np.delete(y_train, remove_idx, axis=0)

    noises = inject_noise(outliers, n_noise, random_seed)
    x_train = np.append(x_train, noises, axis=0)
    y_train = np.append(y_train, np.zeros((noises.shape[0], 1)))

    outlier_indices = np.where(y_train == 1)[0]
    inlier_indices = np.where(y_train == 0)[0]
    n_outliers = len(outlier_indices)
    input_shape = x_train.shape[1:]
    model = deviation_network(input_shape, mode)
    model.fit_generator(batch_generator_sup(x_train, outlier_indices, inlier_indices,
batch_size, nb_batch, rng),
                        steps_per_epoch=nb_batch,
                        epochs=epochs)
    return model
```

```
model = run_devnet('deep', 200, 32, 4)
c = list(zip(x, labels))
random.shuffle(c)
x, labels = zip(*c)
for res, truth in zip(x[:100], labels[:100]):
    res = np.expand_dims(res, axis=0)
    prediction = model.predict(res, verbose=0)
    if prediction > 0:
        print(f"Warning, this request may relate to DDOS or password fishing attack")
        print(f"In reality, request is {'Normal' if truth == 0 else 'Malignant'}\n")
    else:
        print(f"This request seems normal")
        print(f"In reality, request is {'Normal' if truth == 0 else 'Malignant'}\n")
```

ІЛЮСТРАТИВНА ЧАСТИНА

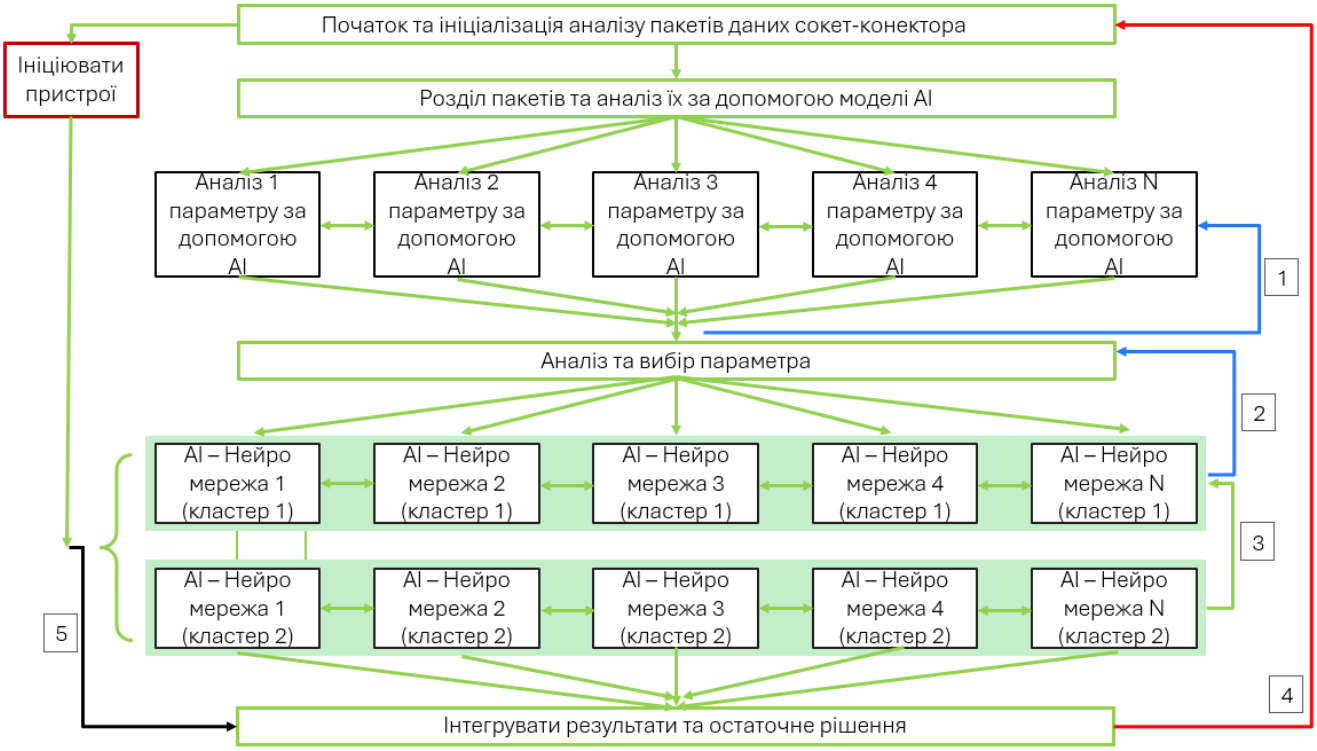
Алгоритм роботи аналізатора трафіку і виявлення кібератак



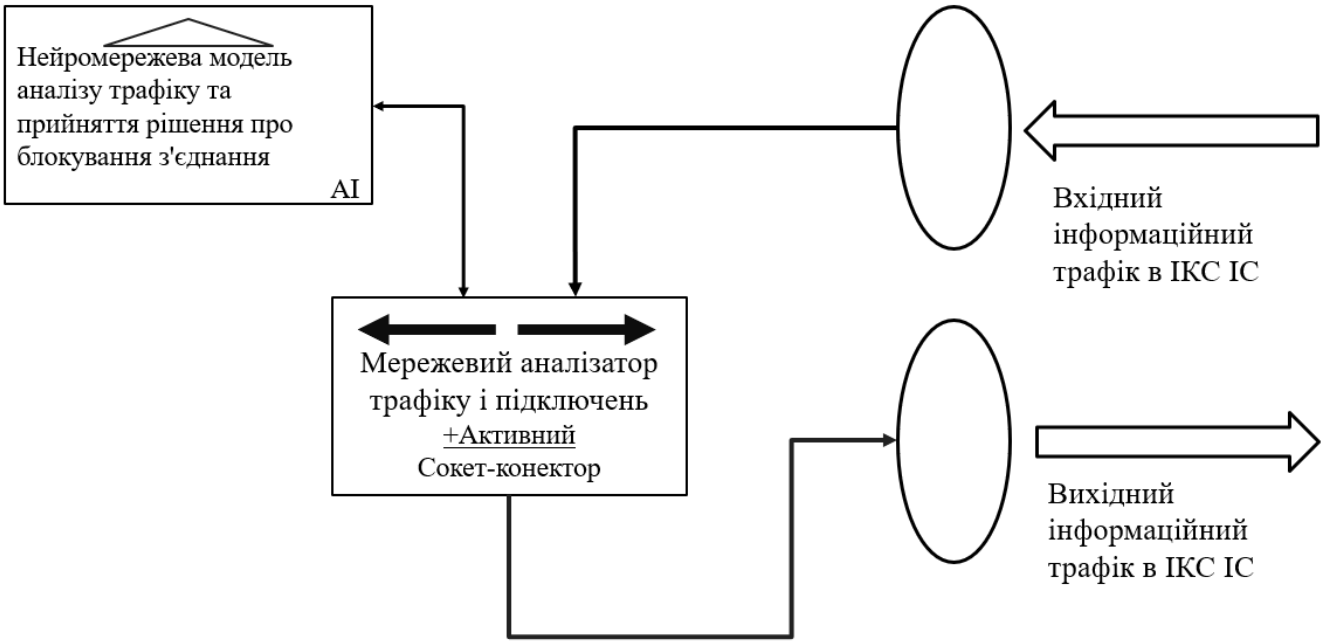
Реалізація циклу по розміру порції

```
for i in range(batch_size):
    if (i % 2 == 0):
        sid = rng.choice(n_inliers, 1)
        ref[i] = x_train[inlier_indices[sid]]
        training_labels += [0]
```

Модель побудови технології аналізу трафіку на базі нейропідходів



Базова модель безпеки ІКС на базі ШІ



Функція створення глибокої архітектури нейронної мережі

```
# Функція створення глибокої архітектури нейронної мережі, глибина - 2
#у мкр не описано
def dev_network_s(input_shape):
    x_input = Input(shape=input_shape)
    intermediate = Dense(20, activation='relu',
                         kernel_regularizer=regularizers.l2(0.01), name='hl1')(x_input)
    intermediate = Dense(1, activation='linear', name='score')(intermediate)
    return Model(x_input, intermediate)
```

Функція створення шуму

```
def inject_noise(seed, n_out, random_seed):
    rng = np.random.RandomState(random_seed)
    n_sample, dim = seed.shape
    swap_ratio = 0.05
    n_swap_feat = int(swap_ratio * dim)
    noise = np.empty((n_out, dim))
    for i in np.arange(n_out):
        outlier_idx = rng.choice(n_sample, 2, replace=False)
        o1 = seed[outlier_idx[0]]
        o2 = seed[outlier_idx[1]]
        swap_feats = rng.choice(dim, n_swap_feat, replace=False)
        noise[i] = o1.copy()
        noise[i, swap_feats] = o2[swap_feats]
    return noise
```