Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інформаційна технологія для виявлення HTTP запитів з аномальною поведінкою»

Виконав: студент 2-го курсу, групи 2КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
_____ Зелений В.Є.
(прізвище та ініціали)
Керівник: к.т.н., доцент каф. КН
_____ Арсенюк І.Р.
(прізвище та ініціали)
«15» 12 2022
Опонент: к.т.н., доцент каф. Автоматизації та Інтелектуальних Інформаційних Технологій
_____ Маслій Р.В.
(прізвище та ініціали)
«15» 12 2022

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)
«16» 12 2022 р.

Вінниця ВНТУ - 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти ІІ-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.
14.09 2022 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Зеленому Владиславу Євгеновичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційна технологія для виявлення HTTP запитів з аномальною поведінкою.

керівник роботи: к.т.н., доцент кафедри КН Арсенюк І.Р.
затверджені наказом вищого навчального закладу від "14" 09 2022 року № 203
2. Строк подання студентом роботи 13 листопада 2022 року
3. Вихідні дані до роботи:
вхідні дані – класифікація щонайменше 8 різних типів аномалій, кількість рядків даних для тренування – не менше 1 мільйону, кількість рядків для тестування – не менше 500 тисяч.
4. Зміст текстової частини:
Вступ, аналіз предметної області та обґрунтування доцільності розробки інформаційної технології для виявлення HTTP запитів з аномальною поведінкою, проєктування інформаційної технології для виявлення HTTP запитів з аномальною поведінкою, розробка програмного забезпечння інформаційної технології для виявлення HTTP запитів з аномальною поведінкою, економічна частина, висновки, перелік використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
Загальний алгоритм надання та відображення даних веб-додатком, структура інформаційної технології, загальна схема алгоритму роботи інформаційної технології, схема алгоритму роботи модуля, що аналізує дані, результат машинного навчання для k-NN моделі, приклад класифікаційного звіту, класифікаційний звіт моделі дерева рішень, приклад роботи API ендпоїнта.

Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата завдання видав	Підпис, дата виконання прийняв
1-4	Арсенюк І.Р., к.т.н., доц. каф. КН	14.09.2022	14.10.2022
5	Бурєннікова Н.В., д. е. н., проф. каф. ЕПВМ	19.09.2022	12.10.2022

Дата видачі завдання 14.09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз предметної області. Сучасний стан інформаційних технологій та сервісів з виявлення HTTP запитів з аномальною поведінкою	14.09.2022 - 1.10.2022	Розд. 1
2	Розробка алгоритму інформаційної технології	2.10.2022 - 10.10.2022	Розд. 2
3	Проєктування інформаційної технології для виявлення запитів з аномальною поведінкою	11.10.2022 - 21.10.2022	Розд. 3
4	Розробка модулів інформаційної технології та мікросервісу	22.10.2022 - 10.11.2022	Розд. 4
5	Тестування інформаційної технології та аналіз результатів	11.11.2022 - 17.11.2022	Розд. 4
6	Економічна частина	18.11.2022 - 24.11.22	Розд. 5
7	Апробація результатів дослідження	25.11.2022 - 1.12.2022	Тези доповіді
8	Оформлення пояснювальної записки, графічного матеріалу, презентації	2.12.2022 - 14.12.2022	Пояснювальна записка, презент.

Студент (підпис) Зелений В.Є.

Керівник роботи (підпис) Арсенюк І.Р.

АНОТАЦІЯ

УДК 004.8

Зелений В.Є. Інформаційна технологія для виявлення HTTP запитів з аномальною поведінкою. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма - комп'ютерні науки. Вінниця: ВНТУ, 2022. 120 с.

Українською мовою. Бібліогр.: 42 назв; рис.: 29; табл. 15.

У магістерській кваліфікаційній роботі розроблено інформаційну технологію для виявлення HTTP запитів з аномальною поведінкою. Проаналізовано комерційні рішення виявлення HTTP запитів з аномальною поведінкою, запропоновано новий формат програмного продукту для виконання задачі аналізу. Проаналізовано способи машинного навчання, імплементовано найдоречніші з метою виявлення найточнішого. Виконано аналіз усіх даних з набору та запропоновано власну модель HTTP запиту для машинного навчання. Обрано модель дерева рішень, результати якої досягли показників точності у 99%. Модель треновано на вибірці з понад 1 мільйон стрічок даних, тестування виконано на вибірці з 500 тисяч стрічок даних. Розроблено швидкісний мікросервіс, що є носієм інформаційної технології та усуває недоліки існуючих аналогів, не потребуючи втручання технічного спеціаліста, знань предметної області та тривалого навчання.

Графічна частина складається з 8 плакатів.

У економічному розділі, оцінивши інформаційну технологію та ідею, визначено, що розробка відповідає при значенні інтегральної новизни 0,515 - рівню достатня новизна, термін окупності становить 0,87 р., що свідчить про комерційну привабливість науково-технічної розробки. Рівень комерційного потенціалу розробки становить 44,3 бали (високий), що свідчить про комерційну важливість проведення даних досліджень.

Ключові слова: виявлення аномалій, машинне навчання, дерево рішень, HTTP запит.

ABSTRACT

Zelenyi V. Y. Information technology for detecting HTTP requests with anomalous behavior. Master's qualification thesis on specialty 122 - computer science, educational program - computer science. Vinnytsia: VNTU, 2022. 120 p.

In Ukrainian language. Bibliographer: 42 titles; fig.: 29; tab: 15.

The master's thesis developed an information technology for detecting HTTP requests with anomalous behavior. Existing commercial publicly available solutions for detecting HTTP requests with anomalous behavior are analyzed, and a new software product format is proposed for performing the analysis task. The methods of machine learning to detect anomalies were analyzed, the most appropriate ones were implemented in order to detect the most accurate ones. Analysis of all data from the set was performed and a custom HTTP request model was proposed for machine learning. Based on the results of the models, a decision tree model was chosen, the results of which reached 99% accuracy. The model was trained on a sample of more than 1 million data tapes, the model was tested on a sample of 500 thousand data tapes. A high-speed microservice has been developed, which is a carrier of information technology for detecting HTTP requests with anomalous behavior and eliminates the shortcomings of existing analogues, without requiring the intervention of a technical specialist, knowledge of the subject area and long-term training.

The graphic part consists of 8 posters.

In the economic section, after evaluating the information technology and the idea, it was determined that the development meets the level of sufficient novelty with an integral novelty value of 0.515, the payback period is 0.87 years, which indicates the commercial attractiveness of the scientific and technical development. The level of commercial development potential is 44.3 points (high), which indicates the commercial importance of conducting these studies.

Keywords: anomaly detection, machine learning, decision tree, HTTP request.

ЗМІСТ

ВСТУП

Актуальність. Сучасне життя напряму залежить від інтернету. Його значимість важко переоцінити, адже з ним пов’язане навчання, робота, комунікація, відпочинок. Інтернет дав можливість реалізувати безліч ідей та задумів, починаючи зі звичайного обміну даними на відстані. Він однозначно спростив людське життя та покращив ефективність будь-якої діяльності.

Для певного ряду ресурсів важливим є питання достовірності, конфіденційності та стабільності. Порушення їх стабільності, несанкціонований вплив на стан даних може вартувати дуже дорого з точки зору чесності, справедливості та порядності. У гірших випадках – порушення вищезазначеного може нанести пряму шкоду людським життям, як фізичну, так і моральну.

Будь-який веб-додаток, що відіграє серйозну роль, повинен бути захищеним. Що вагоміший вплив веб-додатку на ті чи інші процеси – тим вищі повинні бути вимоги до якості його захисту.

Зазвичай, атаки на інтернет-ресурси спрямовані з метою унеможливлення доступу користувачів, але існують атаки, мета яких – зміна або викрадення чутливих даних.

Поширений тип атаки – DoS-атака. Атака на відмову в обслуговуванні, розподілена атака на відмову в обслуговуванні (англ. DoS attack, DDoS attack, (Distributed) Denial-of-service attack) — напад на комп'ютерну систему з наміром зробити комп'ютерні ресурси недоступними користувачам, для яких комп'ютерна система була призначена.

Одним із найпоширеніших методів нападу є насичення атакованого комп'ютера або мережевого устаткування великою кількістю зовнішніх запитів (часто безглуздих або неправильно сформульованих) таким чином атаковане устаткування не може відповісти користувачам, або відповідає

настільки повільно, що стає фактично недоступним. Взагалі відмова сервісу здійснюється:

- примусом атакованого устаткування до зупинки роботи програмного забезпечення/устаткування або до витрат наявних ресурсів, внаслідок чого устаткування не може продовжувати роботу;
- заняттям комунікаційних каналів між користувачами і атакованим устаткуванням, внаслідок чого якість сполучення перестає відповідати вимогам.

Якщо атака відбувається одночасно з великої кількості IP-адрес, то її називають розподіленою (англ. Distributed Denial-of-Service — DDoS) [1]. Один із найпоширеніших способів реалізації такої атаки полягає у виконанні HTTP запитів.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою магістерської кваліфікаційної роботи є підвищення ефективності виявлення HTTP запитів з аномальною поведінкою.

Для досягнення поставленої мети необхідно розв'язати такі наступні завдання:

- проаналізувати предметну область, сучасний стан розвитку програмних продуктів, що виявляють запити з аномальною поведінкою;
- розглянути існуючі методи виявлення аномалій у HTTP запитах та моделі HTTP запитів, обґрунтувати вибір методу машинного навчання та моделі HTTP запиту, що задовільнять мету даної роботи;

– розробити модель HTTP запиту, обравши ключові особливості, що суттєво впливатимуть на ефективність машинного навчання;
– розробити структуру, алгоритми роботи програмного забезпечення;
– імплементувати програмне забезпечення;
– виконати тестування програмного забезпечення та проаналізувати отримані результати.

Об’єкт дослідження – процес виявлення аномалій у HTTP запитах.

Предмет дослідження – інформаційна технологія та програмні засоби виявлення HTTP аномалій у запитах.

Методи дослідження. У роботі використані методи та підходи до розробки штучного інтелекту, методи та підходи до розробки веб-орієнтованих програмних додатків, методи об’єктно-орієнтованого програмування.

Наукова новизна одержаних результатів полягає в наступному:

Набула подальшого розвитку інформаційна технологія для виявлення HTTP запитів з аномальною поведінкою, що відрізняється новою моделлю HTTP запиту, яку було використано для машинного навчання, що дозволило підвищити достовірність роботи програмних засобів розпізнавання та класифікації аномальних HTTP запитів.

Практичне значення одержаних результатів полягає у тому, що на основі проведених досліджень розроблено інформаційну технологію на базі мікросервісу, що аналізує HTTP запити на наявність аномалій.

Запропонована технологія відрізняється від аналогів високою точністю за рахунок керованого підходу до машинного навчання, відсутністю необхідності як втручання технічного спеціаліста, так і інтеграції. Зокрема, розроблено алгоритми роботи програмного забезпечення та інформаційну технологію для виявлення HTTP запитів з аномальною поведінкою.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень,

строгим виведенням аналітичних співвідношень, експериментальними дослідженнями тестування програмної реалізації інформаційної технології для виявлення HTTP запитів з аномальною поведінкою. Адекватність запропонованої моделі HTTP запиту підтверджується результатами експериментальних досліджень та виконаним процесом визначення важливості ознак.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, які написано у співавторстві, здобувачу належать: аналіз процесу виявлення HTTP запитів з аномальною поведінкою та методів підвищення точності [2].

Апробація результатів роботи. Результати роботи були апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН2023)», Вінниця, 15 листопада 2022 року - 12 травня 2023 року [2].

Публікації. За результатами дослідження опубліковано тези доповіді на науково-технічній конференції [2] та подано заяву на авторське право на програмне забезпечення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ ВИЯВЛЕННЯ HTTP ЗАПИТІВ З АНОМАЛЬНОЮ ПОВЕДІНКОЮ

1.1 Аналіз предметної області та сучасних аномалій у HTTP запитах

Комп’ютерна безпека – це безпека інформаційних систем (обладнання та програм).

Комп’ютерна безпека є важливою, оскільки урядові, військові, корпоративні, фінансові та медичні організації збирають, обробляють та зберігають безпрецедентні обсяги даних на комп'ютерах та інших пристроях. Значна частина цих даних може бути конфіденційною інформацією, будь то інтелектуальна власність, фінансові дані, особиста інформація або інші типи даних, для яких несанкціонований доступ або викриття можуть мати негативні наслідки. Організації передають конфіденційні дані через мережі та на інші пристрої в процесі ведення бізнесу. Компаніям та організаціям, особливо тим, яким доручається захищати інформацію, що стосується національної безпеки, охорони здоров'я чи фінансової документації, потрібно вживати заходів для захисту своєї конфіденційної інформації про бізнес та персонал. Вже в березні 2013 року представники розвідки країни застерегли, що кібератаки та цифрове шпигунство є головною загрозою національній безпеці, перевершуючи навіть тероризм [3].

Розглянемо деякі хакерські впливи на веб-додатки. Серед них, наприклад – DDoS, «фішинг», SQL-ін’єкції, «man-in-the-middle». Зазвичай, веб-додатки розробляють таким чином, щоб уникнути основних вразливостей. Наприклад, для REST архітектури використовують різні протоколи авторизації, такі як JWT чи OAuth/OAuth2. Від DoS атак захищаються, тестуючи та перевіряючи веб-додаток навантаженнями. Щоб шифрувати дані – використовують HTTP протокол з додатковим шаром шифрування/автентифікації між HTTP та TCP

[4]. Як висновок – важливо розробити серверну частину таким чином, щоб вона, щонайменше, вирішувала основні проблеми безпеки та «закривала» вразливі місця додатку.

Виявлення атак важливо для реагування на інциденти, обмеження шкоди від атак, переслідування зловмисника, стримування та запобігання шкоди.

Виявлення аномалій – це ідентифікація рідкісних елементів, подій або спостережень у даних, які значно відрізняються від очікуваних.

Виявлення аномалій пов'язано з аналізом великих обсягів корельованих даних і виявлення вразливостей якомога раніше з максимальною точністю. Випадки використання виявлення аномалій на різних ринках включають:

- Нафту і газ;
- Транспорт;
- Виробництво;
- Телекомунікації;
- Банківську справу;
- Страхування;
- Веб-бізнес;
- Електронну комерція.

У кожній із цих вертикалей послугу виявлення аномалій можна використовувати для виявлення небажаних бізнес-інцидентів і спостережень, надання величини аномалії як різниці між очікуваним і фактичним значенням. Це допомагає ідентифікувати аномалії в багатовимірному наборі даних, використовуючи переваги взаємозв'язку між сигналами. Коли сервіс будує модель машинного навчання для кожного сигналу як функціональний взаємозв'язок між сигналами, він максимізує точність виявлених аномалій [5].

1.2 Аналітичний огляд відомих технічних рішень виявлення аномалій у HTTP запитах

Розглянемо сервіси, що надають можливість аналізувати запити для захисту веб-ресурсів.

Anomaly – сервіс, що у режимі реального часу дозволяє відслідковувати аномалії, застосовуючи алгоритми, нейронні мережі, машинне навчання та математичні моделі. Використовуючи всі шаблони, отримані з показників, ми Anomaly виявляють незвичайну поведінку. Спостерігаючи за кожною точкою даних, яка надсилається у режимі реального часу, якщо спрацьовує щось незвичне, надсилається сповіщення. Механізм навчання дозволяє процесу виявлення стати більш-менш гнучким щодо аномалій залежно від того, скільки разів перевіряються прогнози. Коли прогнози часто правильні, виявлення дуже суворе щодо ініціювання сповіщень. Однак, якщо передбачення є правильним рідше, допускається більше гнучкості для відхилень до того, як буде створено сповіщення.

Звичайно, виявлення марне, якщо не отримується інформація, або якщо вона надходить надто пізно. Саме тому Anomaly сповіщають електронною поштою, текстовим повідомленням або за допомогою інших служб оповіщення [6].

Anomaly дозволяє інтегрувати свої можливості різними мовами програмування, виконувати моніторинг в режимі реального часу та будувати графіки.

Anomaly не спеціалізується на одній предметній області та дозволяє виявляти класичні формати аномалій – кореляції, рекурентні події, схожу поведінку, тренди.

Серед недоліків Anomaly – з невідомих причин на поточний момент неможливо зареєструватись та оформити оплату послуг. Anomaly, як зазначено, не спеціалізується виключно на виявленні аномалій, що відносяться до HTTP протоколу, тому потребує додатковий час на навчання та підготовку.

Через роботу у режимі реального часу, найбільш вірогідно, що використовуються менш точні за результатами, але швидші математичні моделі для постійного навчання та виявлення аномалій, що призводить до погіршення якості безпеки інтернет-ресурсу. Оплата за послуги відбувається не у форматі «pay as you go», що економить гроші для інтернет-гігантів, але може бути суттєво менш вигідною для менших інтернет-ресурсів. На рисунку 1.1 зображено сервіс «Anomaly».

Рисунок 1.1 – Сервіс Anomaly

Oracle дозволяє розробити власний проєкт, який буде контейнером для моделей, які необхідно створити та тренувати. Виявлення аномалій використовує два типи ядер машинного навчання, щоб вивчати шаблони та

виявляти аномалії з вашого багатовимірного набору даних. Один тип для багатофакторних корельованих сигналів і один для однофакторних незалежних сигналів.

Механізм ML, вбудований у службу, автоматично вибирає відповідне ядро та оптимізує параметри для створення найкращої моделі та результату. На рисунку 1.2 зображено схему роботи ресурсу Oracle.

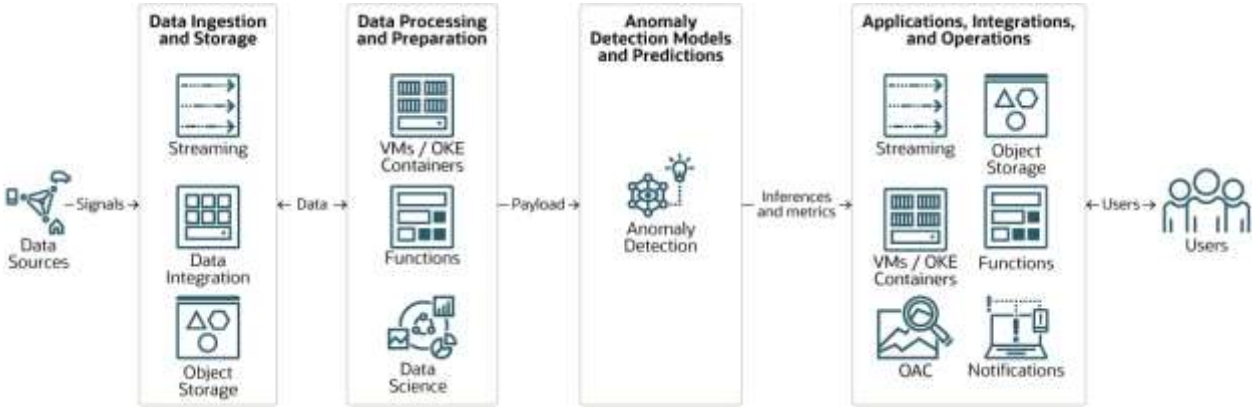
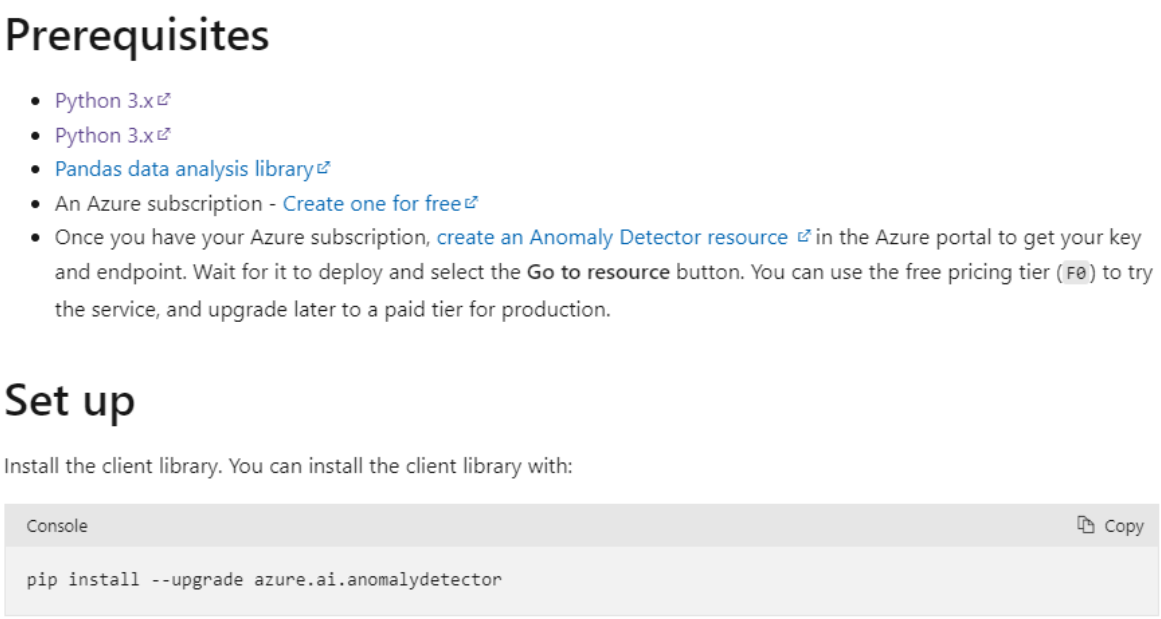


Рисунок 1.2 – Схема роботи ресурсу виявлення аномалій Oracle [7]

Відповідно, для роботи з Oracle необхідно створити самостійно умови для виявлення аномалій та зробити чимало роботи, яка потребує технічних знань та знань з предметної області. У результаті, кожного разу для отримання результату даних щодо аномалій, необхідно вручну завантажувати дані, після чого отримувати результат.

Azure пропонує власну бібліотеку, детектор аномалій, але вона також потребує власних реалізацій. Бібліотека реалізована для наступних мов програмування – C#, JavaScript, Python. На рисунку 1.3 зображено бібліотеку від Azure.



Prerequisites

- Python 3.x
- Python 3.x
- Pandas data analysis library
- An Azure subscription - Create one for free
- Once you have your Azure subscription, create an Anomaly Detector resource in the Azure portal to get your key and endpoint. Wait for it to deploy and select the **Go to resource** button. You can use the free pricing tier (`F0`) to try the service, and upgrade later to a paid tier for production.

Set up

Install the client library. You can install the client library with:

Console Copy

```
pip install --upgrade azure.ai.anomalydetector
```

Рисунок 1.3 – Бібліотека для виявлення аномалій від Azure

Siemens має власний API для виявлення аномалій, що входить у їх продукт – MindSphere. Рішення MindSphere відносяться до інтернету речей, а виявлення аномалій потребує втручання до створеної логіки, та конфігурацій, адже відноситься до секції для розробників. API встановлює певні ліміти на формат даних, які приймає. Навчання відбувається методом навчання без учителя, що, хоч і дає гнучкість, не є найдоречнішим для предметної області комп’ютерної безпеки, адже на сьогодні є велика кількість датасетів та інформації, на основі якої можливо чітко відділити запити з аномаліями від чистих, що дасть змогу отримати більш точні результати на виході. На рисунку 1.4 наведено приклад ендпоїнту MindSphere.

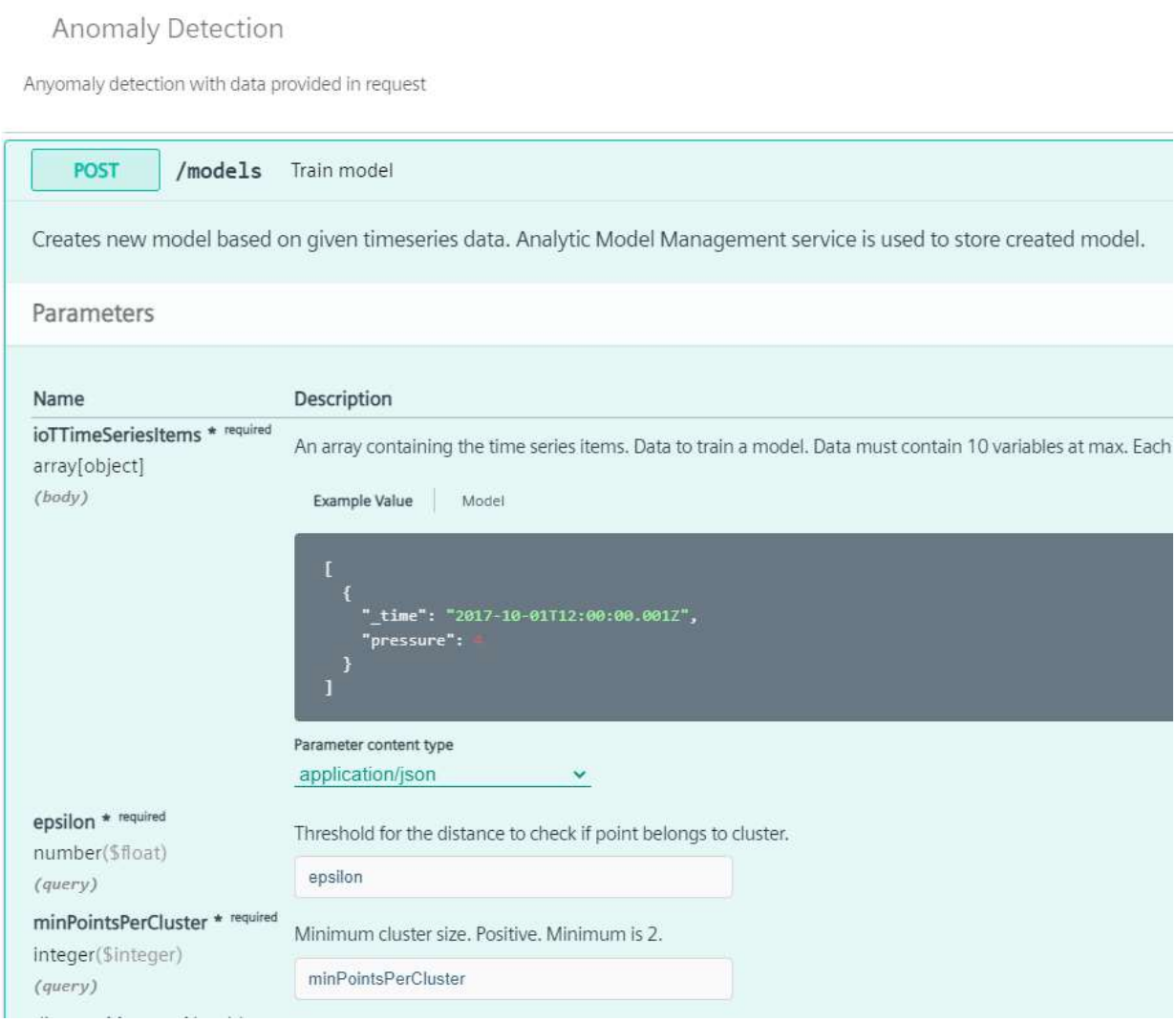


Рисунок 1.4 – Приклад ендпоїнту MindSphere API

Переваги та недоліки розглянутих сервісів наведено на таблиці 1.1.

Таблиця 1.1 – Переваги та недоліки розглянутих сервісів.

	Anomaly	Oracle	Azure	MindSphere
Конкретна область у виявленні аномалій	-	-	-	-
Необхідність технічного втручання	-	+	+	+
Доступність	-	+	+	+
Точність результату	-+	Залежить від методу		
Моніторинг	+	-	-	-
Об’єм даних та метрик	Лімітований	Лімітований	Лімітований	Великий

Anomaly є сервісом, котрий можливо зручно доєднати та почати роботу з ним, не втручаючись у технічну реалізацію та не думаючи про математичні моделі, на відміну від інших розглянутих сервісів. Він також дозволяє спостерігати за поточним станом у режимі реального часу. Тим не менш, Anomaly може потенційно поступатись точністю результатів, адже для режиму реального часу необхідно дуже швидко обробляти велику кількість даних, а існуючі математичні моделі, які дозволяють це зробити максимально швидко, поступаються якістю результату. Також Anomaly не надає можливість безоплатно спробувати його можливості і не працює по формату «pay as you go».

Загалом, усі розглянуті сервіси не спеціалізуються на тій чи іншій предметній області, їх перевага у гнучкості та навчанні без учителя, що дає їм змогу виявляти аномалії у різноманітних областях на високому рівні. Але відсутність спеціалізації створює необхідність у появі сервісу для виявлення аномалій, які відносяться саме до взаємодії з інтернет-ресурсом. Навіть одна невиявлена аномалія може нанести великої шкоди, тому висока точність та специфічність у предметній області може грати суттєву роль у забезпеченні якісного захисту.

1.3 Постановка задачі

Необхідно розробити інформаційну технологію, що спеціалізується на виявленні аномалій у запитах, що буде приймати безліч метрик та величезні обсяги даних для обробки та навчання. При розв’язанні задачі – використати математичну модель, яка демонструє найкращі результати на виході. Для вирішення проблеми технічного втручання клієнта або залежності від мови програмування, розробити автономний мікросервіс з підтримкою REST-стилю взаємодії між компонентами інтернет-ресурсів, використовуючи найсучасніші технології, щоб максимально пришвидшити його роботу та нівелювати втрати

у часі через точність моделі. Інтерфейс користувача повинен бути лаконічним та простим для сприйняття.

Результатом стане легкий та швидкодійний мікросервіс, що спеціалізується на високоточному виявленні аномалій у запитах.

1.4 Висновок

Проаналізовано предметну область, доведено актуальність виявлення аномалій у HTTP запитах.

Розглянуто технічні реалізації сервісів, що надають можливість аналізувати дані та виявляти у них аномалії, наведено переваги та недоліки кожного з них та встановлено, що кожен з них не має спеціалізації, через що може втрачати якість аналізу. Кожен із сервісів має ті чи інші недоліки, такі як необхідність у технічному втручанні, ліміти на обсяги даних та метрик, відсутність можливостей моніторингу.

У результаті аналізу доведено, що існує необхідність у створенні інформаційної технології, яка повинна спеціалізуватись на виявленні аномалій у запитах та повністю відповідати поставленій задачі.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ ВИЯВЛЕННЯ HTTP ЗАПИТІВ З АНОМАЛЬНОЮ ПОВЕДІНКОЮ

2.1 Аналіз методів виявлення аномалій

Аномалії можна визначити як спостереження, які достатньо відрізняються від більшості спостережень у наборі даних, щоб вважати, що вони були згенеровані іншим, ненормальним, генеративним процесом. Аномалія – це будь-яке спостереження, яке настільки сильно відрізняється від інших спостережень у наборі даних, що викликає підозру [8].

Виявлення аномалій – це метод, який використовується для виявлення того, що не відповідає нормальній поведінці набору даних. Іншими словами, виявлення аномалій знаходить точки даних у наборі даних, які відрізняються від решти даних.

Ці незвичайні речі називаються винятками, особливостями, сюрпризами тощо [9].

Аномалії можна класифікувати за такими трьома категоріями.

Точкові аномалії. Якщо один об’єкт можна спостерігати на тлі інших об’єктів як аномалію, то це точкова аномалія. Це найпростіша категорія аномалій, і їх включає багато досліджень.

Контекстуальні аномалії. Якщо об’єкт є аномальним у певному контексті. Тільки в цьому випадку це контекстна аномалія (також відома як умовна аномалія).

Колективні аномалії. Якщо деякі пов’язані об’єкти можна порівняти з іншими об’єктами як аномалію. Аномальним у цьому випадку не може бути окремий об’єкт, а лише сукупність об’єктів. На рисунку 2.1 та 2.2 наведено приклади типів аномалій.

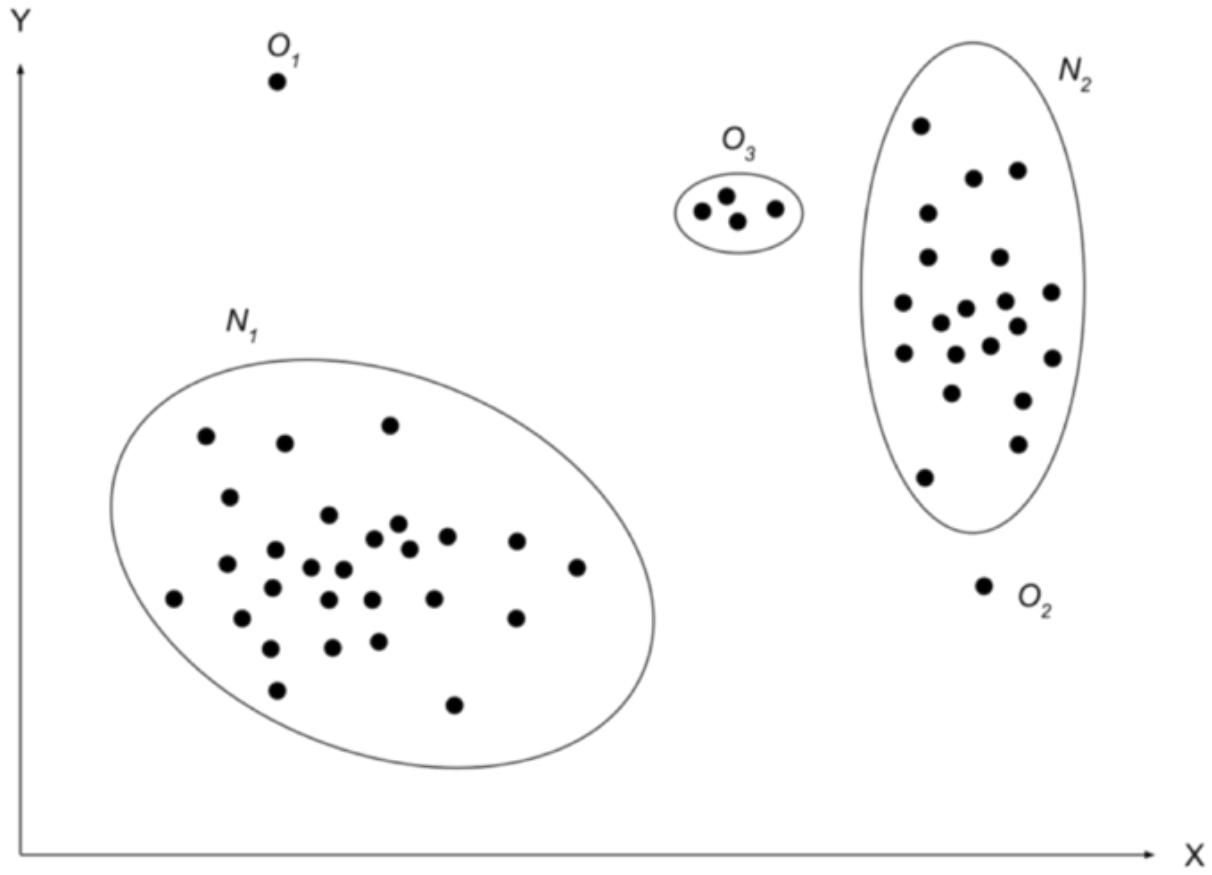


Рисунок 2.1 – Приклад колективних та точкових аномалій

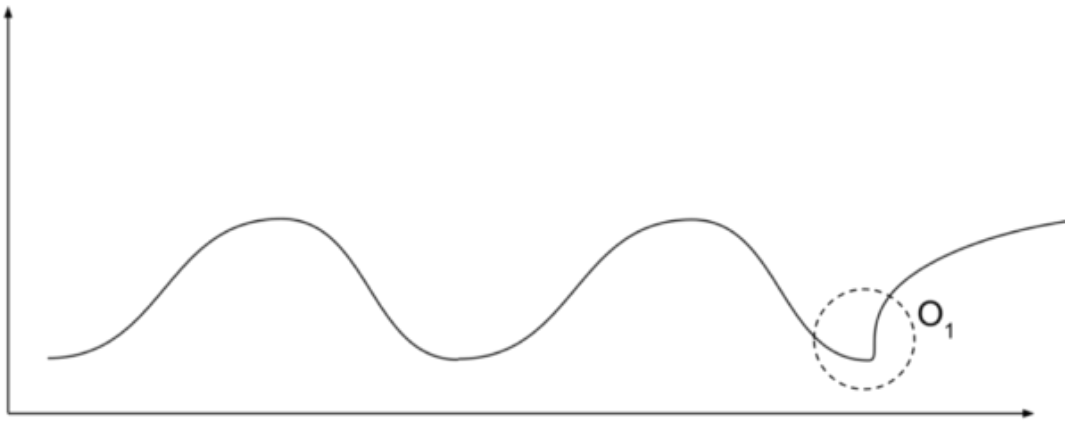


Рисунок 2.2 – Приклад контекстуальної аномалії

На абстрактному рівні виявлення аномалій здається простим завданням. Але це завдання може бути дуже складним.

Визначити нормальні регіони дуже складно. У багатьох випадках межі між аномаліями та нормальними даними не є точними. У цьому випадку нормальні спостереження можна вважати аномаліями і навпаки.

Якщо дія є зловмисною, як-от шахрайство, це вважається аномалією. Дуже часто зловмисники намагаються адаптувати свої дії до нормальної поведінки. І знову ж таки, завдання виявити аномалії в даному випадку не така проста.

Те, що сьогодні вважається нормальним, у майбутньому може стати ненормальним. Більшість бізнес-систем змінюються з часом під впливом різних факторів.

Підходи до виявлення аномалій в одній галузі частіше не можуть бути використані в іншій. У більшості випадків вони будуть неефективними.

Доступність даних навчання та перевірки для навчання моделі є великою проблемою [10].

Алгоритми виявлення аномалій даних низької розмірності не підходять для даних великої розмірності. Якщо d є розмірністю або атрибутом набору даних, високовимірний набір даних має $d > 10$ атрибутів. Дані великої розмірності погіршуються в результаті «прокляття розмірності». Загалом, викид або аномалію можна знайти за допомогою алгоритмів на основі відстані або щільності. Ці алгоритми вимірюють відстань між екземплярами даних і виходячи з припущення, що аномальна точка даних буде віддалена від інших точок даних, де виявлено аномалії. Однак у випадку великої розмірності дані стають розрідженими, і всі точки даних виглядають нормально. На основі алгоритмів можна класифікувати як контрольовані, напівконтрольовані та неконтрольовані. Коли відомі мітки даних як для нормального, так і для аномального, вони класифікуються як контрольовані. Якщо відома лише мітка даних нормального, це називають напівконтрольованими алгоритмами. Якщо мітки даних як нормальних, так і аномальних невідомі, це неконтрольований алгоритм [11].

Контрольоване виявлення аномалій. Налаштування, у якому дані позначені в наборах навчальних і тестових даних; коли простий класифікатор можна навчити та застосувати. Цей випадок схожий на традиційне розпізнавання образів, за винятком класів, які в більшості випадків сильно незбалансовані. Не всі підходи до класифікації підходять для цього завдання. Наприклад, деякі типи дерев рішень не можуть добре працювати з незбалансованими даними. Підтримуючі векторні машини (SVM) або штучні нейронні мережі (ANN) повинні працювати краще. Однак це налаштування не є актуальним, оскільки потрібно знати всі аномалії та правильно позначати дані. У багатьох випадках аномалії невідомі заздалегідь або можуть виникнути як новинки під час фази тестування. На рисунку 2.3 наведено схему контрольованого навчання.

Рисунок 2.3 – Контрольоване виявлення аномалій

Напівконтрольоване виявлення аномалій. Напочатку, коли знань немає, знання набуваються із результатів навчання. Ця установка також використовує навчальні та тестові набори даних, де лише навчальні дані складаються зі звичайних даних без будь-яких аномалій. Ідея полягає в тому, що модель нормального класу вже навчена, а аномалії можна виявити шляхом відхилення від вивченої моделі. Цей підхід також відомий як «однокласова» класифікація. Добре відомими підходами є однокласові SVM і автокодери. Загалом, будь-який підхід до оцінки щільності може бути застосований для моделювання функції щільності ймовірності нормальних класів, наприклад підходи суміші

Гауса або оцінка щільності ядра. На рисунку 2.4 наведено схему напівконтрольованого навчання.

Рисунок 2.4 – Напівконтрольоване навчання

Неконтрольоване виявлення аномалій. Налаштування, коли невідомо, що є нормальним у даних, а що ні. Це найбільш гнучка конфігурація, яка не потребує жодних міток. Також немає різниці між навчальним і тестовим набором даних. Концепція полягає в тому, що підходи до неконтрольованого виявлення аномалій оцінюють дані виключно на основі природних особливостей набору даних. Як правило, відстані або щільності використовуються для оцінки того, що є нормою, а що – викидом. Візуальне представлення можна знайти на малюнку 2.5.

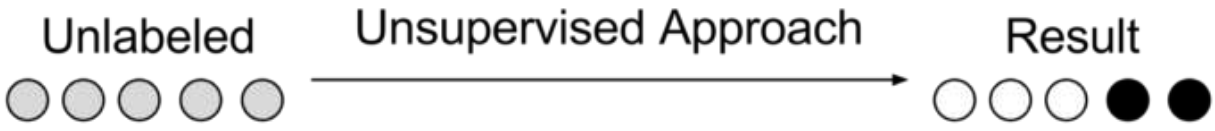


Рисунок 2.5 – Неконтрольоване виявлення аномалій [10]

Інтернет-ресурси функціонують вже досить довгий період часу. Існують великі набори даних, що вказують на ті чи інші аномалії. Це є великою перевагою, тому найдоречніше використати керований підхід до виявлення

аномалій. Кероване навчання доречне за умов прогностичної моделі, позначених даних, тобто відомо, які дані містять конкретні аномалії. Основні типи керованого навчання включають регресію та задачу класифікації.

2.2 Аналіз керованого навчання та математичне формулювання завдання

Навчання з учителем, контрольоване або кероване навчання (англ. Supervised learning) – один зі способів машинного навчання, в ході якого випробувана система примусово навчається за допомогою наявної множини прикладів «стимул-реакція» з метою визначення «реакції» для «стимулів», які не належать до наявної множини прикладів. З точки зору кібернетики, є одним із видів кібернетичного експерименту.

Між входами та еталонними виходами (стимул-реакція) може існувати деяка залежність, але вона не відома. Відома лише кінцева сукупність прецедентів — пар «стимул-реакція», звана навчальною вибіркою. На основі цих даних потрібно відновити залежність (побудувати модель відносин стимул-реакція, придатних для прогнозування), тобто побудувати алгоритм, здатний для будь-якого об'єкта видати досить точну відповідь. Для вимірювання точності відповідей, так само як і в навчанні на прикладах, може вводитися функціонал якості.

Формально задача навчання може бути сформульована наступним чином: існує деяка сукупність «стимулів» x і «реакцій на стимули» y, необхідно визначити залежність між y та x таку, що в межах припустимої помилки e буде справедливим $f(x) - e \leq y \leq f(x) + e$.

Можливо виділити наступні типи вхідних даних навчання з учителем.

– Ознаковий опис – найпоширеніший випадок. Кожен об'єкт описується набором своїх характеристик, що називаються ознаками. Ознаки можуть бути числовими або нечисловими.

– Матриця відстаней між об'єктами. Кожен об'єкт описується відстанями до всіх інших об'єктів навчальної вибірки. З цим типом вхідних даних працюють деякі методи, зокрема, метод k найближчих сусідів, метод парзенівського вікна, метод потенційних функцій.
– Часовий ряд або сигнал є послідовністю вимірювань в часі. Кожен вимір може представлятися числом, вектором, а в загальному випадку — ознаковим описом досліджуваного об'єкта в даний момент часу.
– Зображення або відеоряд.
– Зустрічаються й складніші випадки, коли вхідні дані подаються у вигляді графів, текстів, результатів запитів до бази даних тощо. Як правило, вони приводяться до першого або другого випадку шляхом попередньої обробки даних та виділяння ознак.

Існують наступні типи відгуків.

– Коли множина можливих відповідей нескінченна (відповіді є дійсними числами або векторами), говорять про задачі регресії та апроксимації;
– Коли множина можливих відповідей звичайна, говорять про задачі класифікації та розпізнавання образів;
– Коли відповіді характеризують майбутню поведінку процесу або явища, кажуть про задачі прогнозування [12].

Розглянемо задачу класифікації, яка є одним з типів відгуків.

Задача класифікації — формалізована задача, яка містить множину об'єктів (ситуацій), поділених певним чином на класи. Задана скінченна множина об'єктів, для яких відомо, до яких класів вони належать. Ця множина називається вибіркою. До якого класу належать інші об'єкти невідомо. Необхідно побудувати такий алгоритм, який буде здатний класифікувати довільний об'єкт з вихідної множини.

Класифікувати об'єкт — означає, вказати номер (чи назву) класу, до якого належить цей об'єкт.

Класифікація об'єкта — номер або найменування класу, що видається алгоритмом класифікації в результаті його застосування до цього об'єкта.

В математичній статистиці задачі класифікації називаються також задачами дискретного аналізу. В машинному навчанні завдання класифікації вирішується, як правило, за допомогою методів штучної нейронної мережі при постановці експерименту у вигляді навчання з учителем.

Математичне формулювання завдання виглядає наступним чином.

Нехай X – множина описів об'єктів, Y – множина номерів (чи назв) класів. Існує невідома цільова залежність- відображення $y^*: X \to Y$, значення якої відомі лише на елементах скінченної навчальної вибірки $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$. Потрібно побудувати алгоритм $a: X \to Y$, здатний класифікувати довільний об'єкт $x \in X$ [13].

Отож, для виявлення аномалій, необхідно використати класифікаційну модель.

2.3 Аналіз алгоритмів класифікаційної моделі

Розглянемо алгоритми, які доречно застосувати при класифікаційній моделі керованого навчання для виявлення аномалій у запитах.

Метод k-найближчих сусідів (англ. k-nearest neighbor method) — це непараметричний метод навчання з учителем, вперше розроблений Евеліном Фіксом та Джозефом Ходжесом у 1951 році, а пізніше розвинутий Томасом Ковером. Метод використовується як для класифікації, так і для регресії. В обох випадках вхідні дані складаються з k найближчих навчальних прикладів у наборі даних. Результат залежить від того, для чого використовується k-NN для класифікації чи регресії:

При класифікації k-NN результатом є належність класу. Об'єкт класифікується за допомогою множини голосів його сусідів, при цьому об'єкт відноситься до класу, найбільш поширеного серед його k найближчих сусідів (k — ціле додатне число, як правило, невелике). Якщо k = 1, то об'єкт просто приписується до класу цього єдиного найближчого сусіда.

k-NN – це тип класифікації, де функція локально лише апроксимується, а всі обчислення відкладаються до оцінки функції. Оскільки цей алгоритм покладається на функцію відстані для класифікації, то у випадку, коли ознаки представляють різні фізичні одиниці або мають дуже різні масштаби, то нормалізація навчальних даних може значно підвищити їх точність [14]. На рисунку 2.6 візуалізовано метод k-найближчих сусідів.

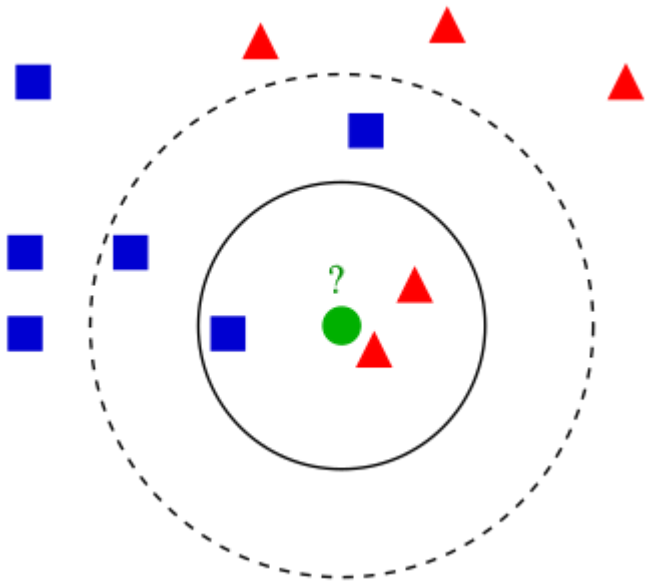


Рисунок 2.6 – Метод k-найближчих сусідів

У виявленні аномалій фактор локальних викидів (LOF) – це алгоритм, запропонований Маркусом М. Бройнігом, Гансом-Пітером Крігелем, Раймондом Т. Нгом і Йоргом Сандером у 2000 році для пошуку аномальних точок даних шляхом вимірювання локального відхилення даної точки даних. по відношенню до своїх сусідів.

Фактор локального викиду базується на концепції локальної щільності, де локальність задана k найближчими сусідами, відстань до яких використовується для оцінки щільності. Порівнюючи локальну щільність об’єкта з локальною щільністю його сусідів, можна ідентифікувати області з подібною щільністю та точки, які мають значно нижчу щільність, ніж їхні сусіди. Вони вважаються викидами.

Локальна щільність оцінюється типовою відстанню, на якій можна «дотягнутися» до точки від її сусідів. Визначення «відстані досяжності», що використовується в LOF, є додатковим заходом для отримання більш стабільних результатів у кластерах. «Відстань досяжності», яку використовує LOF, має деякі тонкі деталі, які часто виявляються неправильними у вторинних джерелах, наприклад, у підручнику Етема Алпайдіна.

Завдяки локальному підходу LOF здатний ідентифікувати викиди в наборі даних, які не були б викидами в іншій області набору даних. Наприклад, точка на «малій» відстані до дуже щільного кластера є викидом, тоді як точка в розрідженому кластері може демонструвати подібні відстані до своїх сусідів.

Хоча геометрична інтуїція LOF застосовна лише до векторних просторів низької розмірності, алгоритм можна застосовувати в будь-якому контексті, де можна визначити функцію неподібності. Експериментально було показано, що він дуже добре працює в численних налаштуваннях, часто перевершуючи конкурентів, наприклад, у виявленні вторгнень у мережу та на оброблених даних класифікаційного тесту. На рисунку 2.7 наведено візуалізацію оцінок LOF.

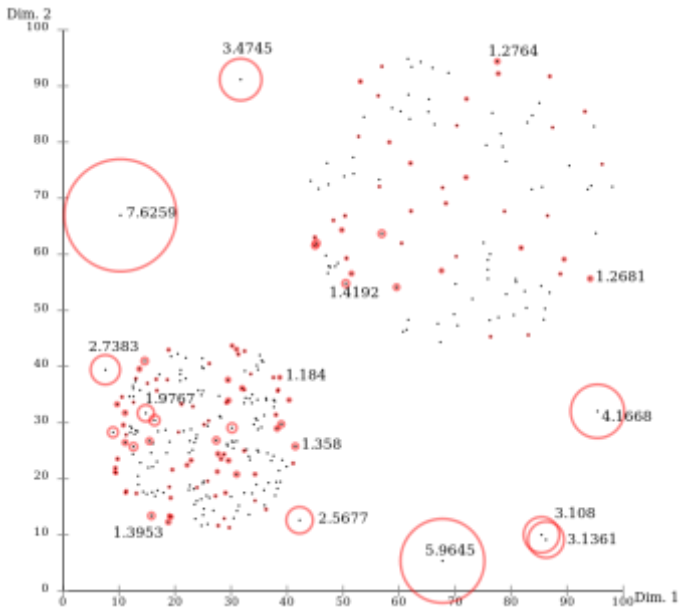


Рисунок 2.7 – Оцінки LOF, візуалізовані ELKI

Отримані значення є частками, і їх важко інтерпретувати. Значення 1 або навіть менше вказує на явне врізне значення, але немає чіткого правила, коли точка є викидом. В одному наборі даних значення 1,1 вже може бути викидом, в іншому наборі даних і параметризації (із сильними локальними коливаннями) значення 2 все ще може бути викидом. Ці відмінності також можуть виникати всередині набору даних через локальність методу.

Існують розширення LOF, які намагаються покращити LOF у цих аспектах [15]. Серед них, до методів, які задовільнять поставлену задачу, відноситься LoOP. Розглянемо його.

Local Outlier Probability (LoOP) – це метод, отриманий від LOF, але з використанням недорогої локальної статистики, щоб стати менш чутливим до вибору параметра k. Крім того, отримані значення масштабуються до діапазону значень [0:1] [15].

Оцінка викиду кожного зразка називається ймовірністю локального викиду. Він вимірює локальне відхилення щільності даної вибірки по відношенню до її сусідів як локальний фактор викиду (LOF), на якому він базується, але надає нормалізовані оцінки викидів у діапазоні [0,1]. Ці показники викидів можна безпосередньо інтерпретувати як ймовірність того, що об’єкт є викидом. Оскільки Local Outlier Probabilities надає бали в діапазоні [0,1], результати легко інтерпретувати відповідно до програми.

Як і LOF, він є локальним, оскільки оцінка аномалії залежить від того, наскільки ізольований зразок відносно навколишнього середовища. Локальність визначається k-найближчими сусідами, чия відстань використовується для оцінки локальної щільності. Порівнюючи локальну щільність зразка з локальними щільностями його сусідів, можна ідентифікувати зразки, які лежать в регіонах з нижчою щільністю порівняно з їхніми сусідами, і таким чином ідентифікувати зразки, які можуть бути викидами відповідно до їхньої ймовірності локального викиду.

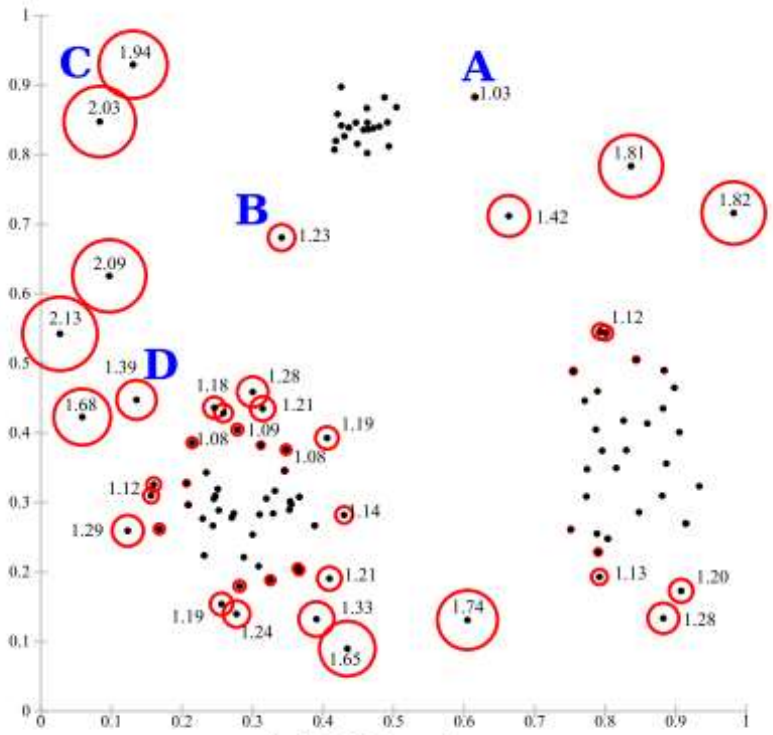
Застосовуючи LoOP до даних, інтуїтивно зрозуміло очікувати, що значення вздовж країв поля розсіювання будуть визначені як більш імовірні

викиди, оскільки вони виходять за межі типових значень у розподілі даних. Два параметри — екстент і кількість сусідів – безпосередньо впливають на отримані оцінки LoOP.

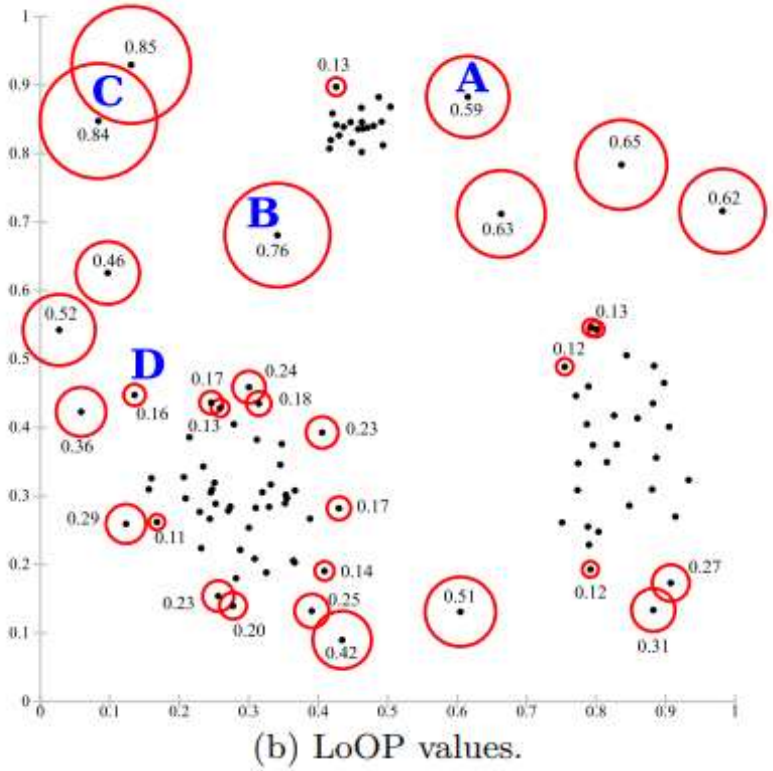
Параметр ступеня контролює чутливість оцінки на практиці. Параметр відповідає статистичному поняттю викиду, визначеного як об’єкт, що відхиляється більше, ніж дана лямбда (ступінь), помножена на стандартне відхилення від середнього. Значення 2 означає викиди, що відхиляються більше ніж на 2 стандартні відхилення від середнього, і відповідає 95,0% за емпіричним правилом «трьох сигм».

Параметр n_neighbors, з іншого боку, визначає кількість сусідів, які слід враховувати щодо кожної вибірки (розмір околиці) під час визначення її локальної ймовірності викиду щодо щільності визначеної околиці вибірки. Ідеальна кількість сусідів, які слід враховувати, залежить від вхідних даних. Однак поняття викиду означає, що воно буде вважатися таким незалежно від кількості розглянутих сусідів. Одним з потенційних підходів є використання кількох різних розмірів сусідства та усереднення результатів для кожного спостереження. Ті спостереження, які мають високий рейтинг із змінними розмірами околиць, більш ніж вірогідно викидаються. Інший варіант – вибрати значення, пропорційне кількості спостережень, таке непарне ціле число, близьке до квадратного кореня з кількості спостережень у ваших даних. Ідеальний вибір, знову ж таки, залежить від даних і їх характеристик.

Local Outlier Probabilities (LoOP) — це корисний підхід, який слід знати та застосовувати в певних контекстах, і з тих пір його було розширено для роботи в конкретних випадках використання, наприклад із потоковими даними (розширення, доступне в PyNomaly, разом з іншими прикладами LoOP на практиці). Цей підхід може допомогти виявити аномалії, визначені відповідно до їх придатності в рамках багатовимірного розподілу набору даних, і часто є чудовою відправною точкою для виявлення менш складних викидів даних (аномалій) практичним, програмованим і інтерпретованим способом [15]. На рисунку 2.8 візуалізовано порівняння LOF та LOOP.



(a) LOF values.



(b) LoOP values.

Рисунок 2.8 – Порівняння LOF та LoOP [17]

Щоб вдало обрати модель для машинного навчання, необхідно порівняти алгоритми. Доступний широкий спектр алгоритмів навчання під наглядом, кожен із яких має свої сильні та слабкі сторони. Не існує єдиного алгоритму

навчання, який найкраще працює з усіма проблемами навчання під наглядом (див. Теорему про відсутність безкоштовного обіду). Розглянемо цю теорему.

В обчислювальних процесах є обставини, за яких усі методи розв'язку однієї задачі виявляються статистично ідентичними. Мальовничим способом представлення таких обставин, запропонованим Девідом Вульпертом та Вільямом Дж. Маккріді в зв'язку з проблемами пошуку та оптимізації, є вислів — «Безкоштовних сніданків не існує». Перед цим Вульперт вивів цю теорему для машинного навчання (статистичного висновування). Перед публікацією роботи Вульперта, Каллен Шаффер підсумував переддруковану версію роботи Вульперта, але використав іншу термінологію.

В розгорненні метафори про безкоштовні сніданки, кожен «ресторан» (процедура, що розв'язує проблему) має «меню», асоціюючи кожну «страву» (проблему) з ціною (продуктивністю процедури, що вирішує цю проблему). Меню ресторанів ідентичні за винятком однієї деталі — ціни перемішані від одного ресторану до іншого. Для всеядної істоти, яка буде куштувати будь-яку страву, середня вартість обіду не залежить від вибору ресторану. Але вегетаріанець, що регулярно снідає разом з плотоядним та бажає заощадити, платить високу середню ціну за один сніданок. Для поступового зниження середньої вартості, потрібно мати поглиблені знання про a) власне саме замовлення та b) вартість цієї страви в усіх ресторанах. Таким чином, підвищення продуктивності розв'язання проблем потребує використання попередньої інформації для зіставлення процедур і проблем.

«Теорема про відсутність безкоштовних сніданків Вульперта та Маккріді», як дослівно зазначено самими Вульпертом і Маккріді, полягає в тому, що «продуктивність будь-яких двох алгоритмів у середньому однакова при розв'язанні всіх можливих задач». Наслідки відсутності безкоштовного сніданку свідчать про те, що підбір специфічного алгоритму для кожної проблеми дає в середньому кращі результати, ніж застосування одного й того самого алгоритму до всієї множини задач. Айгель, Туссон та Інглиш уклали спільну угодну про відсутність безкоштовного сніданку взагалі. Вважаючи,

що фізично це можливо, теорема не завжди виконується точно. Дросте, Джансес та Вегенер довели теорему, яку вони інтерпретують як «Безкоштовних сніданків (майже) не існує», на практиці.

Більш формально, задача — цільова функція, що співставляє допустимий розв'язок з деякою якісною оцінкою. Алгоритм пошуку приймає на вхід цільову функцію та один по одному обчислює допустимі розв'язки. Результатом роботи алгоритму є послідовність знайдених якісних оцінок.

Зручний, але не дуже точний підхід до інтерпретації наслідків теореми полягає в тому, що «універсальна глобальна оптимізація теоретично неможлива, і єдиний шлях для однієї стратегії бути продуктивніше за іншу — це бути спеціально налаштованою для розв'язання певного класу задач» [18].

2.4 Висновок

Розглянуто та проаналізовано методи виявлення аномалій у HTTP запитах. Кероване навчання визначено як доречний метод через наявність позначених даних.

Серед методів керованого навчання проаналізовано та обрано метод класифікації через необхідність визначати різні типи аномалій.

Серед методів класифікації проаналізовано моделі, які доречно застосувати при виявленні аномалій при керованому навчанні, маючи великий обсяг даних.

Проаналізовано теорему про відсутність безкоштовних сніданків Вульперта та Маккріді. Теорема вказує на необхідність обрання методів та алгоритмів у залежності від конкретної області та її специфіки, що підкреслює актуальність та доречність даної роботи.

Зважаючи на теорему, найдоречніше обрати ту модель, результати роботи якої будуть найкращими, а отже – кінцеву модель буде обрано після оцінки результатів роботи усіх моделей.

3 РОЗРОБКА МОДЕЛІ ДЛЯ ВИЯВЛЕННЯ HTTP ЗАПИТІВ З АНОМАЛЬНОЮ ПОВЕДІНКОЮ

3.1 Загальний огляд набору даних NF-UNSW-NB15

Для будь-якої роботи з даними у напрямку машинного навчання, аналізу тощо, необхідно виконати ряд дій, що відносяться до підготовки даних. Це може займати навіть до 80% робочого часу фахівця з області науки про дані. У цій роботі використано NF-UNSW-NB15 – формат набору даних на основі NetFlow під назвою NF-UNSW-NB15, що був розроблений і позначений відповідними категоріями атак. Загальна кількість потоків даних становить 1 623 118, з яких 72 406 (4,46%) є зразками атак, а 1 550 712 (95,54%) є доброякісними. Зразки атак далі класифікуються за дев'ятьма підкатегоріями. Таблиця нижче представляє розподіл усіх потоків за набором даних NF-UNSW-NB15. У таблиці 3.1 наведено опис атак з набору даних.

Таблиця 3.1 – Опис атак з датасету NF-UNSW-NB15 [19]

Клас	Кількість	Опис
Доброякісні	1550712	Нормальні
Фуззери	19463	Атака, під час якої зловмисник надсилає великі обсяги випадкових даних, що спричиняє збій системи, а також має на меті виявити вразливі місця системи безпеки.
Аналіз	1995	Група, яка представляє різноманітні загрози, націлені на веб-програми через порти, електронні листи та сценарії
Бекдор	1782	Техніка, спрямована на обхід механізмів безпеки, відповідаючи на конкретні створені клієнтські програми

Продовження таблиці 3.1.

DoS	5051	Відмова в обслуговуванні - це спроба перевантажити ресурси комп'ютерної системи з метою запобігання доступу або доступності її даних
Експлоїти	24736	Це послідовності команд, які контролюють поведінку хоста через відому вразливість
Дженеріки	5570	Метод, націлений на криптографію та викликає зіткнення з кожним блоковим шифром
Зондування	12291	Техніка для збору інформації про хост мережі, також відома як зондування
Шеллкод	1365	Зловмисне програмне забезпечення, яке проникає в код для керування хостом жертви
Черви	153	Атаки, які тиражуються та поширюються на інші комп’ютери

На рисунку 3.1 зображено графік за типами атак.

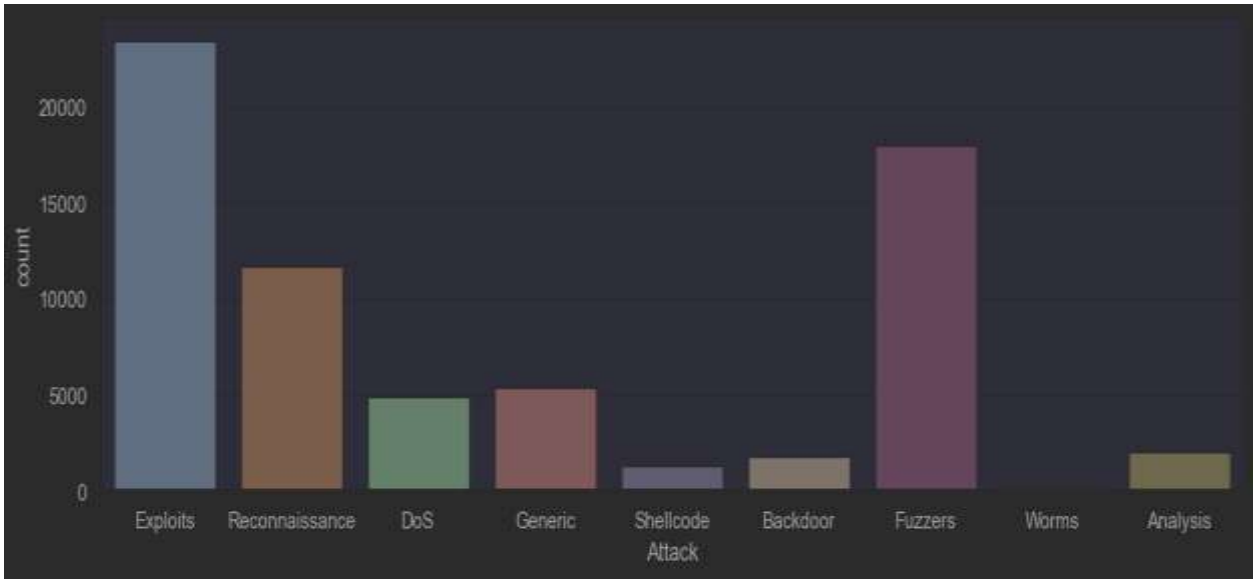
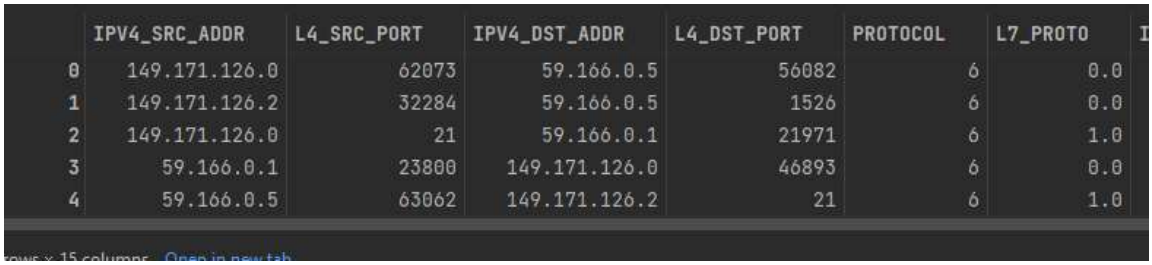


Рисунок 3.1 – Загальний графік атак, що присутні у наборі даних

Розглянемо приклад даних з цього датасету, що зображено на рисунку 3.2.



	IPV4_SRC_ADDR	L4_SRC_PORT	IPV4_DST_ADDR	L4_DST_PORT	PROTOCOL	L7_PROTO
0	149.171.126.0	62073	59.166.0.5	56082	6	0.0
1	149.171.126.2	32284	59.166.0.5	1526	6	0.0
2	149.171.126.0	21	59.166.0.1	21971	6	1.0
3	59.166.0.1	23800	149.171.126.0	46893	6	0.0
4	59.166.0.5	63062	149.171.126.2	21	6	1.0

Рисунок 3.2 – Приклад даних датасету

У таблиці 3.2 наведено усі стовбці та їх типи даних.

Таблиця 3.2 – Стовбці та їх типи даних

IPV4_SRC_ADDR	object
L4_SRC_PORT	int64
IPV4_DST_ADDR	object
L4_DST_PORT	int64
PROTOCOL	int64
L7_PROTO	float64
IN_BYTES	int64
OUT_BYTES	int64
IN_PKTS	int64
OUT_PKTS	int64
TCP_FLAGS	int64
FLOW_DURATION_MILLISECONDS	int64
Label	int64
Attack	object
Dataset	object

Розглянемо кореляцію між змінними набору даних. У статистиці кореляція або залежність – це будь-який статистичний взаємозв'язок, причинний чи ні, між двома випадковими змінними або двовимірними даними. У найширшому сенсі кореляція – це будь-яка статистична пов'язаність, хоча насправді вона стосується ступеню лінійності взаємозв'язку пари змінних. До добре відомих прикладів залежних явищ належать кореляція між зростом батьків та їхніх нащадків, а також кореляція між ціною товару та кількістю, яку споживачі готові придбати, як це зображено на так званій кривій попиту.

Кореляції корисні, бо вони можуть вказувати на передбачальний зв'язок, який можливо використовувати на практиці [20].

Кореляція між змінними набору даних зображена на рисунку 3.3.

Рисунок 3.3 – Кореляція між змінними набору даних

3.2 Розробка моделі для виявлення HTTP запитів з аномальною поведінкою

Модель досягається шляхом підготовки даних та виділенням ознак, що є найбільш вагомими у виявленні аномалій. Першим етапом є масштабування ознак.

Масштабування ознак – це техніка стандартизації незалежних ознак, присутніх у даних у фіксованому діапазоні. Це виконується під час попередньої обробки даних для обробки дуже різноманітних величин, значень або одиниць. Якщо масштабування функції не виконується, тоді алгоритм машинного навчання має тенденцію зважувати більші значення, вищі та вважати менші значення меншими значеннями, незалежно від одиниці значень.

Розглянемо два найважливіших методи виконання масштабування:

Мінімально-максимальна нормалізація: ця техніка повторно масштабує функцію або значення спостереження зі значенням розподілу між 0 і 1, що виражена формулою 3.1.

$$X_{new} = \frac{X_i - \min(X)}{\max(x) - \min(X)} \tag{3.1}$$

Стандартизація: це дуже ефективний метод, який повторно масштабує значення ознаки таким чином, щоб воно мало розподіл із середнім значенням 0 і дисперсією, що дорівнює 1, виражена формулою 3.2 [21].

$$X_{new} = \frac{X_i - X_{mean}}{\text{Standard Deviation}} \tag{3.2}$$

Наступним етапом є кодування атрибутів, що не є числовими. На цьому етапі нечислові дані стають даними з цілочисельним форматом для того, щоб моделі мали можливість працювати з ними.

Після кодування атрибутів настає етап обирання ознак.

В машинному навчанні та статистиці обирання ознак, відоме також як обирання змінних, обирання атрибутів та обирання підмножини змінних – це процес обирання підмножини доречних ознак (змінних, провісників) для використання в побудові моделі.

Методики обирання ознак застосовують із декількома цілями:

- спрощення моделей, щоби зробити їх легшими для інтерпретування дослідниками/користувачами;
- скорочення тривалості тренування;
- уникання прокляття розмірності;
- покращення узагальнення шляхом зниження перенавчання (формально, зниження дисперсії).

Центральною передумовою при застосуванні методики обирання ознак є те, що дані містять деякі ознаки, що є або надлишковими, або недоречними, і тому їх може бути усунено без спричинення значної втрати інформації. «Надлишкові» та «недоречні» є двома різними поняттями, оскільки одна доречна ознака може бути надлишковою в присутності іншої доречної ознаки, з якою вона сильно корелює [22].

Надалі – набори даних випадковим чином розділяються на підмножини і до них починають застосовувати моделі навчання.

Кінцевою моделлю для виявлення HTTP запитів з аномальною поведінкою можливо вважати саме ту модель, що залишилась після етапу обирання ознак, а тому її остаточний вигляд – список, поданий на рисунку 3.4, а графік важливості ознак зображено на рисунку 3.5.

```
8  feature_map = [(i, v) for i, v in itertools.zip_longes
9  selected_features = [v for i, v in feature_map if i ==
10
11 selected_features
```

Out 61

```
['L4_SRC_PORT',
 'L4_DST_PORT',
 'L7_PROTO',
 'IN_BYTES',
 'OUT_BYTES',
 'OUT_PKTS',
 'TCP_FLAGS',
 'FLOW_DURATION_MILLISECONDS',
 'IPV4_SRC_ADDR',
 'IPV4_DST_ADDR']
```

Рисунок 3.4 – Остаточний вигляд моделі для машинного навчання

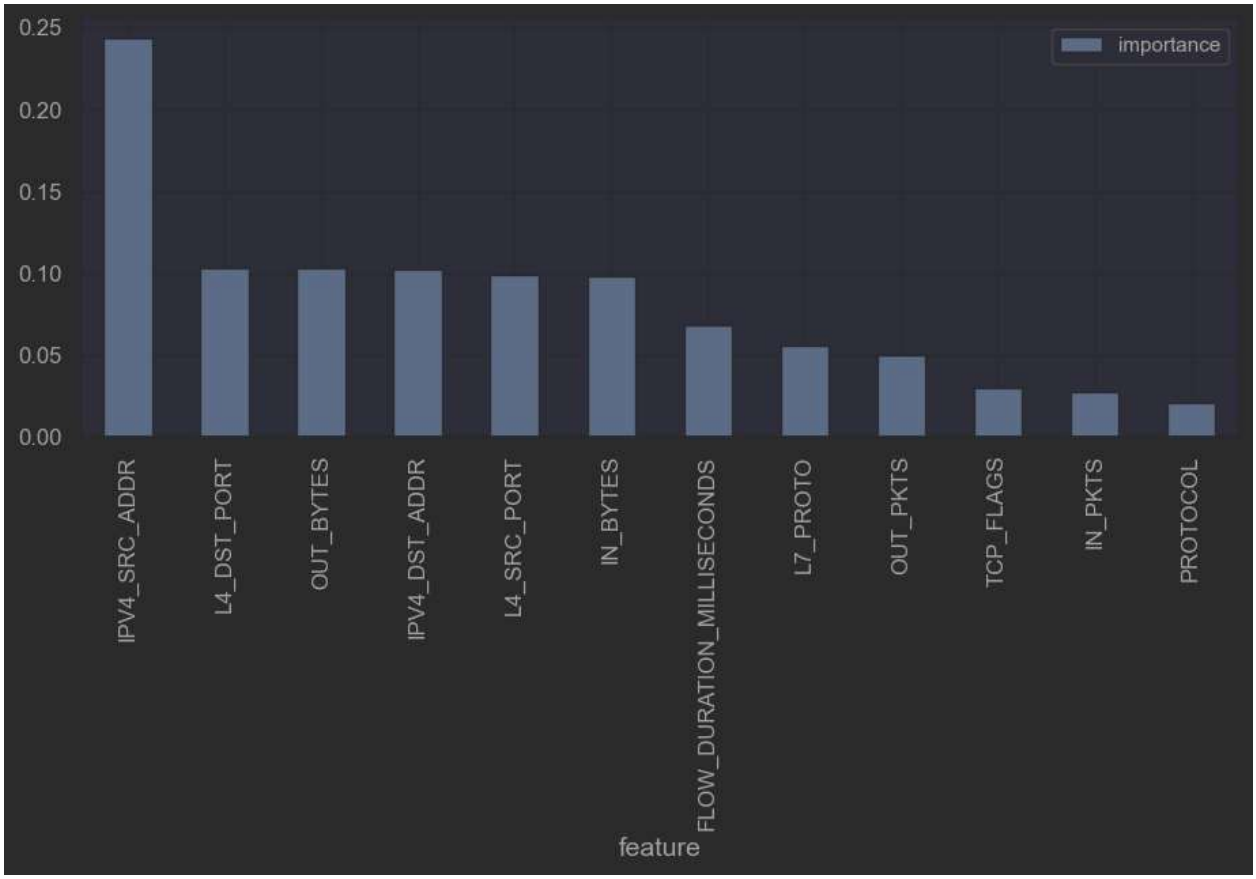


Рисунок 3.5 – Графік важливості ознак для обраної моделі

3.3 Висновок

Розглянуто та проаналізовано набір даних NF-UNSW-NB15.

Досліджено та продемонстровано кореляції між змінними, проаналізовано процес підготовки даних до машинного навчання.

Визначено головні ознаки та запропоновано власну модель, що буде використана під час машинного навчання.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ ВИЯВЛЕННЯ HTTP ЗАПИТІВ З АНОМАЛЬНОЮ ПОВЕДІНКОЮ

4.1 Обґрунтування вибору програмних засобів та мови програмування для реалізації веб-додатку

Для імплементації інформаційної технології, що дає змогу оптимально виконувати машинне навчання з зазначеними моделями у розділі 2, необхідно обрати таку мову програмування та технології, що вже використовуються у цій області. Такий підхід вибору дає можливість відштовхуватись лише від стабільних програмних засобів, які підтримуються як розробниками цих технологій, так і товариством розробників-користувачів.

Для реалізації мікросервіса необхідно обрати таку мову програмування та інструментарії, що підтримують об’єктно-орієнтоване програмування та надають можливість створити RESTful додаток. REST надасть можливість підтримки однорідного інтерфейсу, що надасть гнучкість при розробці. Не менш важливою є як надійність, так і портативність. REST приводить архітектуру додатку до клієнт-серверної.

Існує велика кількість мов програмування, що підтримують ООП. Кожна мова програмування виконує певні задачі краще за інші, відповідно, маючи свої переваги та недоліки. Все більше набирають оберти та використовуються такі мови як Java, C#, JavaScript, Python.

Популярними мовами для програмування як веб-додатків, так і скриптів, що відносяться до них, часто виступають JavaScript та Python. Якщо розглядати JavaScript, то його найпопулярнішим застосуванням є створення сценаріїв веб-сторінок. Завдяки чому на стороні клієнта (пристрої кінцевого користувача) є можливість взаємодіяти з користувачем, наприклад керувати браузером, адаптувати зовнішній вигляд веб-сторінки та її структуру, в

асинхронному режимі обмінюватися даними з серверною частиною, яку можна реалізувати за допомогою Node.js та Express.js.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією [23].

Node.js – платформа з відкритим кодом, основною ціллю якої є виконання високоефективних мережевих застосунків, що написані мовою JavaScript. До появи Node.js, Javascript застосовувався для виконання скриптів на стороні користувача. Та зараз, у JavaScript присутня можливість обробляти дані на стороні сервера, а їх результат відправляти користувачеві. Завдяки платформі Node.js JavaScript перетворився на мову загального використання з великою спільнотою поціновувачів та розробників [24].

Express.js – програмний каркас розробки серверної частини веб-застосунків для Node.js, реалізований як вільне і відкрите програмне забезпечення під ліцензією MIT. Він спроектований для створення веб-застосунків і API. Де-факто є стандартним каркасом для Node.js [25].

Використовуючи JavaScript та його інструментарії, для створення повноцінного веб-додатку популярним та оптимальним рішенням є використання MEAN стеку розробки.

MEAN (абревіатура від MongoDB, Express.js, Angular.js, Node.js) – комплексний набір серверного програмного забезпечення, що застосовується для веб-розробки. У MEAN-додатків одночасно серверна та клієнтська частини можуть бути написані мовою програмування JavaScript.

Значуща особливість MEAN – це поява можливості перенесення ядра реалізації MVC зі сторони сервера на сторону клієнта, завдяки чому відбувається перехід від генерації багатьох веб-сторінок на стороні сервера до створення переважно односторінкових додатків. Такий перехід забезпечує включення в склад стеку AngularJS (причому включений в склад фреймворк Express.js забезпечує і традиційну маршрутизацію і генерацію сторінок на стороні сервера) [26].

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду.

Python має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типів, а також те, що це інтерпретована мова, роблять її ідеальною для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ [27].

Популярними рішеннями для створення веб-додатків на основі Python є фреймворки Django, Flask, FastAPI.

Django – високорівневий відкритий Python-фреймворк (програмний каркас) для розробки веб-систем. Сайт на Django будується з однієї або декількох частин, які рекомендується робити модульними. Архітектура Django подібна на «Модель-Вигляд-Контролер» (MVC). Django підтримує парадигму ООП. Об'єкти БД в термінології Django іменуються «моделями». Фреймворк надає у розпорядження розробникові розвинутий прикладний програмний інтерфейс для високорівневого доступу до даних. У складі Django присутній власний веб-сервер для розробки і налагоджування. Сервер автоматично відслідковує зміни у файлах сирцевого коду і перезапускається, що зручно при розробці проєкту [28].

Flask – мікрофреймворк для веб-додатків, створений з використанням Python. Flask називається мікрофреймворком, оскільки він не вимагає спеціальних засобів чи бібліотек. У ньому відсутній рівень абстракції для роботи з базою даних, перевірки форм або інші компоненти, які надають широковживані функції за допомогою сторонніх бібліотек [29].

Через особливість MEAN стеку, що переносить ядро реалізації MVC архітектури зі сторони сервера на сторону клієнта, перевагу надано мові

програмування Python. Вказані переваги мови Python зміцнюють рішення обрати саме його, адже вони відповідають потребам та здатні розв’язати поставлені задачі витонченим шляхом.

В якості серверної частини обрано FastAPI. FastAPI – це сучасний, швидкий (високопродуктивний) веб-фреймворк для створення API з Python 3.6+ на основі стандартних підказок типу Python.

Основними характеристиками є:

- Швидкість: дуже висока продуктивність, нарівні з NodeJS і Go (завдяки Starlette і Pydantic). Один з найшвидших доступних фреймворків.
- Швидке кодування: швидкість розробки функцій пришвидшується приблизно на 200-300%.
- Менше помилок: кількість помилок зменшується приблизно на 40%, спричинених людиною (розробником).
- Інтуїтивність: чудова підтримка редактора. Скрізь завершення. Менше часу на налагодження.
- Легкість: розроблено, щоб бути простим у використанні та навчанні. Менше часу на читання документів.
- Лаконічність: мінімізує дублювання коду. Кілька функцій із кожного оголошення параметра. Менше помилок.
- Надійність: швидкий, готовий до релізу код з автоматичною інтерактивною документацією.
- На основі стандартів: на основі (і повністю сумісний) відкритих стандартів для API: OpenAPI (раніше відомих як Swagger) і JSON Schema [30].

Існують різні середовища розробки, що підтримують Python. Найпотужнішим та найпопулярнішим з них є PyCharm, через що його і обрано.

PyCharm пропонує великий набір інструментів «з коробки»: вбудований відладчик і інструмент запуску тестів, профілювальник Python, повнофункціональний вбудований термінал, інструменти для роботи з базами

даних. IDE інтегрована з популярними системами контролю версій, містить вбудований SSH-термінал тощо. Надає повноцінну підтримку різних веб-фреймворків і платформ для розробки на Python, підтримує темплейтні мови цих фреймворків, а також JavaScript, CoffeeScript, TypeScript, HTML / CSS, AngularJS, Node.js і багато інших [31].

Існують дві версії PyCharm – Professional та Community. Студенти мають можливість отримати усі професійні продукти від JetBrains, у тому числі PyCharm. Інструменти для веб-розробки та підтримки бази даних доступні лише у професійній версії.

PyCharm це середовище, що створено саме для реалізації програмного забезпечення мовою Python та інструментами, з якими вона працює, а отже, має найбільшу сумісність з обраними інструментами та технологіями в ході дослідження.

Celery — це асинхронна черга завдань або черга завдань із відкритим кодом, яка базується на розподіленій передачі повідомлень. Хоча він підтримує планування, він зосереджений на операціях у режимі реального часу.

Блоки виконання, які називаються завданнями, виконуються одночасно на одному або кількох робочих вузлах за допомогою багатопроцесорної обробки, подій або gevent. Завдання можуть виконуватися асинхронно (у фоновому режимі) або синхронно (чекати, доки вони будуть готові). Селера використовується у виробничих системах, для таких сервісів, як Instagram, для виконання мільйонів завдань щодня [32].

Оскільки понад 8,2 мільйона розробників у всьому світі використовують Python, Python посідає перше місце в останньому щорічному рейтингу популярних мов програмування IEEE Spectrum із результатом 100. Дедалі більше впровадження машинного навчання в усьому світі є головним фактором, що сприяє його зростанню популярності. Існує 69% інженерів машинного навчання, і Python став улюбленим вибором для аналізу даних, науки про дані, машинного навчання та штучного інтелекту – і все це завдяки

його величезній бібліотечній екосистемі, яка дозволяє практикам машинного навчання отримувати доступ, обробляти, перетворювати та обробляти дані з легкістю. Python завойовує серця інженерів машинного навчання своєю незалежністю від платформи, меншою складністю та кращою читабельністю [33].

4.2 Обґрунтування вибору програмних засобів та мови програмування для імплементації інформаційної технології

Наявність великої кількості розробників та бібліотек є індикатором актуальності вибору мови програмування Python в якості мови програмування інтелектуальної технології. Розглянемо бібліотеки, які є поширеними у області машинного навчання.

Першочергово, необхідно підготувати та проаналізувати дані. Для маніпулювання даними популярними бібліотеками є NumPy та pandas.

Numpy — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Оскільки Python — інтерпретована мова, математичні алгоритми, часто працюють в ньому набагато повільніше ніж у компільованих мовах, таких як C або навіть Java. NumPy намагається вирішити цю проблему для великої кількості обчислювальних алгоритмів забезпечуючи підтримку багатовимірних масивів і безліч функцій і операторів для роботи з ними. Таким чином будь-який алгоритм, який може бути виражений в основному як послідовність операцій над масивами і матрицями, працює так само швидко, як еквівалентний код, написаний на C. NumPy можна розглядати як гарну вільну альтернативу MATLAB, оскільки мова програмування MATLAB зовні нагадує NumPy: обидві вони інтерпретовані, і обидві дозволяють користувачам писати швидкі програми поки більшість операцій проводяться над масивами або матрицями, а не над скалярами. Перевага MATLAB у

великій кількості доступних додаткових тулбоксів, включаючи такі як пакет Simulink. Основні пакети, що доповнюють NumPy, це: SciPy — бібліотека, що додає більше MATLAB-подібної функціональності; Matplotlib — пакет для створення графіки в стилі MATLAB. Внутрішньо як MATLAB, так і NumPy базується на бібліотеці LAPACK, призначеної для вирішення основних задач лінійної алгебри [34].

Pandas — програмна бібліотека, написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Вона, зокрема, пропонує структури даних та операції для маніпулювання чисельними таблицями та часовими рядами. pandas є вільним програмним забезпеченням, що випускається за трипунктовою ліцензією BSD.

Pandas в основному використовується для аналізу даних і відповідної обробки табличних даних у DataFrames. Pandas дозволяє імпортувати дані з різних форматів файлів, таких як значення, розділені комами, JSON, Parquet, таблиці або запити бази даних SQL і Microsoft Excel. Pandas дозволяє виконувати різноманітні операції маніпулювання даними, такі як об’єднання, зміна форми, вибір, а також функції очищення та суперечки даних. Розробка pandas ввела в Python багато порівнянних функцій роботи з DataFrames, які були встановлені в мові програмування R. Бібліотека pandas побудована на основі іншої бібліотеки NumPy, яка орієнтована на ефективну роботу з масивами замість функцій роботи з DataFrames [35].

Matplotlib — бібліотека на мові програмування Python для візуалізації даних двовимірною 2D графікою (3D графіка також підтримується). Отримувані зображення можуть бути використані як ілюстрації в публікаціях.

Matplotlib написана і підтримується в основному Джоном Хантером і поширюється на умовах BSD-подібної ліцензії. Зображення, які генеруються в різних форматах, можуть бути використані в інтерактивній графіці, наукових публікаціях, графічному інтерфейсі користувача, вебдодатках, де потрібно будувати діаграми (англ. plotting). В документації автор зізнається, що

Matplotlib починався з імітування графічних команд MATLAB, але є незалежним від нього проектом.

Бібліотека Matplotlib побудована на принципах ООП, але має процедурний інтерфейс pylab, який надає аналоги команд MATLAB.

Matplotlib є гнучким, легко конфігурованим пакетом, який разом з NumPy, SciPy і IPython надає можливості, подібні до MATLAB. В даний час пакет працює з декількома графічними бібліотеками, включаючи wxWindows і PyGTK.

Пакет підтримує багато видів графіків і діаграм:

- Графіки (line plot);
- Діаграми розсіювання (scatter plot);
- Стовпчасті діаграми (bar chart) і гістограми (histogram);
- Секторні діаграми (pie chart);
- Діаграми «Стовбур-листя» (stem plot);
- Контурні графіки (contour plot);
- Поля градієнтів (quiver);
- Спектральні діаграми (spectrogram).

Користувач може вказати осі координат, сітку, додати підписи і пояснення, використовувати логарифмічну шкалу або полярні координати [36].

Seaborn – це бібліотека для створення статистичної графіки на Python. Він будується на основі matplotlib і тісно інтегрується зі структурами даних pandas.

Seaborn допомагає досліджувати та розуміти свої дані. Його функції побудови графіків працюють із кадрами даних і масивами, що містять цілі набори даних, і внутрішньо виконують необхідне семантичне відображення та статистичне агрегування для створення інформативних графіків. Його орієнтований на набори даних декларативний API дозволяє зосередитися на тому, що означають різні елементи ваших графіків, а не на деталях їх малювання [37].

4.3 Програмна реалізація інформаційної технології для виявлення HTTP запитів з аномальною поведінкою

Якщо інформаційну технологію, що розробляється, можливо вважати як окремий програмний модуль, то інтерфейс, що взаємодіє з цим програмним модулем, можливо вважати в якості мікросервісу.

Для реалізації будь якої функціональної логіки потрібно розробити власний «View», у якому описується функціональна реалізація потрібної логіки. «View» приводиться у рух коли викликається відповідним запитом. Для цього потрібно закріпити його за певним маршрутом, перейшовши яким «View» буде приведено у дію. Відображення логіки та результатів користувачеві відбувається завдяки «Template», що може виступати як HTML документом, у випадку веб-сторінки, так і певним вікном, якщо це, наприклад, мобільний додаток. Дані, які репрезентують об’єкти, що демонструються та зберігаються до бази даних, відносяться до спроектованих моделей. Надання та отримання даних відбувається у JSON форматі, в той час як їх зберігання та обробка – у відповідних типах даних, якими оперують Python чи Django, наприклад – тип даних часу з урахуванням часового поясу. Для того, щоб оперувати даними та надавати їх будь-якому пристрою та клієнту, необхідно проводити серіалізацію кожен раз, коли дані надано. Загальний алгоритм відображення даних наведено на рисунку 4.1.

Рисунок 4.1 – Загальний алгоритм надання та відображення даних веб-додатком

Інтелектуальний модуль є окремим програмним модулем, який буде отримувати дані у вигляді файлу зі сторони сервісу та обробляти їх.

Загальна структура інформаційної технології зображена на рисунку 4.2.

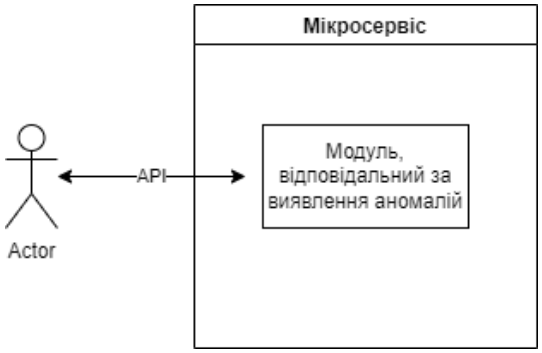


Рисунок 4.2 – Структура інформаційної технології для виявлення HTTP запитів з аномальною поведінкою

Загальна схема алгоритму роботи інформаційної технології зображена на рисунку 4.3.

Рисунок 4.3 – Загальна схема алгоритму роботи інформаційної технології для виявлення HTTP запитів з аномальною поведінкою

Для того, щоб проаналізувати дані, їх необхідно підготувати для аналізу. Для цього використовуються як певні підходи до маніпуляції даними, так і їх візуалізація.

На рисунку 4.4 зображено схему алгоритму роботи модуля, що аналізує дані з HTTP запитами та виявляє аномалії.

Рисунок 4.4 – Схема алгоритму роботи модуля, що аналізує дані з HTTP запитами

Фрагмент коду модулю наведено на рисунку 4.5. Модуль розроблено мовою програмування Python з використанням бібліотек numpy, pandas, seaborn, matplotlib, Scikit-learn.

Рисунок 4.5 – Фрагмент коду інформаційної технології для виявлення HTTP запитів з аномальною поведінкою

Фрагмент коду головного модуля мікросервіса наведено на рисунку 4.6.

```
17  dictConfig(settings.LOGGING_CONFIG)
18  uvicorn_logger = logging.getLogger('uvicorn')
19  uvicorn_handler = logging.StreamHandler(sys.stdout)
20  uvicorn_handler.setFormatter(settings.UvicornFormatter())
21  uvicorn_logger.addHandler(uvicorn_handler)
22
23  sqlalchemy_handler = logging.StreamHandler(sys.stdout)
24  sqlalchemy_logger = logging.getLogger('sqlalchemy.engine')
25  sqlalchemy_handler.setFormatter(settings.SQLAlchemyFormatter())
26  sqlalchemy_logger.addHandler(sqlalchemy_handler)
27
28  app = FastAPI()
29
30
31  @app.get('/favicon.ico', include_in_schema=False)
32  async def favicon():
33      return FileResponse('favicon.ico')
34
35  app.add_middleware(
36      CORSMiddleware,
```

Рисунок 4.6 – Фрагмент коду головного модуля мікросервіса

Модуль розроблено мовою програмування Python з використанням фреймворку FastAPI, бібліотек Celery, SQLAlchemy, Alembic.

4.4 Тестування та аналіз результатів роботи інформаційної технології для виявлення HTTP запитів з аномальною поведінкою

Щоб проаналізувати результати машинного навчання у випадку класифікації, можливо створити класифікаційні звіти.

Класифікаційний звіт надає детальний результат як щодо метрик, так і демонструє точність в залежності від значення цільової змінної.

На рисунку 4.7 наведено фрагмент коду, що відповідає за підготовку даних до машинного навчання.

Рисунок 4.7 – Фрагмент коду підготовки даних

Розглянемо модель k-NN та результати її виконання. На рисунку 4.8 зображено фрагмент коду підготовки k-NN та інших моделей, їх застосування до підготовлених даних.

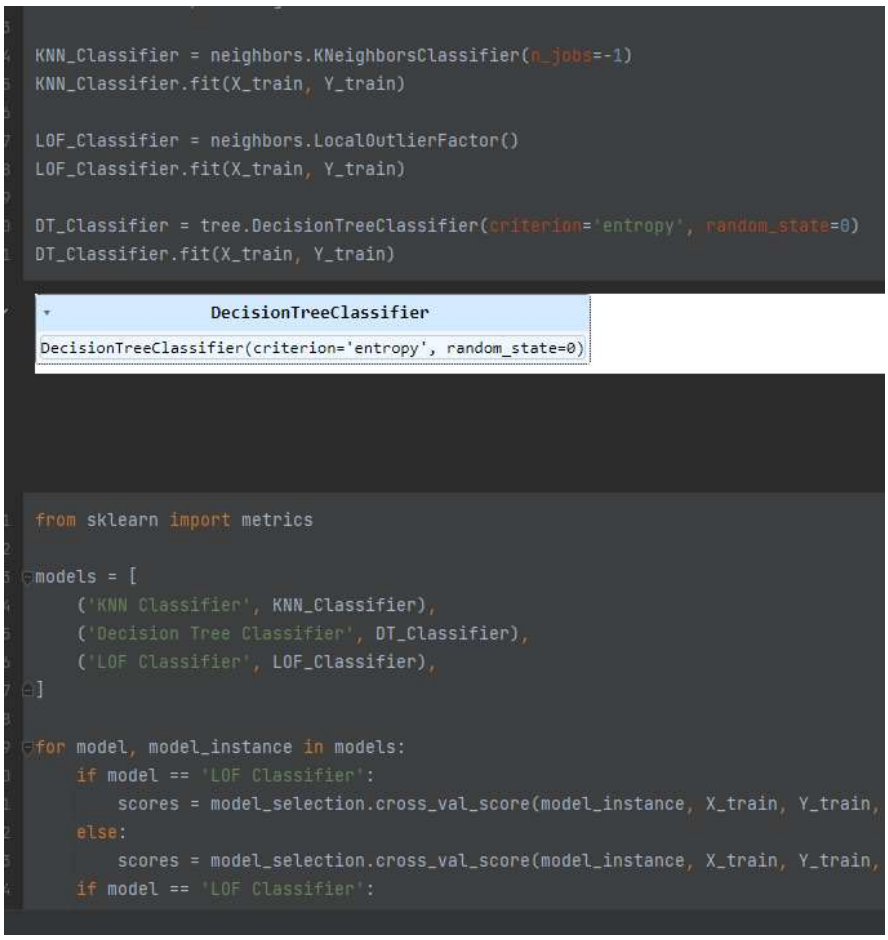


Рисунок 4.8 – Створення об’єктів моделей машинного навчання

На рисунку 4.9 зображені результати k-NN моделі – рахунок перехресної валідації, точність та матриця плутанини.

```
KNN Classifier evaluation

Cross Validation Mean score:
 0.9603352565525338
Accuracy:
 0.9703393178765181
Confusion matrix:
 [[  967    94   204    39    21    29    16    25     0     0]
 [  572   362    42    52   105    35    27    24    21     0]
 [   42     7 682325    55   846  2220    54   413    41     1]
 [  679   279   293   626  1170   149    97   153    16     1]
 [  682   282  1041   348 13150   304   178   357    72     0]
 [  683   267  3318   135   176  7717    56   239     0     0]
 [  691   263   258   215  1145   144   962    79    14     0]
```

Рисунок 4.9 – Результат машинного навчання для k-NN моделі

Щоб зрозуміти, навіщо є перевірка на перехресну валідацію, розглянемо два типи моделей – underfitting та overfitting.

Модель overfitting. Overfitting відбувається, коли статистична модель або алгоритм машинного навчання вловлює шум даних. Інтуїтивно зрозуміло, що overfitting відбувається, коли модель або алгоритм занадто добре відповідають даним.

Overfitting моделі призводить до гарної точності для тренувального набору даних, але поганих результатів для нових наборів даних. Така модель не має користі в реальному світі, оскільки вона не здатна передбачити результати нових випадків.

Underfitting модель. Underfitting виникає, коли статистична модель або алгоритм машинного навчання не можуть охопити основну тенденцію даних. Інтуїтивно зрозуміло, що underfitting виникає, коли модель або алгоритм недостатньо добре підходять для даних. Underfitting часто є результатом надмірно простої моделі. Під простими мається на увазі, що відсутні дані не обробляються належним чином, немає обробки викидів, видалення нерелевантних функцій або функцій, які не роблять великого внеску в змінну предиктора.

Як вирішити проблему overfitting – відповідь – перехресна перевірка.

Ключова проблема з overfitting і з машинним навчанням загалом полягає в тому, що ми не можемо знати, наскільки добре модель працюватиме на нових даних, доки не буде виконано тестування [38].

Щоб уникнути цього, під час виконання експерименту з машинним навчанням (керованим) прийнято зберігати частину доступних даних як тестовий набір X_{test}, y_{test}. Слово «експеримент» не призначене для позначення лише академічного використання, оскільки навіть у комерційних умовах машинне навчання зазвичай починається експериментально. Блок-схема типового процесу перехресної перевірки під час навчання моделі наведена на рисунку 4.10.

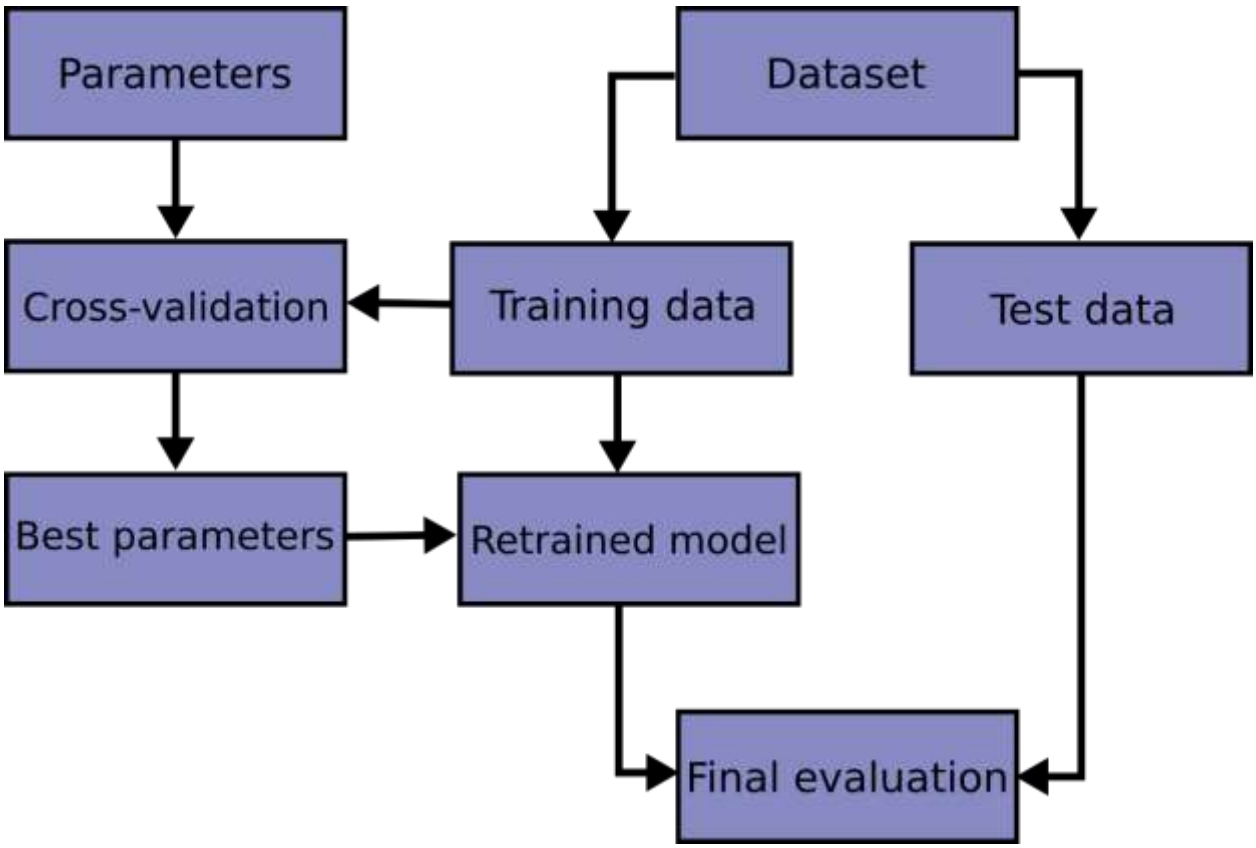


Рисунок 4.10 – Типова схема процесів перехресної перевірки [39]

Розглянемо матрицю плутанини. У сфері машинного навчання та, зокрема, у проблемі статистичної класифікації, матриця плутанини, також відома як матриця помилок, є певним макетом таблиці, який дозволяє візуалізувати продуктивність алгоритму, як правило, навчання під наглядом (за неконтрольованого навчання зазвичай називають відповідною матрицею). Кожен рядок матриці представляє екземпляри в фактичному класі, тоді як кожен стовпець представляє екземпляри в прогнозованому класі, або навпаки – обидва варіанти зустрічаються в літературі. Назва походить від того факту, що вона дозволяє легко побачити, чи система плутає два класи (тобто, як правило, неправильно позначає один як інший).

Це особливий вид таблиці непередбачуваності з двома вимірами («фактичним» і «прогнозованим») і ідентичними наборами «класів» в обох вимірах (кожна комбінація виміру та класу є змінною в таблиці непередбачуваності) [40].

Перейдемо до класифікаційного звіту k-NN, що зображений на рисунку 4.11.

Classification report:

	precision	recall	f1-score	support
Analysis	0.19	0.69	0.30	1395
Backdoor	0.20	0.29	0.24	1240
Benign	0.99	0.99	0.99	686004
DoS	0.39	0.18	0.25	3463
Exploits	0.75	0.80	0.77	16414
Fuzzers	0.70	0.61	0.65	12591
Generic	0.67	0.26	0.37	3771
Reconnaissance	0.81	0.69	0.75	8144
Shellcode	0.75	0.58	0.65	876
Worms	0.55	0.06	0.10	104

Рисунок 4.11 – Класифікаційний звіт

Розглянемо колонки precision, recall, f1-score та support.

Точність рахується за формулою 4.1:

$$precision = \frac{\mathrm{T}P}{\mathrm{TP+FP}} \quad (4.1),$$

та визначається як відношення справжніх позитивних результатів до суми справжніх і помилкових позитивних результатів.

Запам'ятовування, яке також називають чутливістю або швидкістю попадання, повідомляє нам про частку правильно визначених позитивних прогнозів, і рахується за формулою 4.2:

$$recall = \frac{\mathrm{T}P}{\mathrm{TP+FN}} \quad (4.2).$$

Показник f1, або міра F, одночасно вимірює точність і запам’ятовування, знаходячи середнє гармонійне двох значень.

Ця оцінка корисна, коли є протилежні оцінки за точністю та пам’яттю, та обчислюється за формулою 4.3:

$$f1 - score = \frac{2*recall*precision}{\text{recall+precision}} \quad (4.3).$$

Точність вимірює точність усіх передбачень (позитивних і негативних), та обчислюється за формулою 4.4:

$$accuracy = \frac{TP+TN}{\text{TP+TN+FP+FN}} \quad (4.4).$$

Підтримка – це кількість входжень кожного класу до y_test [41].

k-NN дав не найкращі результати. Перевіривши різні моделі, модель дерева рішень дала найточніші результати на поточному наборі даних.

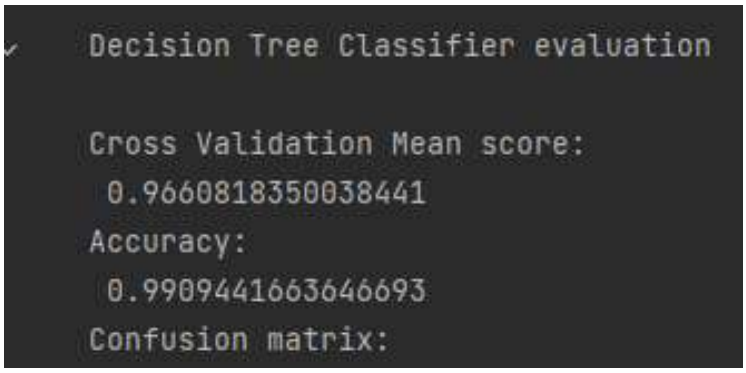


Рисунок 4.12 – Класифікаційний звіт моделі дерева рішень

Ентропія – це кількість інформації, необхідна для точного опису деякої вибірки. Отже, якщо зразок однорідний, це означає, що всі елементи подібні, тоді ентропія дорівнює 0, інакше, якщо зразок поділено порівну, ентропія дорівнює максимум 1.

Отже, ліва чаша має найменшу ентропію, середня чаша має більшу ентропію, а права чаша має найвищу ентропію.

Математична модель представлена формулою 4.5:

$$entropy = \sum_{i=1}^{n} \mathrm{p_i} * \log(p_i) \quad (4.5).$$

4.5 Висновок

У даному розділі проаналізовано та обґрунтовано вибір програмних засобів для імплементації інформаційної технології для виявлення HTTP запитів з аномальною поведінкою. Мову програмування Python було обрано як для імплементації сервісу, так і інформаційної технології у зв’язку з розвинутими і сучасними інструментами, що поширено використовуються для розв’язання поставлених у цій роботі задач.

Імплементовано та виконано тестування результатів інформаційного модулю, що дало можливість визначити дерева рішень як найдоречніший алгоритм машинного навчання.

Інформаційна технологія виконала поставлені задачі, виявивши усі типи аномалій, продемонструвавши високу точність та усунула недоліки аналогів, починаючи з необхідності технічного втручання.

5 ЕКОНОМІЧНА ЧАСТИНА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ ВИЯВЛЕННЯ HTTP ЗАПИТІВ З АНОМАЛЬНОЮ ПОВЕДІНКОЮ

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

2) розраховано витрати на здійснення науково-технічної розробки;

3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Протягом останнього часу зростає кількість та складність кібератак на промислові підприємства, органи влади та державні установи. Атаки виконуються шляхом впливу на веб-сайти, інші мережеві служби, телекомунікаційне обладнання, спрямовані на загрозу конфіденційності, цілісності, доступності даних, а також на послаблення спостережності і керованості системами, де передається, оброблюється, зберігається інформація. Тому застосування додаткових технологій виявлення HTTP запитів з аномальною поведінкою є необхідним, актуальним та комерційно перспективним напрямом.

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 [42].

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	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в
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

Продовження таблиці 5.1.

3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно	Технічні та споживчі влас-тивості продукту трохи гірші ніж в	Технічні та споживчі властивості продукту на рівні	Технічні та споживчі властивості продукту трохи	Технічні та споживчі властивості продукту значно
5	Експлуатаційні витрати значно вищі, ніж в	Експлуатаційні витрати дещо вищі, ніж в	Експлуатаційні витрати на рівні експлуатаційних	Експлуатаційні витрати трохи нижчі, ніж в	Експлуатаційні витрати значно нижчі, ніж в
Ринкові перспективи					
6	Ринок малий і не має позитивної	Ринок малий, але має позитивну	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною
7	Активна конкуренція великих компаній	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	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	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 використовують ся у військово	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуютьс

Продовження таблиці 5.1.

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що	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 наступної таблиці.

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

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	4	5	4
3. Ринкові переваги (ціна продукту)	4	4	4
4. Ринкові переваги (технічні властивості)	4	4	5
5. Ринкові переваги (експлуатаційні витрати)	1	1	1
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	3	2	3
8. Практична здійсненність (наявність фахівців)	4	4	3
9. Практична здійсненність (наявність фінансів)	3	4	3

Продовження таблиці 5.2.

10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	5	4	4
12. Практична здійсненність (розробка документів)	5	4	5
Сума балів	45	44	44
Середньоарифметична сума балів $СБ_с$	44,3		

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 5.3 [42].

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 «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою» становить 44,3 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий). Дослідна розробка матиме такі відмінності від конкурентів та переваги:

1) Наявність спеціалізації на виявлення саме HTTP аномалій;

2) Легка інтеграція - дані надсилаються до інформаційного модуля за допомогою API;

3) Низька вартість послуг, що зумовлена наявністю спеціалізації;

4) Підвищена точність виявлення аномалій порівняно з конкурентами;

5) Швидкодія сервісу, оскільки розроблено найсучаснішими технологіями

Використані програмні засоби:

IDE - PyCharm, мова програмування Python, фреймворк FastAPI, бібліотеки для обробки, аналізу даних та машинного навчання: Numpy, Pandas, TensorFlow, Matplotlib.

5.2 Оцінювання рівня новизни розробки

Виводячи на ринок новинку виробник вважає, що тієї новизни, якою наділена нова розробка є достатньо для того, щоб вона була сприйнята споживачем як нова. Але це не завжди так, в силу того, що споживач і виробник неоднозначно визначать її рівень новизни. Тому доцільним є визначення рівня новизни розробки отриманої в результаті досліджень за темою «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою».

Саме визначення рівня і ступеня інтегральної новизни є найбільш актуальним, оскільки її рівень визначає ступінь однакового позитивного сприйняття новизни розробки як виробником, так і споживачем, а отже і ринком в цілому, а це, у свою чергу, є гарантією того, що новинка знайде своє місце на ринку, користуватиметься попитом у споживачів і забезпечить відшкодування витрат, зазнаних товаровиробником під час розроблення та виробництва технічної розробки [43].

Рівень новизни нової продукції розраховуємо експертним методом шляхом протиставлення нової продукції та її аналогів, що існують в даний час на ринку, за чинниками що визначають її значення, в системі «краще-гірше». Рівень новизни встановлюємо відносно рівня аналога (або продукту, що досить близький до аналога).

Для визначення *i*-го виду новизни, застосуємо чинники, які впливають на її рівень. Кожен чинник *i*-го виду новизни розраховуємо в балах. Більша

кількість набраних балів свідчить про більший рівень новизни. Для оцінювання рівня новизни використаємо думки експертів, які встановлюють визначені бали відповідним чинникам. Бал відповідності проставляється в діапазоні від (-5 –значно гірше аналога до +5 – значно краще аналога). Результати попереднього оцінювання зведемо до відповідного листа оцінювання (таблиця 5.4).

Таблиця 5.4 – Лист оцінювання рівня новизни експертами

Види та чинники		Бали та експерти		
		Експерт 1	Експерт 2	Експерт 3
1		*2*	*3*	*4*
Споживча новизна	Питома вага 0,24	Максимальний бал $Б_{i\ MAX}$		25
1. Зміна поведінкових звичок споживача		4	4	4
2. Ступінь задоволення потреб і запитів		5	5	5
3. Спосіб задоволення потреби		3	3	4
4. Формування нової потреби		3	3	2
5. Формування нового споживача		0	0	0
Середній бал експертів $Б_{i\ отр}$		15		
Товарна новизна	Питома вага 0,202	Максимальний бал $Б_{i\ MAX}$		30
1. Параметричні зміни показників продукції				
1.1. Якісні		3	4	3
1.2. Технічні		4	4	3
1.3. Економічні		3	3	3
1.4. Сервісні		4	4	4
2. Якість продукції по відношенню до конкурентів		3	3	3
3. Функціональні зміни		3	3	3
Середній бал експертів $Б_{i\ отр}$		20		
Виробнича новизна	Питома вага 0,042	Максимальний бал $Б_{i\ MAX}$		25
1. Рівень унікальності товару для підприємства		5	5	5

Продовження таблиці 5.4.

2. Рівень унікальності для галузі		3	4	3
3. Рівень унікальності товару для країни		1	1	1
4. Зміна виробничої системи		4	4	4
5. Відносно існуючого асортименту		2	2	2
Середній бал експертів $Б_{i\ отр}$		15		
Прогресивна новизна	Питома вага 0,2	Максимальний бал $Б_{i\ MAX}$		25
1. Зміна технології виготовлення		4	4	4
2. Рівень застосування нових компонентів і матеріалів		1	2	1
3. Зміна технологічного принципу дії виробу		1	2	1
4. Зміна конструктивного виконання		3	2	3
5. Рівень застосування інновацій		2	2	2
Середній бал експертів $Б_{i\ отр}$		11		
Ринкова новизна	Питома вага 0,1	Максимальний бал $Б_{i\ MAX}$		20
1. Новий виріб на новому ринку		0	0	0
2. Новий виріб на відомому ринку		2	2	2
3. Модернізований виріб		2	2	2
4. Нова модель		1	2	2
Середній бал експертів $Б_{i\ отр}$		6		
Екологічна новизна	Питома вага 0,035	Максимальний бал $Б_{i\ MAX}$		20
1. Рівень екологічної чистоти технології виробництва		5	5	5
2. Рівень впровадження мало- та безвідходних технологій		5	5	5
3. Рівень екологічно небезпечних режимів експлуатації продукції		5	5	5
4. Рівень забруднення навколишнього середовища		5	5	5
Середній бал експертів $Б_{i\ отр}$		20		
Соціальна новизна	Питома вага 0,036	Максимальний бал $Б_{i\ MAX}$		20

Продовження таблиці 5.4.

1. Використання нового товару приводить до покращення стану здоров'я нації		0	0	0
2. Використання нового товару приводить до зростання доходів населення		0	0	0
3. Виробництво нового товару приводить до збільшення (зменшення) кількості робочих місць на підприємстві		4	5	4
4. Виробництво нового товару приводить до підвищення кваліфікації персоналу		3	3	3
Середній бал експертів $Б_{i\ отр}$		7		
Маркетингова новизна	Питома вага 0,145	Максимальний бал $Б_{i\ MAX}$		20
1. Нові методи маркетингових досліджень		0	0	0
2. Вживання нових стратегій сегментації ринку		3	3	3
3. Вибір нової маркетингової стратегії обхвату і розвитку цільового сегмента		2	3	2
4. Побудова нових каналів збуту		0	1	1
Середній бал експертів $Б_{i\ отр}$		6		

Значення і-го виду новизни розрахуємо за формулою [43]:

$$I_i = \frac{Б_{i\,отр}}{Б_{i\,MAX}}, \qquad (5.1)$$

де $Б_{i\ отр}$ – отримана кількість балів за шкалою оцінок чинників, що визначають *і*-й вид новизни;

$Б_{i\ MAX}$ – максимальна кількість балів, що може бути отримана за *і*-м видом новизни.

Загальний рівень інтегральної новизни розраховуємо шляхом перемноження отриманого значення і-го виду новизни на її вагомість, причому вагомість і-го виду новизни визначаємо експертним методом, за формулою [42]:

$$N_{інт} = \sum_{i}^{n} W_i \cdot I_i \ , \qquad (5.2)$$

де $N_{інт}$ – рівень інтегральної (сукупної) новизни;

W_i – вагомість (питома вага) i-го виду новизни;

n – загальна кількість видів новизни.

$$N_{інт} = (0{,}24 \cdot 15/25)+(0{,}202 \cdot 20/30)+(0{,}042 \cdot 15/25)+(0{,}2 \cdot 11/25)+(0{,}1 \cdot 6/20)+$$
$$(0{,}035 \cdot 20/20)+(0{,}036 \cdot 7/20)+(0{,}145 \cdot 6/20) = 0{,}515.$$

Отримане значення інтегрального рівня новизни зіставляємо зі шкалою, що наведена в табл. 5.5 [42].

Таблиця 5.5 – Рівні новизни нового товару та їхня характеристика

Рівні новизни товару	Значення інтегральної новизни	Характеристика товару	Вид нового товару
Найвища	1,00	Абсолютно новий товар	Новий товар, що наділений ознаками інноваційності (інноваційний товар)
Висока	0,8…0,99	Товар, який не має аналогів	
Значуща	0,6…0,79	Принципова зміна споживчих властивостей товару	
Достатня	0,4…0,59	Принципова технологічна модифікація товару	
Незначна	0,2…0,39	Кардинальна зміна параметрів	Новий товар
Помилкова	0,00…0,19	Малоістотна модифікація	

Згідно таблиці 5.5 розробка відповідає рівню при значенні інтегральної новизни 0,515 - достатня новизна; за характеристикою: принципова технологічна модифікація товару; вид розробки - новий товар, що наділений ознаками інноваційності (інноваційний товар).

5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою», 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 ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [42]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \qquad (5.3)$$

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.

$$З_о = 13500{,}00 \cdot 21 / 21 = 13500{,}00 \; грн.$$

Проведені розрахунки зведемо до таблиці, дані якої наведено у таблиці 5.6.

Таблиця 5.6 – 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	13500,00	642,86	21	13500,00
Інженер-розробник програмного забезпечення	12200,00	580,95	21	12200,00
Науковий співробітник дослідження проблем аномальної поведінки в комунікаційних мережах	12500,00	595,24	10	5952,38
Технік	7100,00	338,10	10	3380,95
Всього				35033,33

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i , \qquad (5.4)$$

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 C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} , \qquad (5.5)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $Мм$=6700,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [42];

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.

$C_1 = 6700{,}00 \cdot 1{,}10 \cdot 1{,}7 / (21 \cdot 8) = 74{,}58$ *грн.*

$З_{р1} = 74{,}58 \cdot 6{,}00 = 447{,}46$ *грн.*

На таблиці 5.7 наведено дані величини витрат на основну заробітну плату робітників.

Таблиця 5.7 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка електронно-обчислювального обладнання	6,00	2	1,10	74,58	447,46
Підготовка робочого місця дослідника	2,40	2	1,10	74,58	178,99
Інсталяція програмного забезпечення	2,20	5	1,70	115,26	253,56
Формування дослідної бази даних інформації	12,00	2	1,10	74,58	894,93

Продовження таблиці 5.7.

Налагодження програмних блоків	5,60	5	1,70	115,26	645,43
Монтаж серверного обладнання	10,00	5	1,70	115,26	1152,56
Монтаж блоків пам'яті	3,00	4	1,50	101,70	305,09
Тестування системи	8,00	2	1,10	74,58	596,62
Всього					4474,64

Додаткова заробітна плата дослідників та робітників

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

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (5.6)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 10%.

$З_{дод}$ = (35033,33 + 4474,64) · 10 / 100% = 3950,80 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:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (5.7)$$

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

Зн = (35033,33 + 4474,64 + 3950,80) · 22 / 100% = 9560,93 грн.

5.3.3 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 «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою».

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

$$M = \sum_{j=1}^{n} H_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} B_j \cdot Ц_{вj}, \qquad (5.8)$$

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.

$M_1 = 3,0 \cdot 290,00 \cdot 1,11 - 0,000 \cdot 0,00 = 965,70$ *грн.*

Проведені розрахунки зведемо до таблиці, що виражена таблицею 5.8.

Таблиця 5.8 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір Calipso Plus A4-500-80	290,00	3,0	-	-	965,70
Папір для записів Calipso Parers Light A5	145,00	1,0	-	-	160,95

Продовження таблиці 5.8.

Органайзер офісний Calipso Office	210,00	3,0	-	-	699,30
Канцелярське приладдя (набір офісного працівника)	175,00	5,0	-	-	971,25
Картридж для принтера Canon LBP6500	920,00	1,0	-	-	1021,20
Диск оптичний NewLine CD-RW	23,00	3,0	-	-	76,59
Flesh-пам'ять Kingston 16 GB	320,00	1,0	-	-	355,20
Тека для паперів CALIPSO BOX	120,00	3,0	-	-	397,60
Всього					4647,79

5.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (Кв), які використовують при проведенні НДР на тему «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою» відсутні.

5.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування

необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (5.9)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, (K_i = *1,10…1,12);*

k – кількість найменувань устаткування.

$B_{спец}$ = *45200,00 · 1 · 1,11 = 50172,00 грн.*

Отримані результати зведемо до таблиці, що виражена таблицею 5.9.

Таблиця 5.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Серверне обладнання на основі ПК ZEVS PC 13430U i5 9400F + GTX 1060 3GB	1	45200,00	50172,00
Засоби передачі даних	1	11650,00	12931,50
Всього			63103,50

5.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних)

необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (5.10)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань програмних засобів.

$B_{прг} = 8415,00 \cdot 1 \cdot 1,11 = 9340,65$ *грн.*

Отримані результати зведемо до таблиці, що виражена таблицею 5.10:

Таблиця 5.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows	1	8415,00	9340,65
Прикладний пакет Microsoft Office	1	7810,00	8669,10
Програмний засіб IDE – PyCharm	1	7560,00	8391,60
фреймворк FastAPI (благодійність)	1	4860,00	5394,60
бібліотеки для обробки, аналізу даних та машинного навчання: Numpy, Pandas, TensorFlow, Matplotlib (благодійність)	1	15230,00	16905,30
Всього			48701,25

5.3.7 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:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (5.11)$$

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.

$A_{обл} = (24370{,}00 \cdot 1) / (2 \cdot 12) = 1015{,}42$ *грн.*

Проведені розрахунки зведемо до таблиці, дані якої наведено у таблиці 5.11.

Таблиця 5.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер	24370,00	2	1	1015,42
Робоче місце дослідника	7880,00	5	1	131,33
Пристрої виводу інформації	6875,00	4	1	143,23
Оргтехніка	8675,00	4	1	180,73
Приміщення лабораторії	215000,00	25	1	716,67
Всього				2187,38

5.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (5.12)$$

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

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e = 6,20$ грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

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

$B_e = 0,45 \cdot 160,0 \cdot 6,20 \cdot 0,95 / 0,97 = 446,40$ *грн.*

Проведені розрахунки зведемо до таблиці 5.12.

Таблиця 5.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер	0,45	160,0	446,40
Робоче місце дослідника	0,15	160,0	148,80
Пристрої виводу інформації	0,03	60,0	11,16
Оргтехніка	0,65	12,0	48,36
Серверне обладнання на основі ПК ZEVS PC 13430U i5 9400F + GTX 1060 3GB	0,60	140,0	520,80
Засоби передачі даних	0,05	120,0	37,20
Всього			1212,72

5.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою» 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:

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \qquad (5.13)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 22%.

$B_{св}$ = (35033,33 + 4474,64) · 22 / 100% = 8691,75 грн.

5.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

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

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 35%.

$B_{сп}$ = (35033,33 + 4474,64) · 35 / 100% = 13827,79 грн.

5.3.11 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 $Н_{ів}$ = 55%.

$I_в$ = (35033,33 + 4474,64) · 55 / 100% = 21729,39 грн.

5.3.12 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 $Н_{нзв}$ = 115%.

$В_{нзв}$ = (35033,33 + 4474,64) · 115 / 100% = 45434,17 грн.

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою» розраховуємо як суму всіх попередніх статей витрат за формулою:

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

$B_{заг}$= 35033,33 +4474,64 +3950,80 +9560,930238 +4647,79 +0,00 +63103,50 + 48701,25 + 2187,38 +1212,72 +8691,75 +13827,79 +21729,39 +45434,17 = 262555,45 грн.

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

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

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,9.

ЗВ = 262555,45 / 0,9 = 291728,27 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 «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	650	1300	1600	800

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 3000 осіб;

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки, приймемо 9400,00 грн;

$\pm\Delta Ц_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, приймемо 558,70 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [42]:

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

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2022 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо ρ =30%;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2022 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (558,70 \cdot 3000,00 + 9958,70 \cdot 650) \cdot 0,83 \cdot 0,3 \cdot (1\text{-}0,18/100\%) = 1663914,89$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (558{,}70 \cdot 3000{,}00 + 9958{,}70 \cdot 1950) \cdot 0{,}83 \cdot 0{,}3 \cdot (1 - 0{,}18/100\%) = 4307292{,}46$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (558{,}70 \cdot 3000{,}00 + 9958{,}70 \cdot 3550) \cdot 0{,}83 \cdot 0{,}3 \cdot (1 - 0{,}18/100\%) = 7560680{,}25$ грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4 = (558{,}70 \cdot 3000{,}00 + 9958{,}70 \cdot 4350) \cdot 0{,}83 \cdot 0{,}3 \cdot (1 - 0{,}18/100\%) = 9187374{,}14$ грн.

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

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

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, $\tau = 0{,}22$;

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

$ПП = 1663914{,}89/(1+0{,}22)^1 + 4307292{,}46/(1+0{,}22)^2 + 7560680{,}25/(1+0{,}22)^3 +$

$+ 9187374{,}14/(1+0{,}22)^4 = 1363864{,}66 + 2893907{,}86 + 4163718{,}69 + 4147172{,}31 =$

$= 12568663{,}52$ грн.

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

$$PV = k_{інв} \cdot ЗВ, \qquad (5.21)$$

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 291728,27 грн.

$PV = k_{інв} \cdot ЗВ = 2 \cdot 291728{,}27 = 583456{,}54$ 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, 12568663,52 грн;

PV – теперішня вартість початкових інвестицій, 583456,54 грн.

$E_{абс} = ПП - PV = 12568663{,}52 - 583456{,}54 = 11985206{,}97$ грн.

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

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{5.23}$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 11985206,97 грн;

PV – теперішня вартість початкових інвестицій, 583456,54 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_в = \sqrt[Тж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+11985206,97/583456,54)^{1/4} = 1,15.$$

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

$$\tau_{мін} = d + f, \qquad (5.24)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d =0,2;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,6.

$\tau_{мін}$ = 0,2+0,6 = 0,8 < 1,15 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 виявлення HTTP запитів з аномальною поведінкою» доцільно.

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

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

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 1,15 = 0,87 р.

$T_{ок}$ < 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 Висновок

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою» становить 44,3 бали, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

Розробка відповідає при значенні інтегральної новизни 0,515 - рівню достатня новизна; за характеристикою: принципова технологічна модифікація товару; вид розробки - новий товар, що наділений ознаками інноваційності (інноваційний товар).

Також термін окупності становить 0,87 р., 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 «Інформаційна технологія виявлення HTTP запитів з аномальною поведінкою».

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було створено інформаційну технологію на базі мікросервісу, що дозволяє аналізувати HTTP запити на вміст аномалій.

Проведено аналіз предметної області, розглянуто та проаналізовано аналоги, виявлено їх недоліки та переваги. Відштовхуючись від недоліків, поставлено задачу для створюваної інформаційної технології.

Базуючись на висновках теореми про відсутність безкоштовних сніданків, задля дотримання високоточних результатів, досліджено та виконано практичну перевірку сучасних моделей машинного навчання, задача яких пов’язана з класифікацією об’єктів. На основі результатів моделей обрано дерево рішень в якості головної моделі машинного навчання. Відштовхуючись від найвпливовіших ознак, виявлених на етапі обирання ознак, визначено модель запиту, яка була застосована на етапах машинного навчання.

Інформаційну технологію імплементовано на основі мікросервісу, усунувши недоліки як точності навчання, так і необхідності втручання технічного спеціаліста.

Виконано тестування розробленого програмного забезпечення, що підтвердило факт виконання поставлених задач у повному обсязі.

Розробка відповідає при значенні інтегральної новизни 0,515 - рівню достатня новизна; за характеристикою: принципова технологічна модифікація товару; вид розробки - новий товар, що наділений ознаками інноваційності (інноваційний товар).

Також термін окупності становить 0,87 р., 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.

Мети роботи досягнуто завдяки підходу, за яким класи та методи машинного навчання обираються в залежності від специфіки предметної області, використанню сучасних технологій і визначеній моделі в ході аналізу та підготовки даних з набору.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. DoS-атака [Електронний ресурс] Режим доступу – www.wikiwand.com/uk/DoS-атака
2. Зелений В. Є., Арсенюк І. Р. Інформаційна технологія для виявлення HTTP запитів з аномальною поведінкою. Молодь в науці: дослідження, проблеми, перспективи (МН-2023), 15.11.22 – 11.05.23 : збірник матеріалів. – Вінниця: ВНТУ, 2023. – [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/viewFile/16913/14097Комп'ютерна безпека [Електронний ресурс] Режим доступу – www.wikiwand.com/uk/Комп%27ютерна_безпека
3. HTTPS [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/HTTPS
4. Oracle – Anomaly Detection [Електронний ресурс] Режим доступу – docs.oracle.com/en-us/iaas/Content/anomaly/using/overview.htm
5. Anomaly [Електронний ресурс] Режим доступу – anomaly.io/detect-anomaly/index.html
6. Oracle – AD Overview [Електронний ресурс] Режим доступу – docs.oracle.com/en-us/iaas/Content/anomaly/using/overview.htm
7. Anomaly Detection [Електронний ресурс] Режим доступу – medium.com/analytics-vidhya/algorithm-selection-for-anomaly-detection-ef193fd0d6d1
8. Anomaly Detection [Електронний ресурс] Режим доступу – intellspot.com/anomaly-detection-algorithms/
9. Types of anomalies [Електронний ресурс] Режим доступу – towardsdatascience.com/a-note-about-finding-anomalies-f9cedee38f0b
10. Anomaly detection, Patrick Schneider, Fatos Xhafa, in Anomaly Detection and Complex Event Processing over IoT Data Streams, 2022 [Електронний

ресурс] Режим доступу – sciencedirect.com/topics/engineering/anomaly-detection

11. Навчання з учителем [Електронний ресурс] Режим доступу – wikiwand.com/uk/Навчання_з_учителем
12. Задача класифікації [Електронний ресурс] Режим доступу – wikiwand.com/uk/Задача_класифікації
13. Метод k-найближчих сусідів [Електронний ресурс] Режим доступу – wikiwand.com/uk/Метод_k-найближчих_сусідів
14. Local Outlier Factor [Електронний ресурс] Режим доступу – wikiwand.com/en/Local_outlier_factor
15. Local Outlier Probabilities [Електронний ресурс] Режим доступу – towardsdatascience.com/identifying-outliers-with-local-outlier-probabilities-2b5781e86e01
16. Kriegel, H.-P.; Kröger, P.; Schubert, E.; Zimek, A. (2009). LoOP: Local Outlier Probabilities. Proceedings of the 18th ACM Conference on Information and Knowledge Management. CIKM '09. pp. 1649–1652. doi:10.1145/1645953.1646195. ISBN 978-1-60558-512-3 [Електронний ресурс] Режим доступу – https://www.dbs.ifi.lmu.de/Publikationen/Papers/LoOP1649.pdf
17. No Free Lunch in search and optimization [Електронний ресурс] Режим доступу – wikiwand.com/en/No_free_lunch_in_search_and_optimization
18. ML-Based NIDS Datasets [Електронний ресурс] Режим доступу – https://staff.itee.uq.edu.au/marius/NIDS_datasets/#RA5
19. Кореляція [Електронний ресурс] Режим доступу – wikiwand.com/uk/Кореляція
20. ML Feature Scaling [Електронний ресурс] Режим доступу – https://www.geeksforgeeks.org/ml-feature-scaling-part-2/
21. Обирання ознак [Електронний ресурс] Режим доступу – wikiwand.com/uk/Обирання_ознак

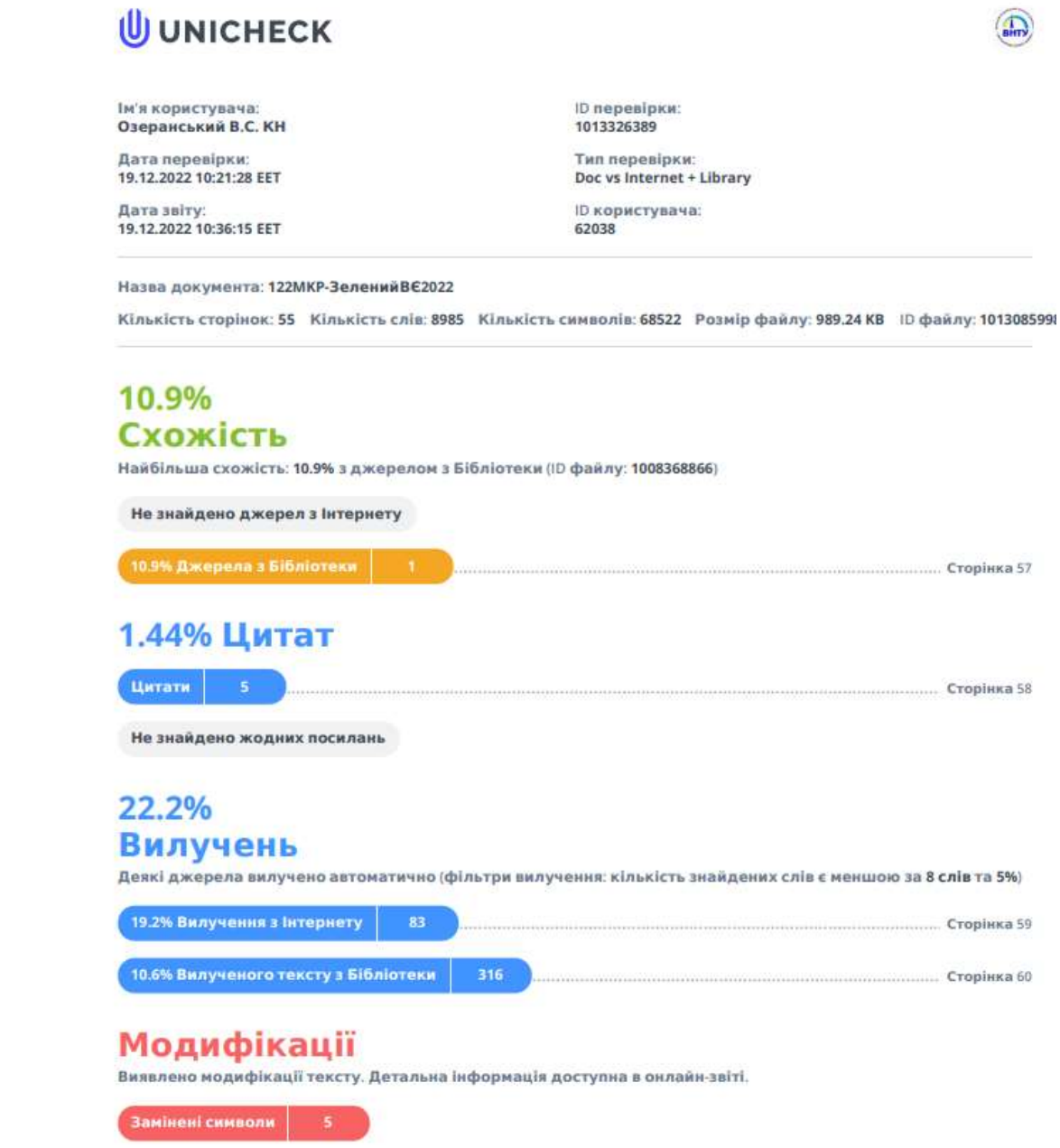
22. JavaScript [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/JavaScript
23. Node.js [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/Node.js
24. Express.js [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/Express.js
25. MEAN [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/MEAN_(веброзробка)
26. Python [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/Python
27. Django [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/Django
28. Flask [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/Flask
29. FastAPI [Електронний ресурс] Режим доступу – fastapi.tiangolo.com
30. PyCharm [Електронний ресурс] Режим доступу – https://www.jetbrains.com/pycharm/features/
31. Celery [Електронний ресурс] Режим доступу – https://www.wikiwand.com/en/Celery_(software)
32. Best language for Machine Learning [Електронний ресурс] Режим доступу – https://www.springboard.com/blog/data-science/best-language-for-machine-learning/
33. NumPy [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/NumPy
34. Pandas [Електронний ресурс] Режим доступу – https://www.wikiwand.com/en/Pandas_(software)
35. Matplotlib [Електронний ресурс] Режим доступу – https://www.wikiwand.com/uk/Matplotlib
36. Seaborn [Електронний ресурс] Режим доступу – https://seaborn.pydata.org/tutorial/introduction.html

37. Cross validation [Електронний ресурс] Режим доступу – https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85
38. Cross validation [Електронний ресурс] Режим доступу – https://scikit-learn.org/stable/modules/cross_validation.html
39. Confusion matrix [Електронний ресурс] Режим доступу – wikiwand.com/en/Confusion_matrix
40. Classification report [Електронний ресурс] Режим доступу – https://www.jcchouinard.com/classification-report-in-scikit-learn/
41. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
42. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

ДОДАТКИ

Додаток А (обов'язковий)

Результат перевірки на плагіат в онлайн-системі UNICHECK



UNICHECK

Ім'я користувача:
Озеранський В.С. КН

ID перевірки:
1013326389

Дата перевірки:
19.12.2022 10:21:28 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
19.12.2022 10:36:15 EET

ID користувача:
62038

Назва документа: 122МКР-ЗеленийВЄ2022

Кількість сторінок: 55 Кількість слів: 8985 Кількість символів: 68522 Розмір файлу: 989.24 KB ID файлу: 101308599

10.9%
Схожість

Найбільша схожість: 10.9% з джерелом з Бібліотеки (ID файлу: 1008368866)

Не знайдено джерел з Інтернету

10.9% Джерела з Бібліотеки 1 ..... Сторінка 57

1.44% Цитат

Цитати 5 ..... Сторінка 58

Не знайдено жодних посилань

22.2%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 5%)

19.2% Вилучення з Інтернету 83 ..... Сторінка 59

10.6% Вилученого тексту з Бібліотеки 316 ..... Сторінка 60

Модифікації

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

Замінені символи 5

Додаток Б (обов'язковий)

Лістинг програми

```
import celery

celery_app = celery.Celery()



@celery_app.task
def analyze(filename):
    import sys

    import pandas as pd
    import numpy as np
    import seaborn as sns
    from matplotlib import pyplot as plt

    df = pd.read_csv(filename)
    df.head()

    df['Label'].unique()

    attacks_data = df[df.Attack != 'Benign']

    sns.set(rc={'figure.figsize': (13, 4)})
    sns.countplot(x='Attack', data=attacks_data)

    plt.figure(figsize=(16, 6))
    corr = df.corr(numeric_only=True)
```

```
        mask = np.triu(np.ones_like(corr, dtype=bool))
        heatmap = sns.heatmap(corr, vmin=-1, vmax=1, annot=True,
cmap='BrBG', mask=mask)
        heatmap.set_title('Triangle correlation map', fontdict={'fontsize': 12},
pad=12)

        simplified_data = df.copy()
        # simplified_data.head()
        simplified_data.loc[df.Attack == 'Benign', 'Attack'] = float(0.0)
        simplified_data.loc[df.Attack != 'Benign', 'Attack'] = float(1.0)
        # simplified_data.head()

        simplified_data.infer_objects()
        convert = {'Attack': 'int64', 'FLOW_DURATION_MILLISECONDS':
'int64'}
        simplified_data = simplified_data.astype(convert)

        simplified_data_heatmap = sns.heatmap(

simplified_data.corr(numeric_only=True)[['Attack']].sort_values(by='Attack',
ascending=False), vmin=-1, vmax=1,
            annot=True, cmap='BrBG')

        plt.plot(df.IN_PKTS, df.OUT_PKTS, "", data=df, linestyle='', marker='o',
markersize=3, alpha=0.5, color="purple")

        # Ignore warnings
        import warnings
        warnings.filterwarnings('ignore')
```

```
pd.set_option('display.max_columns', None)
np.set_printoptions(threshold=sys.maxsize)
np.set_printoptions(precision=3)
sns.set(style="darkgrid")
plt.rcParams['axes.labelsize'] = 14
plt.rcParams['xtick.labelsize'] = 12
plt.rcParams['ytick.labelsize'] = 12

train = pd.read_csv("general_data.csv")

print(train.head(4))



test = pd.read_csv('NF-BoT-IoT.csv')

del train['Dataset']
del train['Label']
del test['Label']



test.head(4)

test_attack = test.pop('Attack')

train.describe()

train['Attack'].value_counts()

from sklearn import preprocessing
scaler = preprocessing.StandardScaler()
```

```
cols = train.select_dtypes(include=['int64', 'float64']).columns

sc_train = scaler.fit_transform(train.select_dtypes(include=['int64',
'float64']))
sc_test = scaler.fit_transform(test.select_dtypes(include=['int64', 'float64']))

sc_train_df = pd.DataFrame(sc_train, columns=cols)
sc_test_df = pd.DataFrame(sc_test, columns=cols)

encoder = preprocessing.LabelEncoder()

categorical_train = train.select_dtypes(include=['object']).copy()
categorical_test = test.select_dtypes(include=['object']).copy()

train_cat = categorical_train.apply(encoder.fit_transform)
test_cat = categorical_test.apply(encoder.fit_transform)

enc_train = train_cat.drop(['Attack'], axis=1)
cat_y_train = train_cat[['Attack']].copy()

train_x = pd.concat([sc_train_df, enc_train], axis=1)
train_y = train['Attack']

test_df = pd.concat([sc_test_df, test_cat], axis=1)

from sklearn import ensemble
rfc = ensemble.RandomForestClassifier()

rfc.fit(train_x, train_y)
```

```
        score = np.round(rfc.feature_importances_, 3)
        importance = pd.DataFrame({'feature': train_x.columns, 'importance':
score})
        importance = importance.sort_values('importance',
ascending=False).set_index('feature')

        plt.rcParams['figure.figsize'] = (11, 4)
        importance.plot.bar()

        from sklearn import feature_selection
        import itertools

        rfc = ensemble.RandomForestClassifier()
        rfe = feature_selection.RFE(rfc, n_features_to_select=10)
        rfe = rfe.fit(train_x, train_y)

        feature_map = [(i, v) for i, v in itertools.zip_longest(rfe.get_support(),
train_x.columns)]
        selected_features = [v for i, v in feature_map if i == True]

        from sklearn import model_selection

        X_train, X_test, Y_train, Y_test = model_selection.train_test_split(train_x,
train_y, train_size=0.70,
                                                            random_state=2)

        from sklearn import tree
        from sklearn import neighbors
```

```
        KNN_Classifier = neighbors.KNeighborsClassifier(n_jobs=-1)
        KNN_Classifier.fit(X_train, Y_train)

        LOF_Classifier = neighbors.LocalOutlierFactor()
        LOF_Classifier.fit(X_train, Y_train)

        DT_Classifier = tree.DecisionTreeClassifier(criterion='entropy',
random_state=0)
        DT_Classifier.fit(X_train, Y_train)



        from sklearn import metrics

        models = [
           ('KNN Classifier', KNN_Classifier),
           ('Decision Tree Classifier', DT_Classifier),
           ('LOF Classifier', LOF_Classifier),
        ]

        for model, model_instance in models:
           if model == 'LOF Classifier':
              scores = model_selection.cross_val_score(model_instance, X_train,
Y_train, cv=10, scoring='top_k_accuracy')
           else:
              scores = model_selection.cross_val_score(model_instance, X_train,
Y_train, cv=10)
           if model == 'LOF Classifier':
              accuracy = metrics.accuracy_score(Y_train,
model_instance.predict(X_train, novelty=True))
           else:
```

```
            accuracy = metrics.accuracy_score(Y_train,
model_instance.predict(X_train))
            confusion_matrix = metrics.confusion_matrix(Y_train,
model_instance.predict(X_train))
            classification = metrics.classification_report(Y_train,
model_instance.predict(X_train))

            print(f'{model} evaluation\n')
            print("Cross Validation Mean score:" "\n", scores.mean())
            print("Accuracy:" "\n", accuracy)
            print("Confusion matrix:" "\n", confusion_matrix)
            plt.figure(figsize=(20, 5))
            sns.heatmap(confusion_matrix, annot=True, cmap='Blues', vmax=1,
vmin=-1)
            print("Classification report:" "\n", classification)


        for model, model_instance in models:
            accuracy = metrics.accuracy_score(Y_test,
model_instance.predict(X_test))
            confusion_matrix = metrics.confusion_matrix(Y_test,
model_instance.predict(X_test))
            classification = metrics.classification_report(Y_test,
model_instance.predict(X_test))
            print(f'{model} Model Test Results')
            print("Model Accuracy:" "\n", accuracy)
            print("Confusion matrix:" "\n", confusion_matrix)
            print("Classification report:" "\n", classification)
```

```
        from sklearn import naive_bayes

        BNB_Classifier = naive_bayes.BernoulliNB()
        BNB_Classifier.fit(X_train, Y_train)

        models = [('BNB Classifier', BNB_Classifier)]

        for model, model_instance in models:
            if model == 'LOF Classifier':
                scores = model_selection.cross_val_score(model_instance, X_train,
Y_train, cv=10, scoring='top_k_accuracy')
            else:
                scores = model_selection.cross_val_score(model_instance, X_train,
Y_train, cv=10)
            if model == 'LOF Classifier':
                accuracy = metrics.accuracy_score(Y_train,
model_instance.predict(X_train, novelty=True))
            else:
                accuracy = metrics.accuracy_score(Y_train,
model_instance.predict(X_train))
            confusion_matrix = metrics.confusion_matrix(Y_train,
model_instance.predict(X_train))
            classification = metrics.classification_report(Y_train,
model_instance.predict(X_train))

            print(f'{model} evaluation\n')
            print("Cross Validation Mean score:" "\n", scores.mean())
            print("Accuracy:" "\n", accuracy)
            print("Confusion matrix:" "\n", confusion_matrix)
            print("Classification report:" "\n", classification)
```

```
import logging
import sys
from logging.config import dictConfig

import fastapi
from starlette import responses
from starlette.middleware import cors

import analyze
import settings

dictConfig(settings.LOGGING_CONFIG)
uvicorn_logger = logging.getLogger('uvicorn')
uvicorn_handler = logging.StreamHandler(sys.stdout)
uvicorn_handler.setFormatter(settings.UvicornFormatter())
uvicorn_logger.addHandler(uvicorn_handler)

sqlalchemy_handler = logging.StreamHandler(sys.stdout)
sqlalchemy_logger = logging.getLogger('sqlalchemy.engine')
sqlalchemy_handler.setFormatter(settings.SQLAlchemyFormatter())
sqlalchemy_logger.addHandler(sqlalchemy_handler)

app = fastapi.FastAPI()



@app.get('/favicon.ico', include_in_schema=False)
async def favicon():
    return responses.FileResponse('favicon.ico')
```

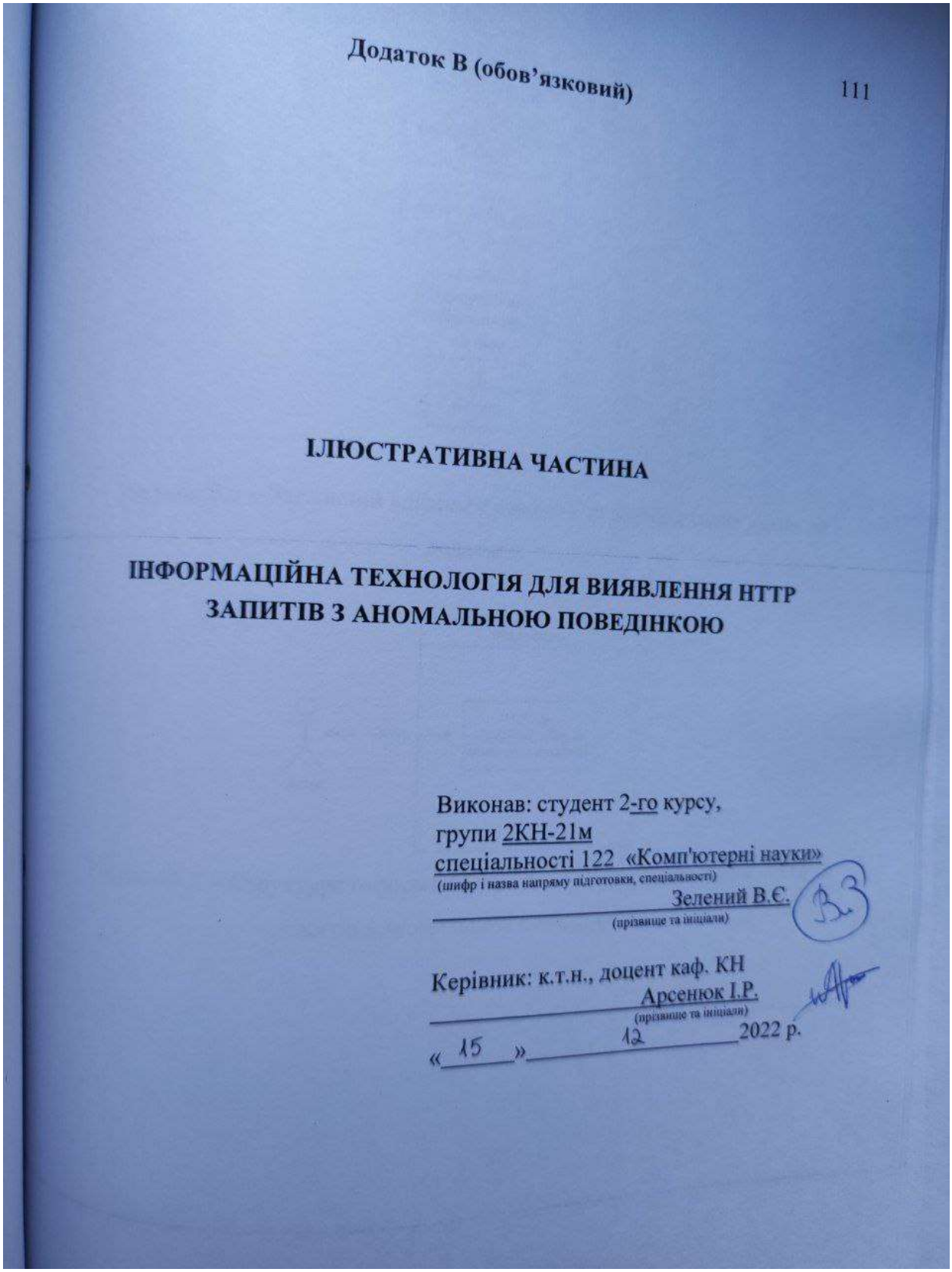
```
app.add_middleware(
    cors.CORSMiddleware,
    allow_origins=['http://127.0.0.1:8000', 'http://localhost:8000'],
    allow_credentials=True,
    allow_methods=['*'],
    allow_headers=['*']
)

app.include_router(analyze.router)

import logging
import pathlib

from starlette import templating

BASE_DIR = pathlib.Path(__file__).resolve().parent

TEMPLATES =
templating.Jinja2Templates(directory=str(pathlib.Path(BASE_DIR, 'templates')))



# Logging
class UvicornFormatter(logging.Formatter):

    def __init__(self):
        super().__init__(datefmt=None, style='{')

    GREY = "\x1b[37m"
    BLUE = "\x1b[36;1m"
    YELLOW = "\x1b[33;1m"
```

```
        RED = "\x1b[31;1m"
        BLACK = "\x1b[30;1m"
        RESET = "\x1b[0m"

        DEBUG_FORMAT = "[" + GREY + "{levelname}" + RESET + "] " +
GREY + "({asctime}) | {filename} | {name} | " + RESET \
                    + "{message} " + GREY + "| line: {lineno}" + RESET
        INFO_FORMAT = "[" + BLUE + "{levelname}" + RESET + "] " + GREY
+ "({asctime}) | {filename} | {name} | " + RESET \
                    + "{message} " + GREY + "| line: {lineno}" + RESET
        WARNING_FORMAT = "[" + YELLOW + "{levelname}" + RESET + "]
" + GREY + "({asctime}) | {filename} | {name} | " \
                     + RESET + "{message} " + GREY + "| line: {lineno}" + RESET
        ERROR_FORMAT = "[" + RED + "{levelname}" + RESET + "] " +
GREY + "({asctime}) | {filename} | {name} | " + RESET \
                    + "{message} " + GREY + "| line: {lineno}" + RESET
        CRITICAL_FORMAT = "[" + BLACK + "{levelname}" + RESET + "] " +
GREY + \
                      "({asctime}) | {filename} | {name} | " + RESET + "{message} "
+ GREY + "| line: {lineno}" + RESET

        def format(self, record):
            original_format = self._style._fmt  # Accessed protected members

            if record.levelno == logging.DEBUG:
                self._style._fmt = self.DEBUG_FORMAT
            elif record.levelno == logging.INFO:
                self._style._fmt = self.INFO_FORMAT
            elif record.levelno == logging.WARNING:
                self._style._fmt = self.WARNING_FORMAT
```

```
            elif record.levelno == logging.ERROR:
                self._style._fmt = self.ERROR_FORMAT
            elif record.levelno == logging.CRITICAL:
                self._style._fmt = self.CRITICAL_FORMAT

            result = logging.Formatter.format(self, record)
            self._style._fmt = original_format

            return result


    class SQLAlchemyFormatter(UvicornFormatter):
        DEBUG_FORMAT = "[" + UvicornFormatter.GREY + "{levelname}" +
UvicornFormatter.RESET + "] " + \
                       UvicornFormatter.GREY + "({asctime}) | {name} | " +
UvicornFormatter.RESET + "{message}"
        INFO_FORMAT = "S | [" + UvicornFormatter.YELLOW + "{levelname}"
+ UvicornFormatter.RESET + "] " + \
                      UvicornFormatter.GREY + "({asctime}) | {name} | " +
UvicornFormatter.RESET + "{message}"
        WARNING_FORMAT = "[" + UvicornFormatter.YELLOW +
"{levelname}" + UvicornFormatter.RESET + "] " + \
                         UvicornFormatter.GREY + "({asctime}) | {name} | " +
UvicornFormatter.RESET + "{message}"
        ERROR_FORMAT = "[" + UvicornFormatter.RED + "{levelname}" +
UvicornFormatter.RESET + "] " + \
                       UvicornFormatter.GREY + "({asctime}) | {name} | " +
UvicornFormatter.RESET + "{message}"
        CRITICAL_FORMAT = "[" + UvicornFormatter.BLACK + "{levelname}"
+ UvicornFormatter.RESET + "] " + \
```

```
                    UvicornFormatter.GREY + "({asctime}) | {name} | " +
UvicornFormatter.RESET + "{message}"


    LOGGING_CONFIG = {
        'version': 1,
        'disable_existing_loggers': False,
        'loggers': {
            'uvicorn': {
                'level': 'INFO',
                'propagate': True,
            },
            'sqlalchemy.engine': {
                'level': 'INFO',
                'propagate': True
            }
        }
    }
```

Додаток В (обов'язковий)



Додаток В (обов'язковий) 111

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ВИЯВЛЕННЯ HTTP ЗАПИТІВ З АНОМАЛЬНОЮ ПОВЕДІНКОЮ

Виконав: студент 2-го курсу,
групи 2КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
Зелений В.Є.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН
Арсенюк І.Р.
(прізвище та ініціали)
«15» 12 2022 р.

Рисунок В.1 – Загальний алгоритм надання та відображення даних веб-додатком

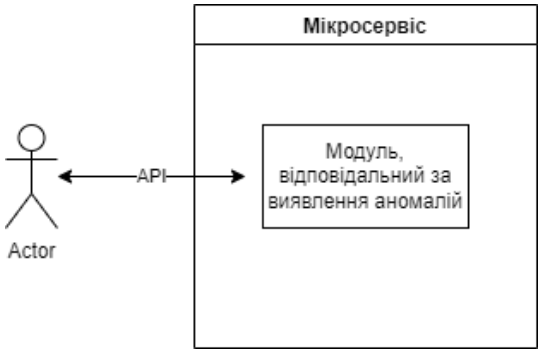


Рисунок В.2 – Структура інформаційної технології для виявлення HTTP запитів з аномальною поведінкою

Рисунок В.3 – Загальна схема алгоритму роботи інформаційної технології для виявлення HTTP запитів з аномальною поведінкою

Рисунок В.4 – Схема алгоритму роботи модуля, що аналізує дані з HTTP запитами

```
KNN Classifier evaluation

Cross Validation Mean score:
 0.9603352565525338
Accuracy:
 0.9703393178765181
Confusion matrix:
 [[   967     94    204     39     21     29     16     25      0      0]
 [   572    362     42     52    105     35     27     24     21      0]
 [    42      7 682325     55    846   2220     54    413     41      1]
 [   679    279    293    626   1170    149     97    153     16      1]
 [   682    282   1041    348  13150    304    178    357     72      0]
 [   683    267   3318    135    176   7717     56    239      0      0]
 [   691    263    258    215   1145    144    962     79     14      0]
```

Рисунок В.5 – Результат машинного навчання для k-NN моделі

```
Classification report:
                precision    recall  f1-score   support

      Analysis       0.19      0.69      0.30      1395
      Backdoor       0.20      0.29      0.24      1240
        Benign       0.99      0.99      0.99    686004
           DoS       0.39      0.18      0.25      3463
      Exploits       0.75      0.80      0.77     16414
       Fuzzers       0.70      0.61      0.65     12591
       Generic       0.67      0.26      0.37      3771
Reconnaissance       0.81      0.69      0.75      8144
     Shellcode       0.75      0.58      0.65       876
         Worms       0.55      0.06      0.10       104
```

Рисунок В.6 – Приклад класифікаційного звіту

```
✓   Decision Tree Classifier evaluation

    Cross Validation Mean score:
     0.9660818350038441
    Accuracy:
     0.9909441663646693
    Confusion matrix:
```

Рисунок В.7 – Класифікаційний звіт моделі дерева рішень

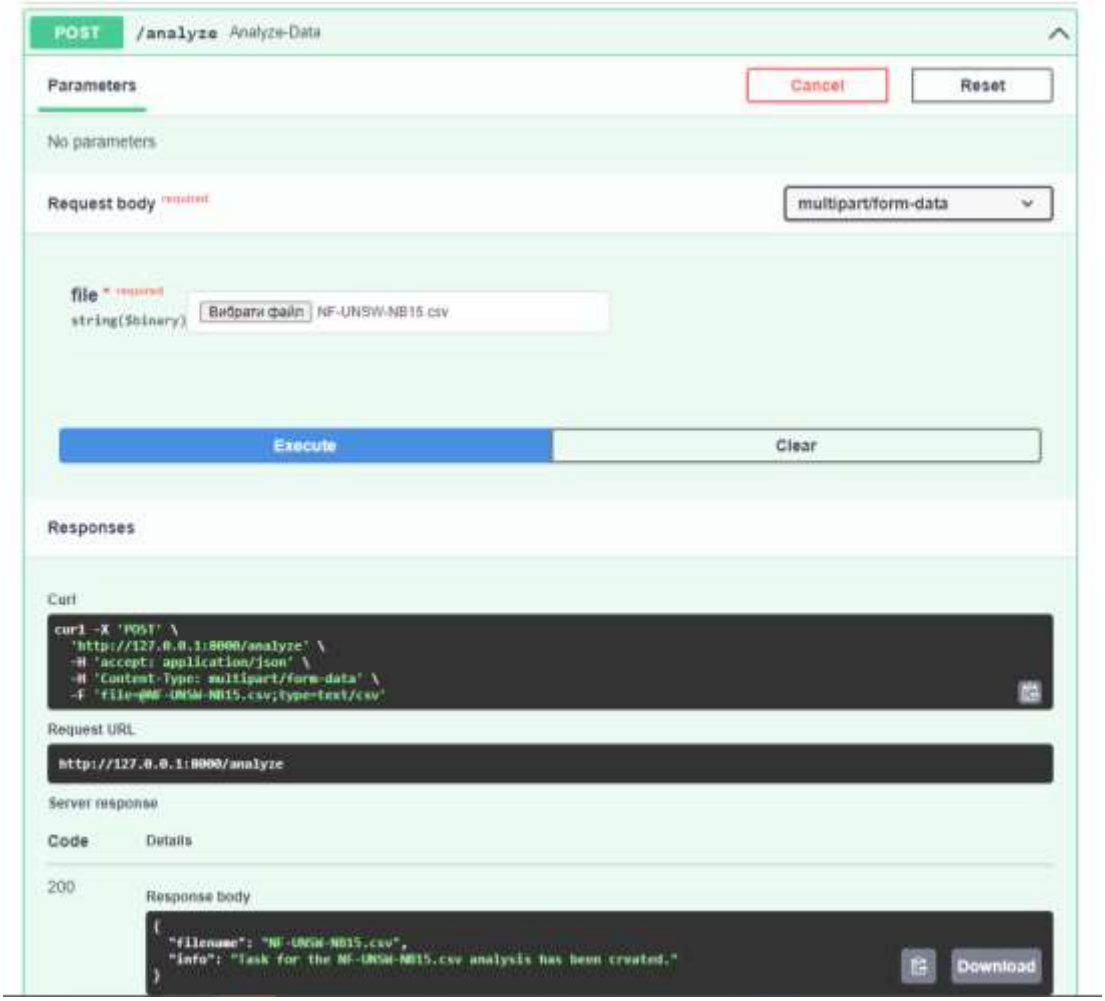


Рисунок В.8 – Приклад роботи API ендпоїнта

Додаток Г (довідниковий)

Інструкція користувача

Розглянемо два формати користування програмним продуктом, а саме зі сторони звичайного користувача, та з іншої – розробника, якому необхідно розвернути продукт та внести певні зміни.

Для типового користувача немає необхідності володіти технічними знаннями, щоб взаємодіяти з програмним продуктом. Користувачеві достатньо звернутись за налаштованим API ендпоїнтом, надіславши файл, який буде проаналізовано.

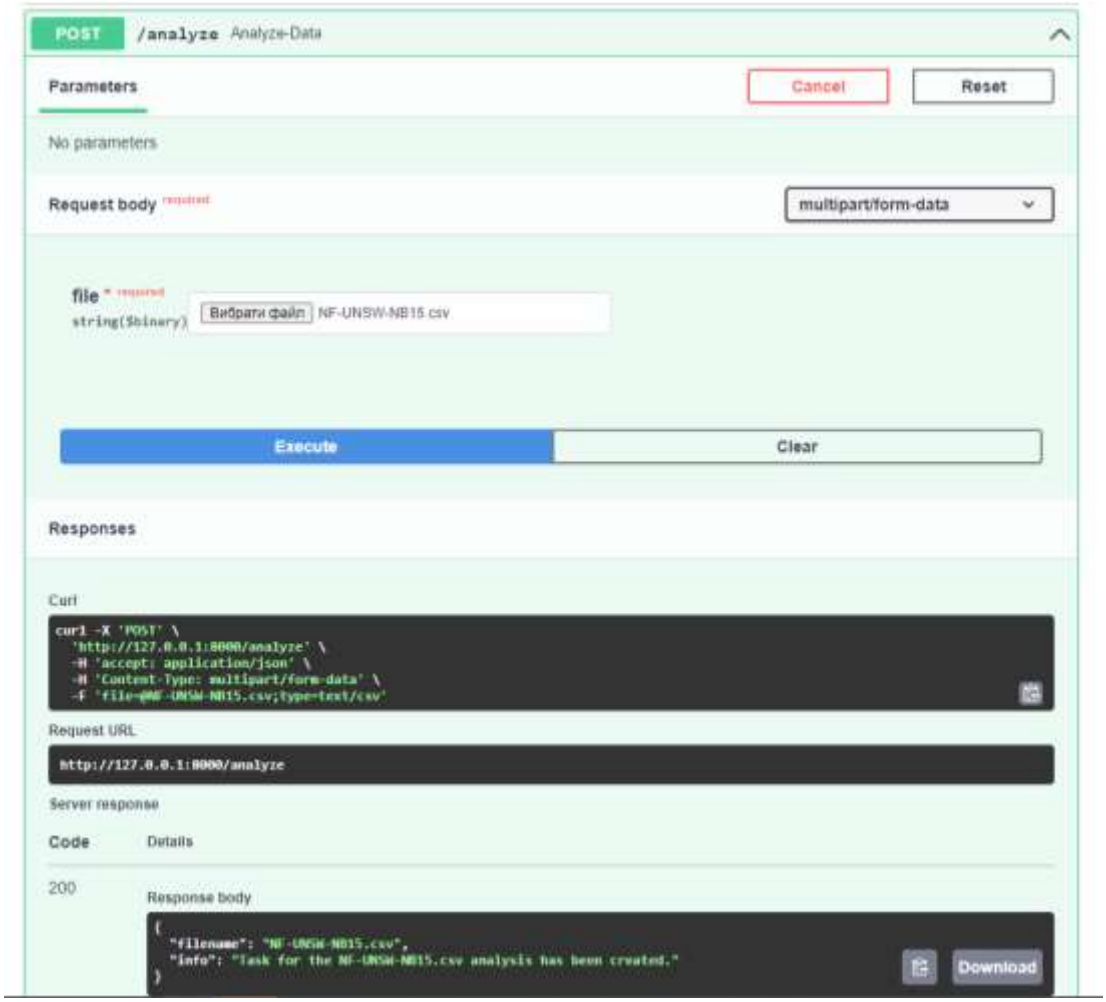


Рисунок Г.1 – Приклад роботи API ендпоїнта

Надалі, користувач у заданому форматі, наприклад, відповідь поштою, отримає у якості відповіді дані про трафік, що було надано у файлі.

Процес аналізу не потребує від користувача дій, тому на цьому його взаємодія з програмним продуктом завершується.

Розроблений програмний продукт потребує розвертання та внесення змін зі сторони розробника, тому варто розглянути способи роботи з продуктом.

Інформаційну технологію імплементовано на базі мікросервісу, а точніше – легковісного та швидкісного серверу, що розроблено використовуючи фреймворк FastAPI. Першочергово, розробнику потрібно підготувати код проєкту, наприклад, склонувати з репозиторію чи використати наданий лістинг.

Програмний продукт не прив’язано до певного середовища розробки, а тому розглянемо універсальний спосіб його запуску. Необхідно виконати наступні кроки:

1. Створити віртуальне середовище за допомогою команди «python3 -m venv /path/to/new/virtual/environment» та активувати його в залежності від операційної системи.
2. З файлу requirements.txt встановити усі необхідні бібліотеки та фреймоврки за допомогою команди «pip install -r requirements.txt».
3. У першому вікні терміналу запустити мікросервіс за допомогою команди «uvicorn app:app –reload».

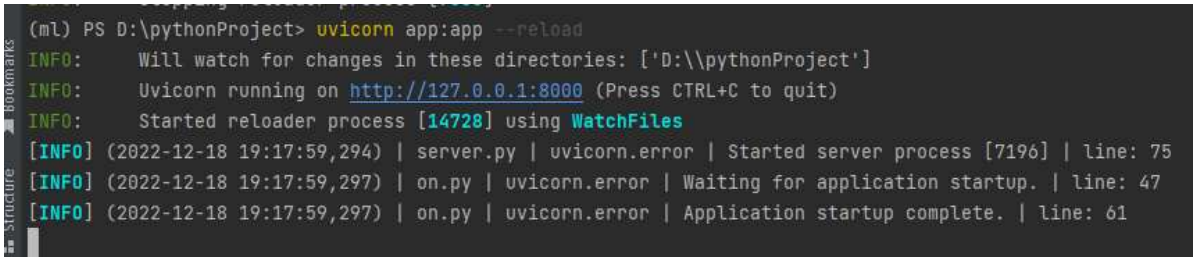


Рисунок Г.2 – Запуск серверу

4. У другому вікні терміналу запустити чергу завдань Celery за допомогою команди «celery -A celery_app worker --loglevel=INFO»

Для зручного корегування модулю, що аналізує дані, створити модуль з розширенням .ipynb. Це модуль формату, що підтримується Jupyter Notebook, в якому зручно покроково редагувати логіку технології, а код буде можливо запускати пострічково у окремих логічних клітинах.

Рисунок Г.3 – Приклад роботи коду у окремій логічній клітині