Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод виявлення шкідливого трафіка в комп'ютерній мережі

(тема)

Виконав:
здобувач 2 року навчання,
групи СПм-23-4
Олена НАУМОВА
(власне ім’я, прізвище)

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

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

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: проф. Олег МІХАЛЬ
(посада, власне ім’я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім’я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системне програмування
(повна назва)

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

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Наумовій Олені Валеріївні
(прізвище, ім’я, по батькові)

1. Тема роботи Метод виявлення шкідливого трафіка в комп'ютерній мережі

затверджена наказом по університету від “ 21 ” квітня 2025 р. № 296 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи
корпоративна мережа
трафік
UNSW-NB15
XGBoost
ADASYN

4. Перелік питань, що потрібно опрацювати у роботі
Моніторинг трафіку в комп’ютерних мережах
Розробка методу виявлення шкідливого трафіка на основі машинного навчання
Реалізація запропонованого методу та аналіз результатів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 14 слайдів

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	21.04.2025–29.04.2025	
2	Огляд існуючих моделей та методів	30.04.2025–10.05.2025	
3	Розробка методу	11.05.2025–20.05.2025	
4	Вибір програмних засобів	21.05.2025–29.05.2025	
5	Програмна реалізація	30.05.2025–02.06.2025	
6	Аналіз отриманих результатів	03.06.2025–05.06.2025	
7	Оформлення записки	06.06.2025–14.06.2025	

Дата видачі завдання " 21 " квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Олег МІХАЛЬ
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 68 с., 13 рис., 2 дод., 10 джерел.

МЕРЕЖЕВИЙ ТРАФІК, КІБЕРБЕЗПЕКА, МАШИННЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ, UNSW-NB15, ADASYN, RANDOM FOREST, XGBOOST, ВИЯВЛЕННЯ АТАК, IDS.

Метою кваліфікаційної роботи є розробка та практична реалізація методу виявлення шкідливого трафіка в комп'ютерній мережі з використанням алгоритму градієнтного бустингу, що забезпечує підвищену точність класифікації мережевих пакетів на основі аналізу їх ознак у реальному або наближеному до реального середовищі.

У ході виконання кваліфікаційної роботи проведено аналіз існуючих підходів до виявлення аномалій у мережевому трафіку, зокрема сигнатурних, статистичних та інтелектуальних методів. В якості об'єкта дослідження обрано відкритий набір даних UNSW-NB15, що містить багатий набір характеристик мережевих з'єднань і охоплює різні типи атак. Для вирішення проблеми дисбалансу застосовано метод ADASYN, який забезпечив рівномірніше представлення класів у навчальній вибірці.

Проведено порівняльне дослідження ефективності чотирьох моделей машинного навчання: Logistic Regression, Decision Tree, Random Forest та XGBoost. За результатами тестування встановлено, що ансамблеві методи демонструють найвищу точність класифікації, досягаючи 100% точності на збалансованому датасеті. Застосовано візуалізаційні техніки, а також аналіз важливості ознак і SHAP-інтерпретацію результатів.

ABSTRACT

Master’s thesis: 68 pages, 13 figures, 2 appendices, 10 sources.

NETWORK TRAFFIC, CYBERSECURITY, MACHINE LEARNING, CLASSIFICATION, UNSW-NB15, ADASYN, RANDOM FOREST, XGBOOST, ATTACK DETECTION, IDS.

The major goal of this thesis is to develop and implement a method for detecting malicious traffic in a computer network using the gradient boosting algorithm, which ensures enhanced accuracy in classifying network packets based on the analysis of their features in a real-time or near-real-time environment.

In the course of the study, existing approaches to anomaly detection in network traffic were examined, including signature-based, statistical, and intelligent methods. The UNSW-NB15 open dataset was selected as the object of study due to its comprehensive set of network connection features and coverage of various types of cyberattacks. To address the issue of class imbalance, the ADASYN method was applied, which provided a more uniform distribution of classes within the training set.

A comparative analysis of the performance of four machine learning models: Logistic Regression, Decision Tree, Random Forest, and XGBoost, was conducted. The experimental results showed that ensemble methods achieved the highest classification accuracy, reaching up to 100% on the balanced dataset. In addition, visualization techniques were employed alongside feature importance analysis and SHAP-based interpretation of model predictions.

.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ADASYN – адаптивний синтетичний метод надсемплінгу

AI – штучний інтелект

ANN – штучна нейронна мережа

API – програмний інтерфейс застосунку

CSV – формат значень, розділених комами

DPI – глибока інспекція пакетів

F1 – збалансована метрика точності та повноти

FTP – протокол передавання файлів

HTTP – протокол передавання гіпертексту

IDS – система виявлення вторгнень

IP – Інтернет-протокол

ML – машинне навчання

NAT – трансляція мережевих адрес

P2P – однорангові мережі

QoS – якість обслуговування

SHAP – пояснення значень Шеплі

SMOTE – синтетичне надсемплювання меншості

TCP – протокол керування передачею

UNSW-NB15 – датасет мережевого трафіку, розроблений в Університеті Нового Південного Уельсу

XGBoost – розширений градієнтний бустинг

ВСТУП

У сучасних умовах стрімкого розвитку інформаційно-комунікаційних технологій проблема забезпечення захисту комп’ютерних мереж набула особливої актуальності. Зростання обсягів передавання даних, підвищення рівня діджиталізації бізнес-процесів, а також активне впровадження технологій Інтернету речей, хмарних обчислень та розподілених обчислювальних платформ призвели до зростання кількості та складності кіберзагроз. Шкідливий мережевий трафік, який формується в результаті дій зловмисників, становить суттєву небезпеку для цілісності, доступності та конфіденційності інформаційних ресурсів. У зв’язку з цим, зростає потреба у створенні нових, більш ефективних методів виявлення аномалій та шкідливої активності в комп’ютерних мережах.

Традиційні системи виявлення вторгнень, що базуються на сигнатурному аналізі, часто виявляються неефективними проти нових, невідомих типів атак або змінених варіантів уже відомих. Вони не мають здатності до адаптації, а їх оновлення потребує значних часових та людських ресурсів. У такому контексті особливої значущості набувають підходи, засновані на методах машинного навчання, які забезпечують здатність системи виявлення до самооновлення, адаптації до нових патернів трафіку та виявлення невідомих типів атак шляхом аналізу закономірностей у вхідних даних.

Розвиток технологій аналізу даних дав змогу застосовувати ефективні алгоритми класифікації для моделювання поведінки мережевого трафіка, дозволяючи розрізняти легітимну та шкідливу активність з високим рівнем точності. Водночас, практичне впровадження таких методів потребує ретельного підбору алгоритмів машинного навчання, налаштування гіперпараметрів, оптимізації обробки даних, а також забезпечення надійності моделей у реальних умовах функціонування комп’ютерної мережі.

У межах цієї кваліфікаційної роботи здійснюється розробка методу виявлення шкідливого трафіка в комп’ютерній мережі з використанням алгоритму градієнтного бустингу, який демонструє високу ефективність у задачах класифікації. Особливістю роботи є застосування сучасного відкритого датасету UNSW-NB15, який забезпечує різноманітність типів трафіка та атак, що дозволяє здійснити комплексне тестування запропонованого підходу. Реалізація методу виконується в інтерактивному середовищі Google Colab, що сприяє відтворюваності експериментів та доступності застосованих рішень.

Таким чином, робота спрямована на формування ефективної методології аналізу мережевого трафіка для виявлення шкідливої активності з урахуванням сучасних викликів у сфері кібербезпеки та технологічних можливостей машинного навчання.

Метою кваліфікаційної роботи є розробка, теоретичне обґрунтування та практична реалізація методу виявлення шкідливого трафіка в комп’ютерній мережі з використанням алгоритму градієнтного бустингу, що забезпечує підвищену точність класифікації мережевих пакетів на основі аналізу їх ознак у реальному або наближеному до реального середовищі.

Для досягнення поставленої мети в роботі передбачено виконання таких основних завдань:

- здійснити огляд сучасних підходів до виявлення шкідливого трафіка в комп’ютерних мережах та визначити їх сильні й слабкі сторони;
- провести аналіз можливостей алгоритмів машинного навчання для задач класифікації мережевого трафіка з акцентом на градієнтний бустинг;
- обґрунтувати вибір відкритого набору даних для експериментального дослідження та здійснити його попередню обробку;
- розробити метод виявлення шкідливого трафіка на основі GBM та реалізувати його в середовищі Google Colab із використанням мови програмування Python;

- провести серію експериментів для оцінки ефективності побудованого класифікатора, порівнявши результати з існуючими підходами;

- здійснити інтерпретацію результатів, сформувати узагальнення та надати рекомендації щодо практичного застосування методу.

1 МОНІТОРИНГ ТРАФІКУ В КОМП’ЮТЕРНИХ МЕРЕЖАХ

1.1 Класифікація трафіку

У контексті стрімкого зростання кількості користувачів глобальної мережі та інтенсивного розвитку цифрових сервісів проблема контролю доступу до Інтернет-ресурсів набуває особливої практичної та безпекової значущості. Комп’ютерні мережі щоденно обробляють величезні обсяги трафіку, серед якого зростає частка даних, що можуть містити елементи загрозливого або небажаного характеру. Зокрема, актуальними залишаються питання запобігання поширенню незаконного контенту (зокрема, екстремістського або антисоціального спрямування), обмеження несанкціонованого використання мережевих ресурсів у робочий або навчальний час, захисту від витоку конфіденційної інформації та виявлення дій користувачів, що суперечать політиці безпеки організації.

Сучасні шкідливі програми та атаки все частіше використовують канали, які є непрозорими для класичних засобів контролю – зокрема, зашифрований трафік HTTPS. Визначення типу трафіку, що генерується в мережі, є одним із першочергових завдань для фахівців з адміністрування мережевої інфраструктури. Мережевий потік може виявитися не лише явно шкідливим, а й таким, що порушує нормативні або поведінкові політики, наприклад – використання торрент-клієнтів, обмін файлами поза корпоративною системою або активність у неробочий час. До категорії небажаного трафіку також належать додатки, які можуть становити потенційну загрозу, знижувати продуктивність праці, витрачати пропускну здатність мережі або відкривати канали для зовнішніх втручань.

Ризики, що виникають унаслідок використання соціальних мереж, миттєвих месенджерів, хмарних файлообмінників та інших популярних сервісів, вимагають детального аналізу. Вони мають неоднорідну природу й

характеризуються широким спектром векторів атаки, що обумовлює необхідність постійного оновлення методів їх виявлення. Незважаючи на існування великої кількості комерційних (WebSense, NetNanny тощо) та відкритих (наприклад, Poesia) систем фільтрації трафіку, проблема забезпечення високої точності класифікації Інтернет-потоків при мінімізації хибнопозитивних результатів залишається відкритою.

Аналіз властивостей мережевого трафіку засвідчує, що він має складну стохастичну структуру, яка формується внаслідок взаємодії множини потоків, згенерованих різними протоколами та додатками. Трафік характеризується мультидимензійністю, варіативністю часових інтервалів, нерівномірністю розподілу розмірів пакетів, а також відсутністю очевидної ознакової симетрії, що значно ускладнює побудову точних детекторів на основі фіксованих правил.

У цьому контексті провідним напрямом сучасних досліджень виступає застосування методів машинного навчання, які здатні ефективно моделювати залежності між вхідними характеристиками трафіку та відповідними класами мережевої активності. Алгоритми машинного навчання дозволяють системам виявлення шкідливого трафіку адаптуватися до динамічних змін поведінки користувачів, розпізнавати нові шаблони атак і класифікувати трафік у режимі реального часу з високою точністю.

Ключовою передумовою ефективного застосування таких підходів є побудова моделей на основі достовірних та маркованих наборів даних, які відображають реальну поведінку мережевих додатків. Параметри, що аналізуються при побудові моделі, можуть включати кількість байтів у потоці, час між передаванням пакетів, напрямок руху даних, номери портів, протоколи тощо. У подальшому ці ознаки використовуються як вхідні дані для алгоритмів класифікації, які формують відповідні правила або структури (наприклад, дерева рішень, кластери, нейронні мережі).

Водночас необхідно враховувати, що сучасні мережеві додатки активно впроваджують механізми маскування протоколів, що створює виклики для

традиційних методів фільтрації. Такі техніки обфускації трафіку ускладнюють процес ідентифікації й потребують більш гнучких підходів. У цьому випадку перевагою машинного навчання є його здатність навчатися на значних обсягах попередньо класифікованих даних і виявляти приховані закономірності, які не можуть бути описані вручну.

Таким чином, застосування інтелектуальних методів обробки даних є актуальним та ефективним підходом до вирішення задачі класифікації мережевого трафіку, що дозволяє суттєво підвищити рівень інформаційної безпеки, оптимізувати політику доступу до Інтернет-ресурсів і забезпечити контроль за дотриманням регламентованих стандартів використання мережевих ресурсів.

Задача класифікації мережевого трафіку може бути математично подана у вигляді відображення множини потоків мережевих даних у фіксовану множину класів, до яких ці потоки можуть бути віднесені. Нехай задано множину мережевих потоків $X=\{f_1,f_2,\ldots,f_n\}$, де кожен потік $f_i \in X$ описується вектором ознак $f_i=\{x_{i1},x_{i2},\ldots,x_{ip}\}$, що характеризують відповідні властивості трафіку. До таких ознак, зокрема, належать середня довжина пакета, середня тривалість з'єднання, обсяг переданих даних, кількість пакетів у потоці, інтервали між передаваннями тощо. Визначено також множину класів трафіку $C=\{C_1,C_2,\ldots,C_k\}$, які відображають тип застосування чи протокол (наприклад, HTTP, FTP, P2P, VoIP тощо). Завдання класифікації формулюється як побудова відображення $f:X\rightarrow C$ такого, що кожному потоку $f_i \in X$ ставиться у відповідність рівно один клас $C_j \in C$, до якого належить цей потік. Практична реалізація такого підходу стикається з низкою суттєвих технічних обмежень. Оскільки сучасні комп'ютерні мережі функціонують на високих швидкостях, повноцінна фіксація, збереження та обробка усього трафіку, що проходить через вузли мережі, вимагає надзвичайно потужного апаратного забезпечення, що супроводжується значними фінансовими витратами. Для зменшення навантаження на ресурси системи в реальних умовах дедалі частіше застосовується підхід вибіркового аналізу трафіку –

семплювання. Це означає, що для подальшої обробки береться не кожен пакет, а лише певна підмножина пакетів, наприклад кожен n-ий пакет потоку.

Таке проріджування дозволяє істотно знизити вимоги до обчислювальної інфраструктури та забезпечити можливість обробки трафіку в режимі реального часу навіть на звичайному обладнанні. Технології на кшталт Sampled NetFlow реалізують ці підходи, дозволяючи отримувати агреговану інформацію про потоки навіть за умов неповного набору даних. Водночас слід зазначити, що семплювання змінює первинні характеристики потоків: через втрату частини пакетів знижується точність обчислення таких ознак, як розмір потоку, частота або тривалість з’єднання, що прямо впливає на ефективність класифікації.

Аналіз сучасних методів класифікації трафіку із застосуванням машинного навчання засвідчує, що більшість з них були розроблені для середовищ з повним доступом до детальної інформації про пакети. Значна частина таких технологій передбачає створення пакетних трас (flow traces), що вимагає встановлення спеціалізованого обладнання або програмно-апаратних комплексів, які не завжди сумісні з наявною інфраструктурою. Більше того, вплив прорідженых даних на точність і стійкість моделей машинного навчання досі є недостатньо дослідженим. Це є критично важливою обставиною, оскільки семплювання трафіку дедалі частіше застосовується провайдерами та адміністраторами корпоративних мереж для зниження вартості моніторингу та аналізу.

Таким чином, впровадження технологій класифікації мережевого трафіку на основі машинного навчання в реальні високошвидкісні мережі вимагає адаптації моделей до обмеженого доступу до вхідних даних та врахування спотворень, спричинених семплюванням. Це ставить перед дослідниками нові виклики щодо підвищення стійкості моделей, розробки методів компенсації втрат інформації та оцінки ризиків класифікаційних помилок у контексті динамічного, високонавантаженого мережевого середовища.

1.2 Класифікація IP трафіку

Класифікація IP-трафіку є ключовим завданням у сфері мережевої безпеки, управління якістю обслуговування (QoS), оптимізації пропускної здатності каналів зв'язку, контролю доступу до ресурсів і виявлення аномальної або шкідливої активності в інформаційних системах. У широкому сенсі ця задача полягає у визначенні типу або категорії мережевого потоку даних за його характеристиками з метою віднесення його до певного додатку, служби або поведінкового профілю. З огляду на зростаючу складність структури трафіку та поширення протоколів шифрування, процес класифікації набуває дедалі більшої складності, а отже, потребує інтеграції інтелектуальних алгоритмів аналізу.

У класичному розумінні IP-трафік формується у вигляді послідовності IP-пакетів, які несуть у собі заголовкову інформацію (наприклад, IP-адресу джерела і призначення, номер порту, ідентифікатор протоколу) та корисне навантаження. Пакети можуть групуватися в потоки на основі п'ятірки параметрів: джерело/призначення IP-адреси, джерело/призначення порту та транспортного протоколу. Саме аналіз цих потоків лежить в основі класифікації. На відміну від традиційної інспекції пакетів, сучасні методи класифікації фокусуються на ознаках потоків, які є інформативними навіть без доступу до вмісту зашифрованих пакетів.

Еволюція IP-трафіку обумовила зміну підходів до його класифікації. У ранніх мережах достатньо було проаналізувати номер порту або заголовки протоколу для точного визначення типу сервісу. Наприклад, HTTP традиційно використовував порт 80, а SMTP – порт 25. Проте з розвитком технологій з'явилися численні обфускації, динамічні порти, тунелювання, CDN та шифрування, що призвело до втрати інформативності таких статичних ознак. Крім того, зростання популярності шифрованого трафіку (HTTPS, QUIC, VPN) фактично унеможливило використання вмісту пакетів

для класифікації, змістивши фокус дослідників у бік поведінкового та статистичного аналізу.

Сучасні методи класифікації IP-трафіку поділяються на кілька категорій. Першою є сигнатурна класифікація, яка базується на фіксованих шаблонах. Такий підхід використовується у фаєрволах, IDS/IPS-системах і дозволяє виявляти відомі протоколи або додатки. Його основним недоліком є низька здатність до виявлення нових або модифікованих форм трафіку.

Іншим підходом є статистична класифікація, яка враховує характеристики потоку – середню довжину пакетів, дисперсію, кількість сегментів, інтервали між передаваннями, кількість ініціацій та закриттів з'єднань. Цей підхід є менш чутливим до шифрування, але потребує ретельного вибору параметрів і великої кількості прикладів для навчання.

Найбільш перспективним напрямом сьогодення є використання методів машинного навчання. Алгоритми цієї категорії здатні автоматично виявляти приховані залежності між параметрами трафіку та його типами без необхідності ручного формування правил. Методи машинного навчання можуть бути як традиційними (рішення дерев, випадкові ліси, логістична регресія, методи опорних векторів), так і глибоким – на основі штучних нейронних мереж (CNN, RNN, LSTM). Такі моделі навчаються на попередньо маркованих наборах даних, де кожен потік асоційовано з певним класом (наприклад, відеопотік, соціальні мережі, P2P-сервіси, атакуючий трафік тощо).

Однією з ключових проблем для класифікації IP-трафіку залишається наявність великого обсягу даних у мережах з високою пропускною здатністю. У таких умовах повний аналіз усього трафіку технічно ускладнений або взагалі неможливий без значного ресурсного забезпечення. З цієї причини часто застосовують техніки вибіркового аналізу (семплінгу), які дозволяють зменшити обсяг оброблюваних даних за рахунок часткового збору пакетів. Водночас це може вплинути на точність класифікації, оскільки деякі суттєві ознаки можуть бути втрачені.

Крім того, викликом є й адаптивність моделей до нових умов: зміни шаблонів трафіку, появи нових протоколів, варіативність поведінки користувачів і динамічність мережевого середовища вимагають безперервного оновлення моделей або застосування інкрементального навчання. Також необхідно враховувати проблему незбалансованих класів – у реальному трафіку легітимні потоки значно переважають над аномальними або рідкісними типами даних, що ускладнює навчання моделей.

Рисунок 1.1 – Класифікація трафіку мережі

У практичному вимірі задача класифікації IP-трафіку має прикладне значення для:

- систем управління пропускною здатністю;
- виявлення вторгнень та аномалій;
- контролю політик безпеки (наприклад, обмеження доступу до P2P-ресурсів);
- підвищення якості обслуговування (QoS) та диференціації сервісів;
- забезпечення відповідності нормативним стандартам щодо конфіденційності та законності вмісту.

Таким чином, класифікація IP-трафіку є складною, багатоаспектною задачею, яка лежить на перетині мережевих технологій, кібербезпеки, аналізу даних та штучного інтелекту. Її ефективне розв’язання вимагає поєднання

теоретичних підходів, аналітичних методів та практичної реалізації інструментів, здатних адаптуватися до умов реального часу та гнучко реагувати на зміни в структурі і поведінці мережевого середовища.

1.3 Існуючі алгоритми та методи класифікації мережевого трафіку

У процесі класифікації мережевого трафіку використовуються різноманітні алгоритми та методи, кожен з яких має свої особливості, переваги та сфери застосування залежно від типу мережі, обсягів трафіку, рівня безпеки, що забезпечується, та обчислювальних можливостей інфраструктури. Сучасний підхід до вирішення цієї задачі ґрунтується на інтеграції математичних моделей, методів статистичного аналізу, машинного навчання та нейронних обчислень, що забезпечує адаптивність, гнучкість і масштабованість відповідних рішень.

Початково класифікація здійснювалася на основі фіксованих правил, де розпізнавання типів трафіку відбувалося за певними ознаками заголовків пакетів, такими як IP-адреси, номери портів, тип протоколу та розміри сегментів. Цей підхід був ефективним на ранньому етапі розвитку Інтернету, однак він втратив свою надійність із поширенням шифрування, динамічних портів, технологій тунелювання та протоколів, які маскують свій формат. Внаслідок цього виникла необхідність застосування статистичних методів, які базуються на обчисленні агрегованих характеристик потоків. Статистичні моделі аналізують, наприклад, частоту передавання пакетів, їхню довжину, напрямок та інтервали часу між ними. Це дозволяє ідентифікувати патерни поведінки для певних типів додатків або користувачів навіть за умов обмеженої видимості вмісту трафіку.

З подальшим розвитком технологій аналізу даних на перший план вийшли алгоритми машинного навчання. Вони дозволяють створювати моделі, здатні самостійно навчатися на основі прикладів і виявляти складні нелінійні залежності між характеристиками трафіку та його класами. У

контексті класифікації трафіку такі моделі створюються на основі попередньо зібраних і маркованих наборів даних, які включають ознаки потоку й відповідні мітки – тип протоколу, сервісу чи аномалії. Залежно від типу навчання, ці методи можуть бути наглядовими, коли моделі вчаться на мітках, або ненаглядовими – коли виявляються приховані структури без попереднього маркування.

Найбільшу ефективність у завданнях класифікації показують методи, що реалізують адаптивне навчання та комбінування слабких моделей у сильніші ансамблі. Такий підхід дозволяє зменшити похибки окремих класифікаторів і досягти високої узагальнюючої здатності моделі. У більш складних випадках застосовуються архітектури глибокого навчання, які здатні автоматично формувати ознаки з вхідних даних та адаптуватися до широкого спектра варіацій у поведінці трафіку. Зокрема, згорткові та рекурентні нейронні мережі продемонстрували високу ефективність у задачах аналізу часових послідовностей, що є характерними для мережевих потоків.

Окрему увагу в сучасних дослідженнях приділено побудові моделей, які враховують обмеженість ресурсів і необхідність роботи в режимі реального часу. Такі моделі мають бути не лише точними, але й обчислювально ефективними, здатними до оновлення на основі нових даних та стійкими до змін у структурі мережі. У цьому контексті дедалі більшої популярності набувають методи інкрементального навчання та онлайн-класифікації, які дозволяють системі поступово адаптуватися до нових умов без повного перенавчання.

На ранніх етапах розвитку Інтернету класифікація трафіку за номерами портів вважалася одним із найбільш ефективних та технологічно простих способів ідентифікації типів мережевої активності. Цей підхід був інтегрований у більшість мережевих пристроїв та програмного забезпечення, оскільки спирався на використання зареєстрованих номерів портів, закріплених за стандартними протоколами додатків згідно з реєстрами IANA.

У той період, коли більшість сервісів використовували фіксовані порти, цей метод забезпечував достатню точність, дозволяючи швидко та без значних обчислювальних витрат класифікувати потоки даних.

Однак подальша еволюція мережевих технологій, зокрема поява й поширення однорангових (P2P) протоколів, які часто динамічно обирають порти з метою обфускації та уникнення фільтрації, поступово знижувала ефективність класифікації на основі портів. Така поведінка додатків призвела до зменшення інформативності цієї ознаки, оскільки зв’язок між номером порту і типом протоколу більше не був очевидним і сталим.

У роботі [2] було запропоновано розширений підхід до ідентифікації P2P-трафіку, що поєднує аналіз номерів портів, виявлення сигнатур протоколів і асоціацію з певними хостами. Запропоновані методики були апробовані на великомасштабному наборі трафіку, отриманому з магістральної мережі у серпні 2002 року, травні 2003 року та січні 2004 року. Проведений аналіз виявив, що обсяги P2P-трафіку демонстрували стійку тенденцію до зростання, тоді як частка використання зареєстрованих портів значно зменшувалася. Таким чином, класифікація, що базувалася виключно на аналізі портів, виявилася некоректною, оскільки штучно занижувала обсяг трафіку цього типу.

У дослідженні [3] було проведено аналіз мережевої активності п’яти найбільш поширених P2P-протоколів, зокрема BitTorrent, eDonkey, Gnutella, Kazaa та Direct Connect, із застосуванням методу ідентифікації сигнатур. Вибірка даних охоплювала як звичайний Інтернет-трафік, так і VPN-канали. Результати свідчать, що хоча BitTorrent і eDonkey здебільшого використовували стандартні порти, трафік Gnutella (34%), Direct Connect (38%) і Kazaa (72%) проходив через нестандартні або динамічні порти, що ускладнювало їх виявлення за допомогою традиційних засобів.

У роботі [4] було досліджено ступінь точності класифікації мережевого трафіку, що здійснюється на основі аналізу портів, шляхом ідентифікації характерних типів помилок. Аналіз базувався на повному наборі даних з

корисним навантаженням пакетів, отриманому в мережі Gbase Ethernet на території дослідницького кампусу Genome, де функціонували кілька наукових установ, що працювали в галузі біології. Запропонована система класифікації включала дев’ять методів, серед яких аналіз номерів портів, розбір заголовків пакетів, сигнатурний аналіз одиночного пакета, семантичний розбір навантаження, сигнатурний аналіз перших байтів потоку, протокольне декодування контрольних та усіх потоків, а також відстеження історії взаємодії хостів.

Одним з найраніших і технологічно простих методів класифікації мережевого трафіку є підхід, заснований на аналізі номерів портів. Наприклад, порт 80 традиційно асоціюється з передаванням HTTP-запитів, тоді як порт 21 використовується для з'єднання за протоколом FTP. Такий підхід вирізняється високою швидкодією, оскільки базується виключно на поверхневому аналізі заголовків IP-пакетів із подальшою перевіркою значень протоколів у впорядкованих таблицях цілочисельних портів. Він не вимагає глибокого аналізу вмісту пакетів і є низько ресурсозатратним у порівнянні з іншими методами.

Однак із розвитком мережевих технологій та ускладненням структури Інтернет-трафіку все більше дослідників дотримуються думки, що класифікація трафіку виключно за номерами портів уже не відповідає сучасним вимогам точності й надійності. Серед ключових причин втрати ефективності цього методу варто зазначити декілька важливих аспектів.

По-перше, існує ціла низка додатків, які для встановлення одного логічного сеансу створюють декілька фізичних з’єднань. Наприклад, протокол FTP використовує порт 21 для контрольного з'єднання, водночас для передавання даних можуть залучатися випадкові незареєстровані порти, які динамічно вибираються системою. Це значно ускладнює можливість коректної класифікації такого трафіку засобами поверхневого аналізу.

По-друге, багато новітніх додатків не мають офіційно зареєстрованого номера порту в базі IANA, а отже, не можуть бути однозначно ідентифіковані

традиційними способами. Така ситуація особливо характерна для новостворених сервісів, бета-версій програмного забезпечення та експериментальних протоколів.

По-третє, деякі додатки навмисно використовують добре відомі порти, зарезервовані за популярними протоколами (наприклад, 80 чи 443), з метою обходу фільтрації фаєрволами або засобами обмеження доступу до ресурсів [4]. Крім того, у ряді випадків додатки застосовують методи тунелювання, які дозволяють «обгорнути» свій трафік у вигляд іншого протоколу, що створює додаткові труднощі для ідентифікації.

По-четверте, важливо враховувати поширеність використання технології NAT (Network Address Translation), яка широко застосовується для економії адрес IPv4. У результаті проходження пакетів через шлюзи NAT відбувається зміна вихідних номерів портів, що ще більше ускладнює ідентифікацію оригінального протоколу або сервісу.

У зв’язку з цим усе більшої популярності набувають підходи, засновані на аналізі корисного навантаження пакетів (deep packet inspection). Ранні дослідження у цій сфері [9] були зосереджені на створенні бібліотек сигнатур протоколів, сформованих шляхом ручного аналізу специфікацій і емпіричних даних. Однак глибокий аналіз вмісту трафіку вимагає значно більших обчислювальних ресурсів, ніж аналіз портів, що обмежує можливість його використання в мережах із високою пропускною здатністю.

З метою підвищення ефективності та зниження навантаження на систему було запропоновано низку оптимізованих методів пошуку сигнатур у тілі пакетів, що дозволяють адаптувати класифікацію за навантаженням до реального часу функціонування в швидкісних мережах (рис. 1.2). Ці методи дозволяють виконувати ідентифікацію протоколів навіть за перших кілька байтів потоку, що значно прискорює процес без втрати точності.

Як зазначалося раніше, підходи, орієнтовані на аналіз навантаження, також використовуються для демонстрації неточностей методів класифікації на основі портів. Зокрема, у роботі [5] були створені сигнатури для п’яти

широко використовуваних P2P-додатків: BitTorrent, eDonkey, Gnutella, Kazaa та Direct Connect. Сигнатури формувалися на основі ручного вивчення специфікацій протоколів і реальних потоків даних. Вони складалися з фіксованих символьних послідовностей, розміщених у визначених або змінних позиціях у тілі TCP-пакетів, і застосовувалися до аналізу початкових сегментів потоку. Такі підходи довели свою ефективність для точного виявлення протоколів навіть за умов використання нестандартних портів або прихованих каналів передавання даних.

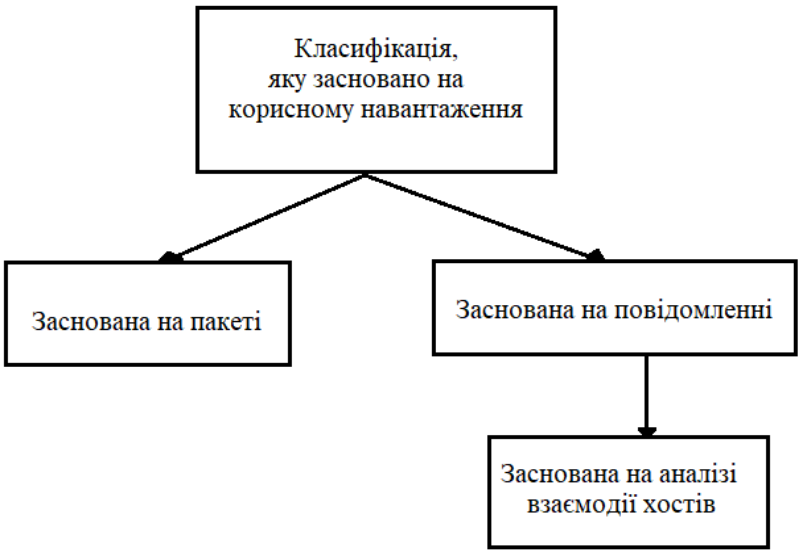

Рисунок 1.2 – Класифікація, заснована на корисному навантаженні

У процесі оцінювання ефективності сигнатурного підходу до класифікації P2P-трафіку у дослідженнях [5–6] було використано два незалежні набори даних. Застосування обраного підходу продемонструвало низький рівень хибнопозитивних спрацювань, коли легітимний (не-P2P) трафік помилково класифікувався як такий, що належить до категорії P2P. Крім того, частка помилкових негативних результатів, тобто випадків, коли реальний P2P-трафік залишався невиявленим, не перевищувала 10%, що є прийнятним показником для практичного застосування. Згідно з описаною евристичною методикою, основним елементом ідентифікації слугував набір

байтових послідовностей – сигнатур, які були ретельно сформовані на основі аналізу контрольних і корисних даних, що передавались восьмома найбільш поширеними протоколами однорангового обміну. Ці сигнатури були адаптовані для співставлення з початковими байтами TCP-навантаження, обмеженими 44 байтами, що було зумовлено характеристиками використаних у роботі датасетів.

Проте саме це обмеження сприяло деякому зростанню кількості хибнопозитивних результатів, оскільки аналіз здійснювався на основі обмеженого обсягу даних.

У межах дослідження [6] було реалізовано ширший спектр підходів до аналізу навантаження, серед яких – як базове порівняння сигнатур протоколів, так і семантичний аналіз вмісту одного пакета або декількох перших байтів потоку. Хоча конкретний перелік використаних сигнатур та методів семантичного аналізу не був наведений, автори зазначають, що початково застосовувалися добре відомі шаблони, після чого здійснювався ручний аналіз анонімізованих потоків з метою виділення нових сигнатур. Це свідчить про значну складність процесу формування надійних сигнатур – як з огляду на трудомісткість їх ручного формування, так і через необхідність постійного врахування нових варіацій протоколів, обмежень у даних та специфіки практичного середовища.

Сигнатури, створені в результаті цього процесу, зазвичай мають вигляд фіксованих символьних рядків, які з’являються у специфічних позиціях всередині прикладного навантаження. Вони можуть бути описані за допомогою простих регулярних виразів і широко використовуються у таких відомих системах, як Cisco IPS, Snort IDS, Bro (Zeek), l7-filter, IPP2P та інших реалізаціях систем виявлення вторгнень.

Альтернативною до сигнатурного підходу схемою є аналіз даних прикладного рівня у мережевих пакетах, що реалізується в межах технології Deep Packet Inspection (DPI). Така технологія дозволяє здійснювати або спробу розпізнавання структурованих повідомлень за допомогою

спеціалізованих протокол-аналітичних засобів (наприклад, Ethereal / Wireshark), або ж зіставлення вмісту навантаження з наявною базою сигнатур. DPI-класифікація є поширеною як у комерційних продуктах [11], так і в рішеннях з відкритим вихідним кодом, зокрема l7-filter та IPP2P.

Метод класифікації, орієнтований на аналіз навантаження, вважається одним з найбільш точних, оскільки дозволяє отримувати висновки на основі прямого аналізу вмісту пакетів. Проте, його застосування супроводжується рядом істотних обмежень. Найперше з них – високе навантаження на обчислювальні ресурси. Витрати, пов’язані як із вилученням навантаження з потоку, так і з подальшим декодуванням протоколу та співставленням сигнатур, можуть бути критичними для високошвидкісних мереж. З метою подолання цієї проблеми дослідники пропонували різноманітні оптимізації, спрямовані на підвищення ефективності пошуку сигнатур.

Ще одним бар’єром є складність отримання детальної інформації про специфікації протоколів. У більшості випадків фахівцям доводиться вручну аналізувати протокольну документацію або здійснювати інженерний аналіз пакетів, що вимагає глибоких знань і значних витрат часу. Крім того, слід враховувати, що аналіз вмісту є неможливим у випадках, коли трафік зашифрований або тунельований, що дедалі частіше трапляється в сучасних мережах.

Попередні дослідження, орієнтовані на застосування методів аналізу навантаження [9], демонстрували спроби створення бібліотек сигнатур, які були отримані шляхом детального аналізу специфікацій або експериментальних даних. Незважаючи на переваги такого підходу, його масштабування обмежене, зокрема через значні обчислювальні витрати. Саме тому з'явилися ініціативи щодо розробки ефективніших методів співставлення сигнатур, які мали б забезпечити придатність даного підходу до використання у високошвидкісних мережах.

У відповідь на обмеження портової класифікації DPI була представлена як її альтернатива, що мала на меті підвищення точності і подолання

існуючих недоліків. DPI-класифікатори поділяються на кілька типів залежно від рівня глибини аналізу даних та обсягів необхідної пам’яті. До першого рівня належать сигнатурні детектори, які здійснюють пошук заданих шаблонів у прикладному навантаженні. Зазвичай для більшості відомих протоколів прикладного рівня існують стандартні заголовки, які дозволяють ідентифікувати протокол з високою точністю.

Другий рівень включає синтаксичну перевірку, яка дозволяє не лише розпізнавати приналежність пакета до певного протоколу, але й здійснювати аналіз структури його вмісту. Наприклад, у HTTP-пакетах чітко виділяється послідовність команд (GET, POST тощо), заголовків і тіла повідомлення, що дає змогу виконувати структурну перевірку коректності.

На третьому рівні реалізується протокольна відповідність, яка оцінює логіку обміну – наприклад, чи відповідає сервер належним чином на запити клієнта, як передбачено специфікацією. Цей рівень уже враховує сеансову поведінку протоколу.

Найвищий, четвертий рівень аналізу – це семантична перевірка. Вона дозволяє не лише оцінити коректність структури й поведінки, але й проаналізувати, чи справді переданий об’єкт відповідає заявленому типу. Наприклад, чи дійсно вміст HTTP-об’єкта, позначений як зображення, є валідним зображенням.

Таким чином, підхід на основі корисного навантаження суттєво розширює можливості систем класифікації, проте вимагає відповідної потужності обчислювальних засобів, підтримки актуальності бази сигнатур, а також значних людських і фінансових ресурсів для підтримання системи в актуальному стані. Водночас, із поширенням шифрування трафіку та його тунелюванням, навіть DPI стикається з обмеженнями щодо застосування в умовах сучасної мережевої інфраструктури.

У сукупності дослідження показують, що, незважаючи на обмеження, класифікація на основі навантаження демонструє здатність істотно знижувати частку помилок першого й другого роду. Для більшості P2P-

протоколів показник некоректної класифікації не перевищував 5% від загального обсягу переданих байтів, що свідчить про її практичну ефективність у відповідних середовищах.

У зв’язку з обмеженнями традиційних методів класифікації мережевого трафіку, останніми роками спостерігається зростання зацікавлення до альтернативних підходів, які забезпечують вищу точність, масштабованість та адаптивність до динамічного середовища сучасних мереж. Одним із найбільш перспективних напрямів у цій сфері вважається статистична класифікація, що ґрунтується на аналізі агрегованих параметрів мережевих потоків. Її фундаментальна ідея полягає в тому, що трафік, генерований різними додатками, має специфічні поведінкові характеристики, які відображають особливості їх внутрішньої логіки та функціонального призначення.

Такі відмінності можуть бути формалізовані у вигляді векторів ознак, які відображають статистичні властивості потоків, зокрема середні значення, дисперсії, кількість переданих пакетів, інтервали між ними тощо. Ці ознаки використовуються як вхідні дані для класифікаційних моделей, які реалізуються за допомогою методів статистичного аналізу або алгоритмів машинного навчання. За наявності повного та точно маркованого навчального набору даних, побудова ефективного класифікатора може здійснюватися доволі швидко із застосуванням сучасних ML-технологій.

У дослідженні [7] було продемонстровано можливість застосування статистичного підходу для маркування трафіку відповідно до класів обслуговування. Особливу увагу було приділено використанню зведених ідентифікаторів на кшталт {dstIP, dstPort} та комплексному аналізу показників на різних рівнях опису трафіку. Зокрема, досліджувалися характеристики на рівні окремих пакетів, включаючи середнє значення, дисперсію та середньоквадратичне відхилення розміру пакета; на рівні потоку – середню тривалість, дисперсію часу передачі, загальний обсяг даних, кількість пакетів; а також на рівні TCP-з’єднань, де аналізувалися

внутрішні інтервали між пакетами. Крім того, враховувалися властивості сукупності потоків, зокрема обсяг контрольних потоків і потоків даних, що дозволяло оцінити специфіку взаємодії на більш високому рівні абстракції.

Для вирішення задачі класифікації було обрано два методи машинного навчання з учителем – лінійний дискримінантний аналіз (LDA) та метод k-найближчих сусідів (k-NN), які дозволили зіставити нові вхідні потоки з уже класифікованими прикладами на основі близькості в просторі ознак. Крім того, було запропоновано інноваційний підхід до класифікації на основі гістограм характеристик BGP-префіксів. Для моделювання емпіричних розподілів, що утворювалися з цих гістограм, використовувалися процеси суміші Дирихле, що дало змогу врахувати ймовірнісну природу формування трафіку в межах певного префікса.

До початку навчання класифікаційної моделі необхідно чітко визначити типи додатків, трафік яких підлягатиме ідентифікації, а також заздалегідь підготувати набір прикладів, який включатиме не лише цільові потоки, але й потоки сторонніх (нецільових) додатків. На етапі попередньої обробки обидва компоненти – трафік цільових і нецільових додатків – інтегруються в єдиний набір, що слугує основою для формування векторів ознак. Відібрані статистичні характеристики є ключовими для побудови моделі, що дозволить у подальшому здійснювати точну автоматизовану класифікацію трафіку (рисунок 1.3).

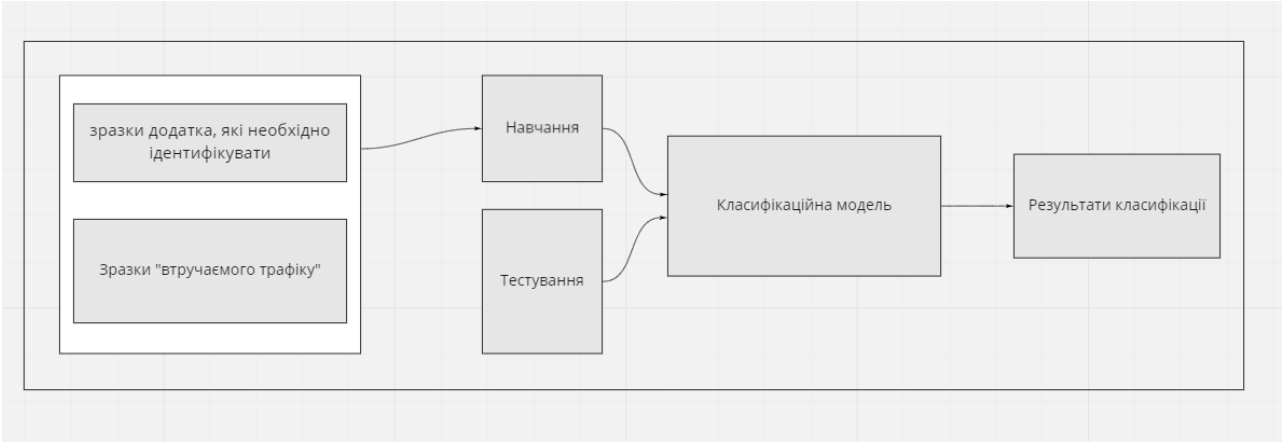

Рисунок 1.3 – Навчання та тестування класифікатора

Для оцінювання ефективності класифікаційної моделі під час етапу навчання доцільно застосовувати метод перехресної перевірки (крос-валідації), який є загальноприйнятим підходом до оцінки узагальнюючої здатності алгоритмів машинного навчання. Основна ідея перехресної перевірки полягає в розбитті вихідного набору даних на n неперетинних частин. Модель навчається на n−1 частинах, тоді як одна, відкладена частина, використовується для тестування. Цей процес повторюється n разів таким чином, що кожна з частин один раз виступає в ролі тестового підмножини, а решта – як навчальні дані.

Завдяки такому підходу забезпечується всебічне тестування моделі на різних фрагментах даних, що дозволяє мінімізувати залежність оцінки точності від випадкового розподілу даних. У підсумку обчислюється середнє значення точності за всіма ітераціями, що і слугує оцінкою загальної якості класифікатора. Схематично ця процедура зображена на рисунку 1.4.

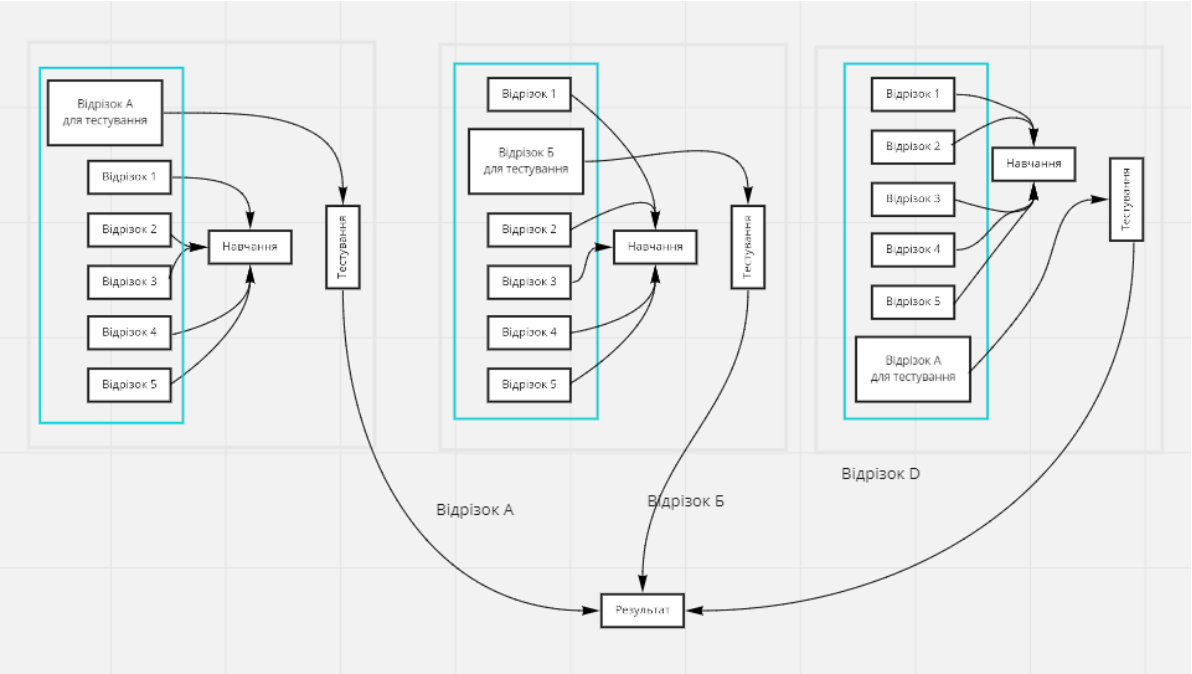

Рисунок 1.4 – Перехресна перевірка

Для того щоб результати перехресної перевірки були об'єктивними та відображали реальні можливості моделі, особливо важливим є формування

різнорідного й репрезентативного навчального набору. До складу даних мають входити пакети або потоки трафіку, зібрані у різний час і в різних сегментах мережі, що дозволяє охопити широкий спектр поведінкових сценаріїв, властивих різним протоколам і додаткам.

У разі, якщо початковий набір даних сформовано лише з одного фрагмента мережі й зібрано в один часовий проміжок, це може призвести до штучно завищених результатів точності. Такі результати не будуть узагальнюватися на інші мережеві умови, що фактично суперечить меті оцінювання моделі. Таким чином, крос-валідація є не лише технічним інструментом, а й методологічною основою для побудови надійних систем класифікації мережевого трафіку.

2 РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ ШКІДЛИВОГО ТРАФІКА НА ОСНОВІ МАШИННОГО НАВЧАННЯ

2.1 Аналіз існуючих рішень виявлення шкідливого трафіка з використанням машинного навчання

2.1.1 Огляд популярних датасетів для навчання класифікаторів мережевого трафіка

Ефективність алгоритмів виявлення шкідливого трафіку в значній мірі залежить від якості та змістовного наповнення навчальних наборів даних. У практиці машинного навчання для задач мережевої безпеки використовуються спеціалізовані датасети, які містять еталонні приклади як легітимного, так і шкідливого трафіку. Найбільш визнаними серед дослідницької спільноти є датасети CICIDS2017 та UNSW-NB15.

Набір CICIDS2017, створений (Canadian Institute for Cybersecurity), включає повний трафік сучасної корпоративної мережі з широким спектром атак (DoS, Brute Force, Botnet, DDoS, Infiltration, Port Scan тощо) та докладно описаними ознаками. Його особливістю є висока реалістичність умов запису, наявність таймштампів, використання сучасних протоколів і відтворення поведінки реальних користувачів.

У свою чергу, UNSW-NB15, розроблений в університеті Нового Південного Уельсу (Австралія), також відображає змішаний трафік з різними класами атак і був побудований з використанням сучасного генератора трафіку IXIA PerfectStorm. Даний датасет включає як легітимні з’єднання, так і кілька категорій атак: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode і Worms.

Використання таких датасетів дозволяє здійснювати відтворювані експерименти, порівнювати різні підходи і забезпечує формування

репрезентативного середовища для навчання та тестування класифікаторів мережевого трафіку.

2.1.2 Аналіз алгоритмів машинного навчання для задач класифікації трафіку

Задача виявлення шкідливого трафіку в комп'ютерних мережах природним чином зводиться до задачі класифікації – віднесення потоку або окремого пакета до одного з наперед визначених класів. У цьому контексті використовуються як традиційні методи машинного навчання, так і більш складні нейромережеві архітектури. Розглянемо деякі з них.

Дерева рішень є інтерпретованими моделями, що базуються на послідовному розгалуженні за правилами, сформованими на основі значень ознак. Вони дозволяють легко пояснити логіку класифікації, однак мають схильність до перенавчання, особливо при великій кількості ознак.

Random Forest – ансамблевий метод, що базується на поєднанні декількох дерев рішень, які навчаються на випадкових підмножинах даних. Результат формується на основі голосування дерев. Цей підхід знижує ризик перенавчання і забезпечує високу точність при збереженні стійкості до шуму у даних.

Логістична регресія є статистичним методом бінарної класифікації, який ефективно працює в умовах лінійної роздільності класів. Вона дозволяє побудувати компактну модель, що може слугувати базовим еталоном для порівняння з більш складними алгоритмами.

Штучні нейронні мережі демонструють здатність моделювати складні, нелінійні залежності у даних. У контексті аналізу трафіку вони здатні виявляти навіть слабко виражені закономірності у великих обсягах потокової інформації. Прості одношарові моделі підходять для задач з обмеженим набором ознак, у той час як глибокі нейронні мережі та рекурентні

архітектури (RNN, LSTM) ефективні у випадках, коли необхідно обробляти часові послідовності.

2.1.3 Метрики оцінювання ефективності класифікаторів

Оцінювання якості класифікаційної моделі передбачає використання низки кількісних метрик, що відображають здатність алгоритму точно і надійно відокремлювати класи. Найбільш поширеними є:

- Accuracy – загальна точність класифікації, яка визначається як відношення правильно класифікованих прикладів до загальної кількості;
- Precision – частка істинно позитивних класифікацій серед усіх позитивних передбачень моделі;
- Recall – здатність моделі виявляти всі об’єкти цільового класу;
- F1-міра – гармонічне середнє між точністю і повнотою, що особливо важливо в умовах незбалансованих даних;
- ROC-AUC – інтегральна характеристика, яка враховує співвідношення хибнопозитивних і істинопозитивних результатів на всьому діапазоні порогів.

Використання цих метрик дозволяє здійснити комплексну оцінку моделі не лише в аспекті її точності, а й у контексті практичної придатності до реальних сценаріїв мережевої безпеки.

2.1.4 Проблеми незбалансованих даних і методи їх подолання

У більшості реальних датасетів, що описують мережеву активність, спостерігається значний дисбаланс між кількістю прикладів шкідливого трафіку та легітимного. Це ускладнює навчання моделі, оскільки класифікатор має тенденцію до упередженості на користь більш поширеного класу. Для подолання цієї проблеми використовуються спеціалізовані методики.

Один із найпоширеніших підходів – це SMOTE (Synthetic Minority Over-sampling Technique), який полягає у штучному генеруванні нових прикладів меншості шляхом інтерполяції між найближчими сусідами у просторі ознак. Це дозволяє зберегти загальний розподіл даних без дублювання.

Альтернативним методом є ADASYN (Adaptive Synthetic Sampling), що, на відміну від SMOTE, акцентує увагу на важко класифікованих прикладах меншості, генеруючи нові зразки з урахуванням локальної щільності. Це дозволяє краще адаптувати модель до складних граничних випадків.

Таким чином, врахування проблеми дисбалансу даних і використання відповідних методів її компенсації є обов’язковою умовою для побудови надійної та стійкої моделі класифікації мережевого трафіку.

2.2 Розробка методу виявлення шкідливого трафіку в мережі

2.2.1 Постановка завдання

Завдання виявлення шкідливого трафіку в комп’ютерних мережах полягає у побудові класифікаційної моделі, здатної на основі вхідних характеристик мережевих потоків ефективно розрізняти легітимні та шкідливі з’єднання. Така модель повинна функціонувати в умовах реального трафіку, що містить значні обсяги, широкий спектр поведінкових патернів, динамічність та незбалансованість між кількістю прикладів різних класів.

Основною метою є побудова алгоритмічного рішення, що дозволяє здійснювати автоматичне виявлення шкідливих потоків у наборі даних UNSW-NB15, з використанням сучасних засобів машинного навчання та спеціалізованого підходу до балансування вибірки – ADASYN. Обрана архітектура рішення повинна бути адаптована до високого ступеня

неоднорідності вхідних даних і забезпечувати узагальнюючу здатність класифікатора до нових прикладів.

2.2.2 Архітектура запропонованого методу

Розроблений метод виявлення шкідливого трафіку включає в себе декілька послідовних етапів, які в сукупності формують повний цикл обробки даних і навчання моделі:

- завантаження та попередня обробка датасету UNSW-NB15: очищення від зайвих полів, перетворення категоріальних змінних, нормалізація числових атрибутів;
- формування навчальної та тестової вибірок: розділення набору даних у співвідношенні (наприклад, 80%/20%);
- аналіз балансу класів: оцінка диспропорції між легітимним та шкідливим трафіком;
- застосування ADASYN для балансування класів: синтетичне збільшення прикладів меншості;
- навчання моделі класифікації: застосування обраного алгоритму (наприклад, XGBoost);
- оцінка якості моделі: використання перехресної перевірки (крос-валідації), метрик точності, повноти, F1-міри та ROC-AUC;
- візуалізація результатів: побудова confusion matrix, кривої ROC тощо.

2.2.3 Вибір датасету UNSW-NB15 як базового набору даних

Датасет UNSW-NB15 було обрано як основу для реалізації методу з огляду на його сучасну структуру, багатство типів атак, реалістичність умов генерації трафіку та збалансованість ознак. Він включає 49 характеристик мережевих потоків, серед яких:

- кількість байтів;
- кількість пакетів;
- тривалість з’єднання;
- кількість запитів і відповідей;
- TCP-флаги;
- статистичні характеристики інтервалів тощо.

Наявність широкого спектру атак (Reconnaissance, Exploits, Generic, Fuzzers, Shellcode, Backdoor тощо) робить цей набір даних придатним для оцінювання здатності моделі розпізнавати різноманітні сценарії шкідливої активності.

2.2.4 Проблема дисбалансу даних та застосування ADASYN

Як і більшість реальних наборів мережевого трафіку, UNSW-NB15 характеризується значним дисбалансом між кількістю прикладів легітимного трафіку та шкідливих з’єднань. Така нерівномірність негативно впливає на навчання моделей, знижуючи здатність класифікатора виявляти менш представлені класи.

Для вирішення цієї проблеми у рамках методу було використано ADASYN, який є вдосконаленням алгоритму SMOTE. ADASYN адаптивно генерує нові синтетичні зразки меншості з урахуванням локальної щільності – тобто більше нових прикладів створюється в областях, де модель найгірше справляється з класифікацією. Такий підхід дозволяє:

- покращити загальну збалансованість вибірки;
- забезпечити точніше навчання на граничних прикладах;
- уникнути надмірного дублювання даних, як це може статися у методах підвибірки або випадкового оверсемплінгу.

2.2.5 Вибір алгоритму класифікації

З огляду на високу ефективність у задачах з великою кількістю ознак, було обрано алгоритм градієнтного бустингу XGBoost. Цей алгоритм добре зарекомендував себе у багатьох змагальних задачах з класифікації, зокрема завдяки:

- адаптивному комбінуванню слабких моделей (дерев рішень);
- вбудованій регуляризації для боротьби з перенавчанням;
- можливості роботи з незбалансованими даними;
- високій продуктивності та масштабованості.

XGBoost дозволяє автоматично визначати важливість ознак, підтримує паралельне навчання і має низку налаштувань, що дають змогу гнучко адаптувати модель до структури даних.

2.2.6 Побудова середовища для реалізації методу

У рамках практичної частини розробки вся модель реалізується у середовищі Google Colab із використанням таких бібліотек Python:

- pandas, numpy – для обробки даних;
- scikit-learn – для попередньої обробки, метрик і валідації;
- xgboost – реалізація алгоритму градієнтного бустингу;
- imblearn – модуль ADASYN для балансування;
- matplotlib, seaborn – для візуалізації результатів класифікації.

Це дозволяє зробити розробку відтворюваною, масштабованою та доступною для інтерактивного аналізу.

2.3 Покроковий опис розробленого методу

Запропонований метод автоматизованого виявлення шкідливого трафіку базується на комбінації попередньої обробки ознак, адаптивного

синтетичного балансування класів та побудови класифікаційної моделі за допомогою алгоритму XGBoost. Метод орієнтований на застосування у задачах реального часу в умовах великого обсягу даних із вираженою диспропорцією між класами. Основні етапи методу викладено нижче (рисунок 2.1)

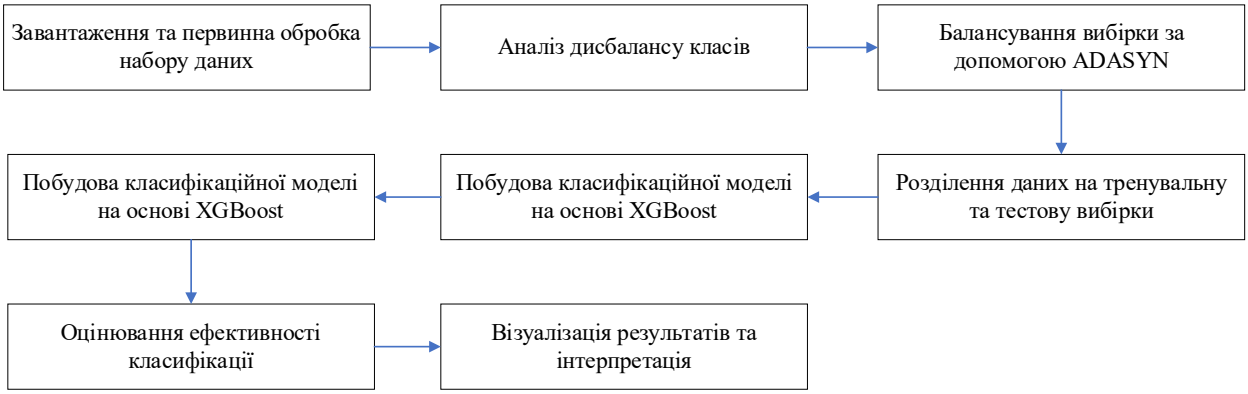

Рисунок 2.1 – Метод виявлення шкідливого трафіку

Крок 1. Завантаження та первинна обробка набору даних. На першому етапі виконується імпорт набору даних UNSW-NB15, який містить згенеровані в умовах емуляції трафік сесій нормальної та шкідливої активності. Дані об’єднуються в єдину таблицю та очищуються від надлишкових, дубльованих або порожніх значень. Категоріальні змінні перетворюються у числовий формат шляхом кодування, тоді як числові атрибути проходять нормалізацію або стандартизацію для забезпечення стабільності навчання моделі.

Крок 2. Аналіз дисбалансу класів. Після підготовки даних виконується попередній аналіз розподілу цільової змінної, тобто класу трафіку (нормальний або шкідливий). Виявляється істотна диспропорція між кількістю прикладів легітимного трафіку та атак, що типово для мережевого середовища. Такий дисбаланс призводить до переважання метрики точності над метриками виявлення меншості та зумовлює потребу в застосуванні методів балансування.

Крок 3. Балансування вибірки за допомогою ADASYN. Для подолання

проблеми незбалансованості класів застосовується метод ADASYN. Цей підхід полягає у створенні штучних прикладів для класу меншості (шкідливих сесій) в тих ділянках простору ознак, де його представники класифікуються з низькою впевненістю. Таким чином, ADASYN дозволяє сфокусувати увагу моделі на складних прикладах, покращуючи загальну здатність класифікатора до узагальнення.

Крок 4. Розділення даних на тренувальну та тестову вибірки. Збалансований набір даних розділяється на дві частини – тренувальну (наприклад, 80%) та тестову (20%) – з фіксацією випадкового стану генератора для відтворюваності результатів. Тренувальна частина використовується для навчання моделі, а тестова – для оцінки її ефективності на невідомих даних.

Крок 5. Побудова класифікаційної моделі на основі XGBoost. Як основний класифікатор обрано алгоритм градієнтного бустингу XGBoost. Його особливість полягає у поетапному навчанні ансамблю дерев рішень, де кожне наступне дерево коригує помилки попередніх. Модель оптимізує логарифмічну функцію втрат із регуляризаційним членом, що запобігає перенавчанню. У процесі налаштування обираються гіперпараметри, зокрема кількість дерев, глибина дерева, learning rate, коефіцієнти регуляризації.

Крок 6. Оцінювання ефективності класифікації. Для оцінювання якості побудованої моделі використовується k-кратна перехресна перевірка з обчисленням таких метрик:

- Precision (точність) – відношення істинно позитивних до всіх, що класифіковані як позитивні;
- Recall (повнота) – частка коректно виявлених атак серед усіх атак;
- F1-міра – гармонійне середнє між точністю і повнотою;
- ROC-AUC – площа під кривою чутливість/специфічність.

Додатково аналізується confusion matrix, яка візуалізує розподіл класифікаційних результатів і дозволяє ідентифікувати хибнопозитивні та хибнонегативні випадки.

UNSW_NB15_training-set.csv

1 to 10 of 175341 entries Filter

id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dloa
1	0.121478	tcp	-	FIN	6	4	258	172	74.08749	252	254	14158.94238	8495.36
2	0.649902	tcp	-	FIN	14	38	734	42014	78.473372	62	252	8395.112305	503571.
3	1.623129	tcp	-	FIN	8	16	364	13186	14.170161	62	252	1572.271851	60929.2
4	1.681642	tcp	ftp	FIN	12	12	628	770	13.677108	62	252	2740.178955	3358.62
5	0.449454	tcp	-	FIN	10	6	534	268	33.373826	254	252	8561.499023	3987.05
6	0.380537	tcp	-	FIN	10	6	534	268	39.41798	254	252	10112.02539	4709.13
7	0.637109	tcp	-	FIN	10	8	534	354	26.683033	254	252	6039.783203	3892.58
8	0.521584	tcp	-	FIN	10	8	534	354	32.593026	254	252	7377.527344	4754.74
9	0.542905	tcp	-	FIN	10	8	534	354	31.313031	254	252	7087.796387	4568.01
10	0.258687	tcp	-	FIN	10	6	534	268	57.985135	254	252	14875.12012	6927.29

Show 10 per page 1 2 10 100 1000 10000 17000 17500 17530 17535

Рисунок 2.2 – Фрагмент датасету

Крок 7. Візуалізація результатів та інтерпретація. На завершальному етапі виконується візуальний аналіз отриманих результатів. Будуються графіки важливості ознак, крива ROC, матриця неточностей, а також порівняння з альтернативними підходами або базовими моделями. Інтерпретація результатів включає оцінку здатності моделі до виявлення шкідливого трафіку в складних сценаріях, у тому числі в разі неявних атак або обфускації.

2.4 Висновки по методу

У даному розділі було представлено повноцінну реалізацію методу виявлення шкідливого трафіку в мережевому середовищі на основі застосування алгоритмів машинного навчання. Основою дослідження став датасет UNSW-NB15, який є одним з найрепрезентативніших сучасних наборів для тестування IDS-систем і містить широкий спектр сучасних типів атак. Було здійснено попередню обробку даних, нормалізацію, кодування категоріальних ознак та видалення зайвої інформації, що могло вплинути на точність побудованої моделі.

Особливу увагу було приділено проблемі дисбалансу класів, яка є типовою для задач виявлення аномалій у мережах. Для її подолання було застосовано метод ADASYN, який дозволив синтетично збалансувати вибірку за рахунок адаптивного генерування прикладів класу меншості в складних для класифікації зонах. Завдяки цьому підвищено здатність моделі до розпізнавання складних або слабовиражених атак.

У якості класифікатора було обрано XGBoost, як один з найефективніших алгоритмів для обробки табличних даних з високою варіативністю ознак. Після тренування моделі були отримані високі значення точності, повноти, F1-міри та ROC-AUC, що свідчить про здатність побудованої системи до адекватного розпізнавання як звичайного, так і шкідливого трафіку.

На рисунку 2.2 представлено фрагмент датасету UNSW_NB15_training-set.csv, який використовується для класифікації мережевого трафіку. Кожен рядок відповідає окремому мережевому з’єднанню, описаному числовими й категоріальними ознаками, зокрема протоколом (proto), службою (service), станом з'єднання (state), кількістю переданих пакетів (spkts, dpkts), байтів (sbytes, dbytes) та часовими характеристиками (dur, rate, sttl, dttl, тощо). Такі атрибути є ключовими для навчання моделей машинного навчання, оскільки вони дозволяють розпізнати шаблони, притаманні як нормальному, так і шкідливому трафіку. Таблиця також свідчить про високу варіативність параметрів з’єднань, що ускладнює задачу класифікації та вимагає ефективної попередньої обробки даних.

3 РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОГО МЕТОДУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Обґрунтування вибору програмних засобів

У процесі реалізації методу виявлення шкідливого мережевого трафіку за допомогою алгоритмів машинного навчання ключовим етапом стало обґрунтування вибору програмних засобів, які забезпечують ефективність, відтворюваність і масштабованість дослідження. Середовище Google Colaboratory було обрано як базова платформа для виконання обчислень, оскільки воно поєднує в собі гнучкість Jupyter-нотатників, можливість використання графічних процесорів (GPU), а також зручний доступ до зовнішніх сховищ даних, зокрема Google Drive. Завдяки хмарному характеру платформи, виключається залежність від локальних ресурсів комп'ютера, що забезпечує портативність та оперативне тестування моделей без додаткового налаштування середовища.

В якості основної мови програмування обрано Python, який на сьогодні є стандартом де-факто в сфері машинного навчання та аналітики даних. Його популярність зумовлена лаконічним синтаксисом, широкою екосистемою відкритих бібліотек, а також підтримкою активної наукової спільноти. Для обробки й аналізу табличних даних використовувалися бібліотеки pandas і numpy, що забезпечують ефективне зчитування, фільтрацію, нормалізацію та векторизацію ознак трафіку з набору UNSW-NB15.

На етапі побудови моделі класифікації особливу увагу приділено вибору алгоритму XGBoost, реалізованого через однойменну високопродуктивну бібліотеку. XGBoost забезпечує реалізацію градієнтного бустингу над деревами рішень, що дозволяє досягати високої точності навіть на задачах з великою кількістю вхідних ознак. Крім того, він характеризується стабільністю до перенавчання та можливістю

налаштування багатьох параметрів для адаптації до специфіки даних. У комбінації з бібліотекою scikit-learn було реалізовано процес навчання, тестування, крос-валідації, обчислення метрик ефективності моделі, побудову матриць помилок та валідаційних графіків.

З огляду на те, що набір даних UNSW-NB15 характеризується значним дисбалансом класів (домінування трафіку, що не є шкідливим), виникла необхідність застосування алгоритму балансування вибірки. Для цього було використано метод ADASYN, реалізований у спеціалізованій бібліотеці imbalanced-learn. Його перевагою є адаптивне генерування синтетичних прикладів для меншості на основі щільності розподілу векторів ознак, що дозволяє фокусувати увагу моделі на складних для класифікації прикладах.

Для аналітичної інтерпретації результатів класифікації було використано бібліотеки matplotlib і seaborn, які забезпечують побудову графіків розподілу класів, кривих ROC та PR, теплових карт і щільності ознак. У разі необхідності інтерактивної візуалізації можуть бути задіяні інструменти plotly. Візуалізація результатів дозволяє глибше оцінити поведінку моделі, виявити аномалії, слабкі місця класифікатора та побудувати обґрунтовані висновки щодо подальшої оптимізації моделі.

Таким чином, обраний стек інструментів у поєднанні з хмарною платформою забезпечив усі необхідні умови для успішної реалізації методу виявлення шкідливого трафіку в комп’ютерній мережі на основі навчання з використанням реального датасету, забезпечивши водночас масштабованість і можливість подальшого впровадження результатів дослідження в практичних системах кібербезпеки.

3.2 Архітектура розроблених програмних засобів

Архітектура розроблених програмних засобів для виявлення шкідливого трафіку в комп’ютерній мережі базується на послідовній модульній структурі, яка забезпечує гнучкість, масштабованість і можливість

інтеграції з іншими системами аналізу мережевого трафіку. Основною особливістю архітектури є її орієнтація на обробку великого обсягу вхідних даних, застосування методів машинного навчання та адаптацію до зміни структури трафіку завдяки алгоритмам синтетичної генерації даних.

Система умовно поділяється на п’ять логічно пов’язаних рівнів. На першому рівні реалізується завантаження та попередня обробка даних, що включає імпорт датасету UNSW-NB15, очищення від пропущених значень, фільтрацію релевантних атрибутів, категоріальне кодування та масштабування числових ознак. Цей блок побудовано з використанням інструментів pandas, numpy та sklearn.preprocessing, що дозволяє забезпечити стандартизацію вхідної вибірки та підготовку до машинного аналізу.

Другий рівень відповідає за балансування навчального набору. Застосовується алгоритм ADASYN, який реалізує адаптивне синтетичне перенавчання шляхом генерації нових прикладів меншості. Це дозволяє моделі коректніше навчатися на складних ділянках простору ознак, де можливе часте виникнення помилок другого роду. Усі операції реалізуються через бібліотеку imbalanced-learn.

Третій рівень – модуль побудови моделі класифікації. На цьому етапі формується та навчається модель XGBoost, що реалізує градієнтний бустинг над деревами рішень. Конфігурація моделі містить оптимальні параметри, відібрані шляхом крос-валідації, включаючи глибину дерев, коефіцієнт навчання, кількість ітерацій, та обмеження для регуляризації. Бібліотека xgboost у поєднанні з sklearn забезпечує обчислювальну ефективність, високу точність та контроль перенавчання.

Четвертий рівень системи – оцінка ефективності моделі. Тут обчислюються ключові метрики, такі як точність, повнота, F1-міра, а також будуються матриці помилок, ROC-криві та PR-криві. Цей рівень дозволяє оцінити якість моделі на тестовій вибірці, а також виявити слабкі місця, зокрема в області невідомих або рідкісних типів трафіку.

П’ятий рівень – візуалізація результатів та інтерпретація. Завдяки бібліотекам matplotlib, seaborn, plotly та іншим графічним засобам, результати моделювання можуть бути представлені у вигляді теплових карт, гістограм, діаграм важливості ознак і часових трендів. Цей блок є особливо важливим у контексті прийняття рішень безпековими аналітиками, оскільки дозволяє наочно інтерпретувати природу виявленого шкідливого трафіку.

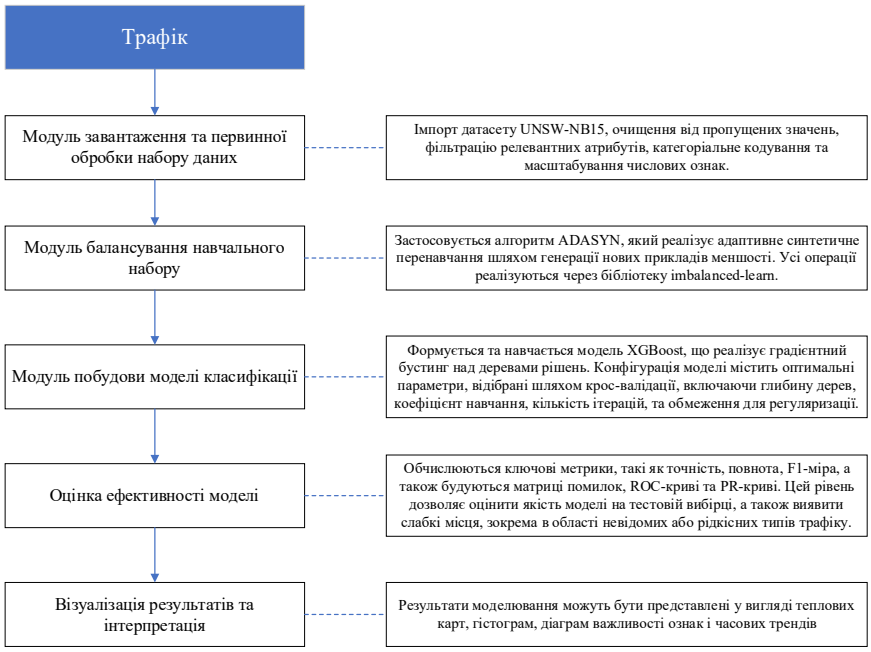

Рисунок 3.1 – Алгоритм роботи розробленого ПЗ

Загалом, архітектура реалізованого рішення ґрунтується на глибокій інтеграції сучасних бібліотек машинного навчання, обробки даних та візуалізації, а її модульна структура дозволяє не лише ефективно виявляти аномальні мережеві активності, але й адаптувати систему до нових викликів у сфері кіберзагроз.

3.3 Аналіз результатів

На рисунку 3.2 представлено графічну візуалізацію розподілу категорій трафіку в наборі даних UNSW-NB15. Вісь X відображає типи атак і нормальної активності, а вісь Y – кількість відповідних зразків.

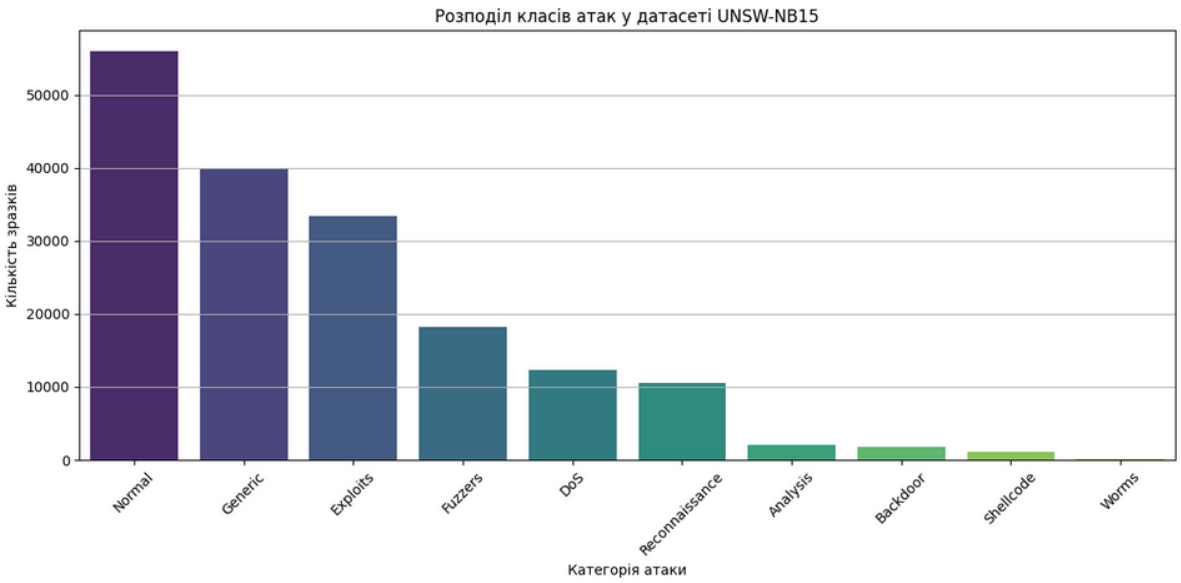

Рисунок 3.2 – Розподіл класів атак у датасеті UNSW-NB15

```
Logistic Regression Accuracy: 0.9437
[[16065   742]
 [ 1260 17480]]
              precision    recall  f1-score   support

           0       0.93      0.96      0.94     16807
           1       0.96      0.93      0.95     18740

    accuracy                           0.94     35547
   macro avg       0.94      0.94      0.94     35547
weighted avg       0.94      0.94      0.94     35547


Decision Tree Accuracy: 1.0000
[[16807     0]
 [    0 18740]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     16807
           1       1.00      1.00      1.00     18740

    accuracy                           1.00     35547
   macro avg       1.00      1.00      1.00     35547
weighted avg       1.00      1.00      1.00     35547


Random Forest Accuracy: 1.0000
[[16807     0]
 [    0 18740]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     16807
           1       1.00      1.00      1.00     18740

    accuracy                           1.00     35547
   macro avg       1.00      1.00      1.00     35547
weighted avg       1.00      1.00      1.00     35547


XGBoost Accuracy: 1.0000
[[16807     0]
 [    0 18740]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     16807
           1       1.00      1.00      1.00     18740

    accuracy                           1.00     35547
   macro avg       1.00      1.00      1.00     35547
weighted avg       1.00      1.00      1.00     35547
```

Рисунок 3.3 – Порівняння точності 4х моделей класифікації

Діаграма демонструє значну диспропорцію в представленості класів: найбільше спостерігається нормального трафіку, а також атак типу Generic,

Exploits і Fuzzers. Значно менше зразків мають категорії Shellcode, Backdoor, Worms і Analysis, що вказує на проблему незбалансованості даних, яка може негативно вплинути на якість класифікації без застосування методів балансування, таких як ADASYN.

На рисунку 3.3 представлено порівняння результатів класифікації чотирма моделями машинного навчання: Logistic Regression, Decision Tree, Random Forest і XGBoost. Основними метриками оцінки виступають точність, повнота, точність передбачень та F1-міра для кожного класу (0 – нормальний трафік, 1 – атака).

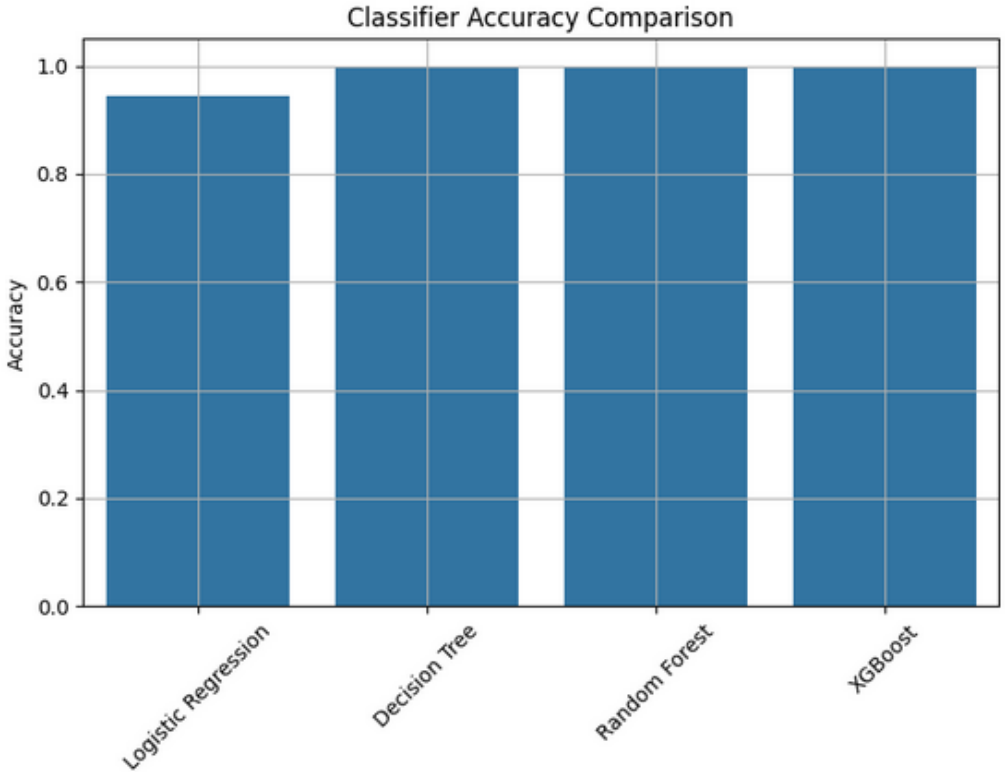

Рисунок 3.4 – Порівняння точності 4х моделей класифікації

Модель Logistic Regression досягає точності 94.37%, однак має відносно вищий рівень помилок другого роду, що свідчить про обмеження цієї лінійної моделі в умовах складної багатовимірної структури даних. Натомість усі три інші моделі – Decision Tree, Random Forest та XGBoost – демонструють ідеальні показники (100%) за всіма метриками, що вказує на

повну відповідність передбачень істинним міткам у тестовій вибірці. Така висока точність може свідчити як про сильну ефективність моделей, так і про потенційний ризик перенавчання, якщо моделі недостатньо протестовані на зовнішніх чи реалістично незбалансованих даних.

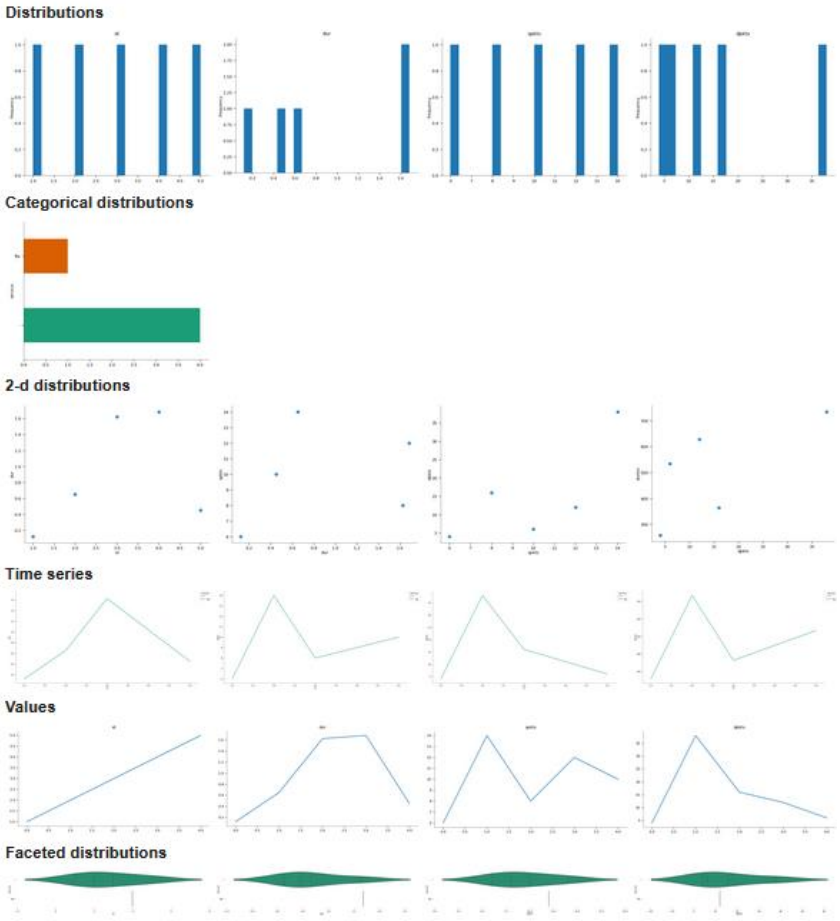

Рисунок 3.6 – Узагальнений візуальний аналіз різних типів розподілів і динаміки ознак у наборі даних UNSW-NB15

На рисунку 3.6 представлено узагальнений візуальний аналіз різних типів розподілів і динаміки ознак у наборі даних UNSW-NB15. Верхній блок містить гістограми числових змінних, що відображають частотний розподіл для різних параметрів, таких як тривалість з’єднань, кількість пакетів, кількість байтів тощо. Далі йдуть графіки категоріальних розподілів, які ілюструють кількість зразків для окремих категорій, наприклад, для протоколів чи станів з’єднання.

У секції 2-d distributions подано точкові діаграми, що дозволяють візуалізувати зв’язки між парами змінних, тоді як time series і values відображають зміну обраних параметрів у часовому порядку або в межах вибірки. Завершальний блок містить розподіли з фацетами, які дозволяють порівняти поведінку окремих змінних за різних умов. Такий набір графіків дає змогу швидко оцінити внутрішню структуру даних, наявність трендів, варіацій та потенційних аномалій, що є критично важливим перед застосуванням алгоритмів класифікації.

На рисунку 3.5 зображено порівняння точності чотирьох моделей класифікації: Logistic Regression, Decision Tree, Random Forest і XGBoost. Візуалізація демонструє, що всі три дерева-орієнтовані алгоритми показують майже однакову і дуже високу точність, близьку до 100%, тоді як логістична регресія поступається іншим підходам. Такий результат свідчить про те, що нелінійні моделі краще адаптовані до складної структури мережевого трафіку у задачі виявлення атак.

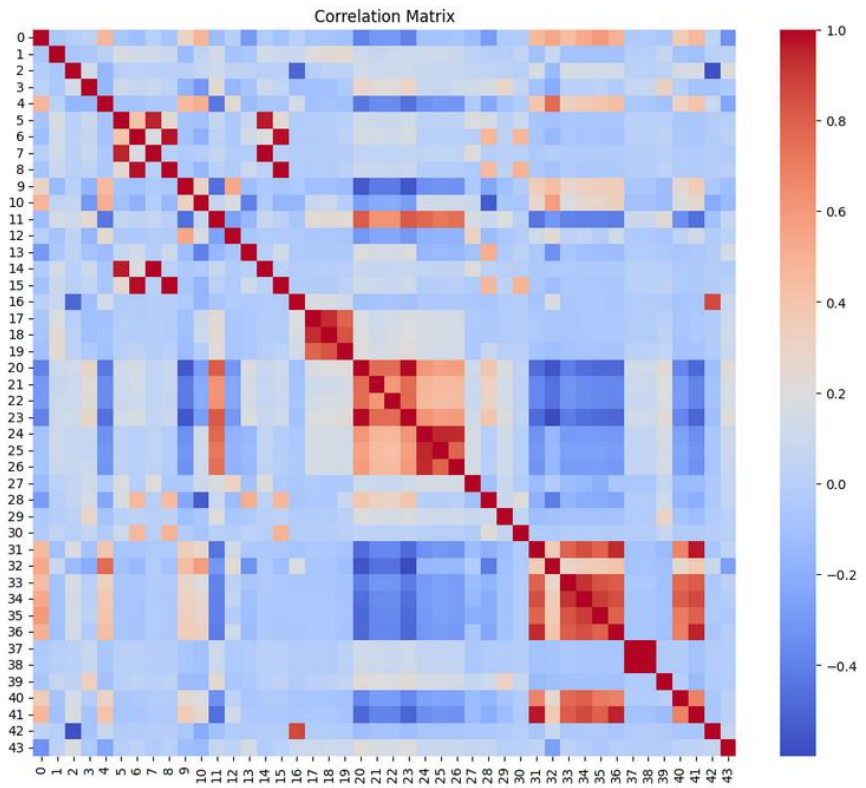

Рисунок 3.7 – Кореляційна матриця

На рисунку 3.7 наведено кореляційну матрицю, яка відображає ступінь взаємозв’язку між числовими ознаками у датасеті UNSW-NB15. Теплова карта побудована за шкалою від -1 до 1: червоні квадрати відповідають високій позитивній кореляції, сині – негативній, а світліші тони вказують на слабкий або відсутній зв’язок. Видимі скупчення червоних блоків уздовж діагоналі свідчать про наявність груп ознак, які мають сильний лінійний взаємозв’язок між собою.

Такі візуалізації допомагають виявити ознаки, що дублюють одна одну, і які можуть бути видалені або зведені до меншої кількості компонентів за допомогою методів зниження розмірності. Наявність значних кореляцій також дозволяє оптимізувати вибір ознак для моделей машинного навчання та зменшити ризик мультиколінеарності, що особливо важливо при використанні алгоритмів, чутливих до надмірності даних.

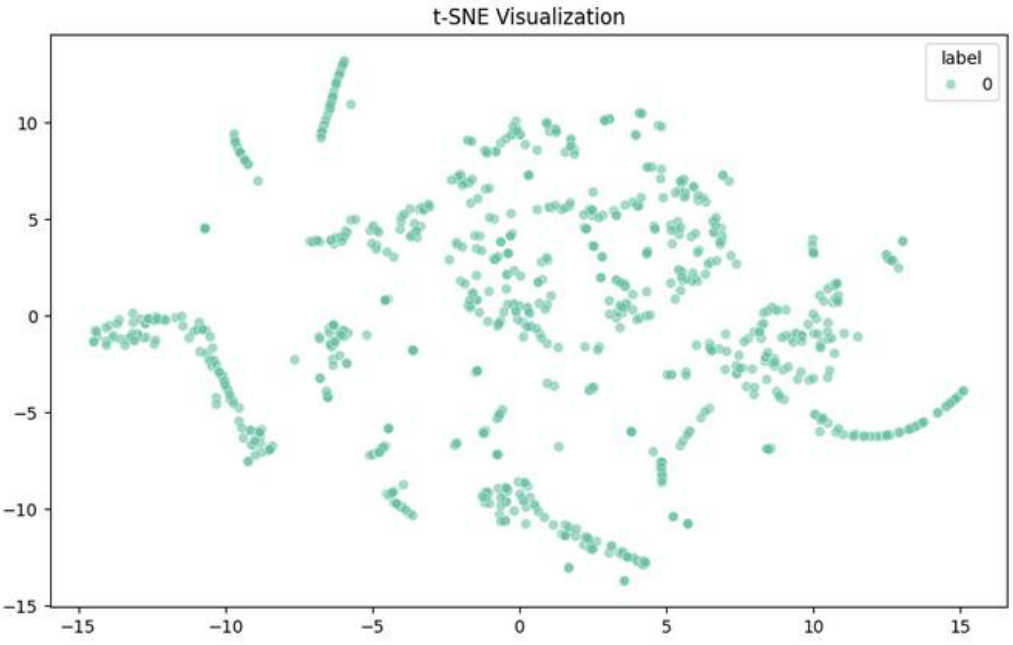

Рисунок 3.7 – Двовимірна проєкція

На рисунку 3.7 представлено двовимірну проєкцію простору ознак мережевого трафіку, отриману за допомогою методу нелінійного зниження розмірності t-SNE. Візуалізація дає змогу оцінити структуру та внутрішню

організацію даних класу з міткою 0, що відповідає нормальному (безпечному) трафіку. Незважаючи на відсутність чітких кластерів, можна спостерігати згущення точок у певних зонах, що може свідчити про подібність у поведінці або параметрах мережевих з'єднань. Метод t-SNE дозволяє краще інтерпретувати багатовимірні дані, однак для виявлення атак критичною є наявність також зразків шкідливого трафіку, оскільки лише на основі одного класу важко робити висновки щодо ефективності відокремлення.

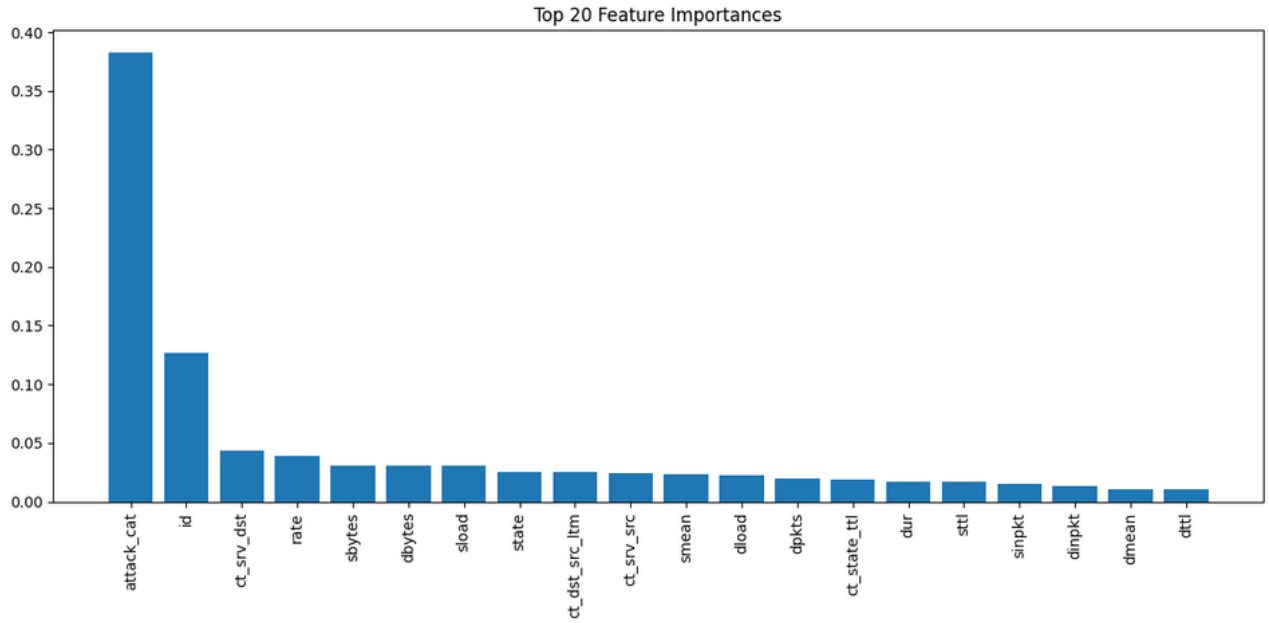

Рисунок 3.8 – Графік важливості ознак

На рисунку 3.8 представлено графік важливості ознак (Feature Importances) для моделі машинного навчання, зокрема для алгоритму Random Forest або XGBoost. Візуалізація демонструє двадцять найбільш значущих атрибутів, які найбільше впливають на рішення моделі при класифікації мережевого трафіку.

Найбільшою важливістю володіє ознака attack_cat, що свідчить про її тісний зв'язок із цільовою змінною або її потенційну наявність у навчальному наборі як дубльовану мітку, яку варто перевірити на предмет витоку інформації. Інші суттєві ознаки включають id, ct_srv_dst, rate, sbytes,

dbytes та state, що є критичними для аналізу мережевого трафіку, оскільки вони описують параметри з’єднань, обсяг переданих даних і стан сесій.

Графік дозволяє зробити висновки щодо доцільності зменшення розмірності або відбору найбільш інформативних ознак з метою покращення швидкодії та узагальненості моделі, особливо при подальшому використанні більш обчислювально затратних підходів, як-от нейронні мережі або бустинг з гіперпараметричною оптимізацією.

ВИСНОВКИ

У результаті проведеного дослідження було розроблено, обґрунтовано та програмно реалізовано метод діагностування несправностей у розподіленій системі моніторингу довкілля на основі технологій Інтернету речей з використанням інструментів машинного навчання. Запропонована архітектура передбачає організацію потокової обробки даних з великої кількості сенсорних пристроїв, що працюють у режимі реального часу, з подальшою реконструкцією нормальної поведінки системи за допомогою нейронної мережі типу автоенкодер. Це дозволяє виявляти відхилення, які потенційно сигналізують про вихід з ладу обладнання або порушення в роботі компонентів системи.

У дослідженні було реалізовано повноцінний цикл аналізу: від генерації імітованих даних із вбудованими аномаліями до навчання моделі на нормальних вибірках та ідентифікації відхилень на основі реконструктивної похибки. Побудовані графіки підтвердили високу ефективність запропонованого підходу – всі ключові аномалії були точно виявлені навіть на складних ділянках сигналу. Метод продемонстрував здатність адаптуватися до різних типів порушень, таких як імпульсні збурення, плавні зміщення або комбінації відхилень.

Особливу увагу приділено структурній моделі обробки діагностичної інформації, у межах якої були розглянуті способи виявлення аномалій на основі класифікаційних, кластеризаційних, статистичних та гібридних методів. У якості найбільш придатного підходу для цільового завдання було обґрунтовано застосування автоенкодера з можливістю реконструкції часових рядів і виявлення нетипових сегментів.

Важливим елементом роботи стало також формування прогнозних моделей для предиктивного обслуговування, що базуються на накопичених даних та історії змін у поведінці сенсорів. Це дозволяє не лише фіксувати

факт наявності несправностей, а й формувати інтервали ймовірного відмовлення з метою завчасного втручання. Такий підхід значно підвищує надійність, безперервність і адаптивність функціонування розподіленої IoT-системи моніторингу.

Окремо було проаналізовано актуальні наукові публікації, що підтвердили міждисциплінарний характер тематики та потребу в об’єднанні знань з інформатики, штучного інтелекту, телекомунікацій та інформаційної безпеки. Зроблено висновок, що перспективи розвитку таких систем мають прямий зв’язок із вдосконаленням механізмів захисту даних, підвищенням прозорості прийняття рішень у нейромережах та етичним впровадженням аналітики в чутливих середовищах.

Загалом, результати роботи засвідчили доцільність використання інтелектуальних методів для моніторингу стану сенсорних пристроїв у великих розподілених IoT-мережах. Запропоноване рішення може стати основою для побудови систем раннього попередження про несправності, систем екологічного контролю, а також платформ для цифрового управління розумними середовищами. За результатами роботи опубліковано статтю в фаховому виданні [10].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Eric Renault, Selma Boumerdassi, Paul Mühlethaler. Machine Learning for Networking: Third International Conference, MLN 2020, Paris, France, November 24–26, 2020, Revised Selected Papers. Springer, 2021, сторінки: 324. DOI: 10.1007/978-3-030-70866-5.

2. B. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti. A Survey of Machine Learning Techniques for Network Traffic Classification. IEEE Communications Surveys & Tutorials, 2016, сторінки: 56-76.

3. Mohammed Hussein Thwaini. Anomaly Detection in Network Traffic using Machine Learning for Early Threat Detection//Data & Metadata 2022, p. 1-13. https://doi.org/10.56294/dm202272

4. Aburomman AA, Reaz MBI. A survey of intrusion detection systems based on ensemble and hybrid classifiers. Computers & Security. 2017;65:135-152.

5. Agrawal S, Agrawal J. Survey on anomaly detection using data mining techniques. Procedia Computer Science. 2015;60:708-713.

6. Ahmad S, Lavin A, Purdy S, Agha Z. Unsupervised real-time anomaly detection for streaming data. Neurocomputing. 2017;262:134-147.

7. Aissa NB, Guerroumi M. Semi-supervised statistical approach for network anomaly detection. Procedia Computer Science. 2016;83:1090-1095.

8. Bhati BS, Rai CS, Balamurugan B, Al-Turjman F. An intrusion detection scheme based on the ensemble of discriminant classifiers. Computers & Electrical Engineering. 2020;86:106742.

9. Aung YY, Min MM. An analysis of K-means algorithm-based network intrusion detection system. Advances in Science, Technology and Engineering Systems Journal. 2018;3(1):496-501.

10. Do K., Klymova I., Naumova E., Herevych M., Yankovskyi O. Data processing and analysis methods in IOT using machine learning. Системи управління, навігації та зв’язку, вип.2. Полтава, 2025. С. 119-124.