

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

До захисту допущено:

В.о.завідувача кафедри

_____ Сергій КРАВЧУК

«__» _____ 2026 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія та програмування
інфокомунікацій»**

спеціальності 172 «Телекомунікації та радіотехніка»

**на тему: «Виявлення аномалій і мережевих атак у операторському
трафіку методами неконтрольованого навчання»**

Виконав:

студент IV курсу, групи ТЗ-22

Ходанич Анатолій Олександрович _____

Керівник:

Доцент кафедри ТК НН ІТС, с.н.с., к.т.н.

Міночкін Дмитро Анатолійович _____

Рецензент:

Доцент кафедри ІТТ НН ІТС, к.т.н., доцент

Суліма Світлана Валеріївна _____

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

Студент _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри

_____ Сергій КРАВЧУК

«__» _____ 2026 р.

ЗАВДАННЯ

на дипломну роботу студенту

Ходаничу Анатолію Олександровичу

1. Тема роботи «Виявлення аномалій і мережевих атак у операторському трафіку методами неконтрольованого навчання», керівник роботи Міночкін Дмитро Анатолійович, с.н.с., к.т.н., затверджені наказом по університету від «26» травня 2026 р. № 1912-с.
2. Термін подання студентом роботи 10 червня 2026 р.
3. Вихідні дані до роботи: набори даних мережевого трафіку KDD Cup 99 та CICIDS2017; протоколи збору мережевих потоків NetFlow v9 та IPFIX; методи неконтрольованого машинного навчання (k-means, DBSCAN, Isolation Forest, Local Outlier Factor, LSTM-автоенкодер); бібліотеки Python: scikit-learn, TensorFlow/Keras, pandas, numpy; програмне забезпечення для збору та обробки трафіку: nfdump, Apache Kafka, Apache Spark Streaming.
4. Зміст роботи: Провести аналіз сучасного стану проблеми виявлення аномалій і мережевих атак в операторському трафіку та огляд існуючих підходів; Дослідити та систематизувати методи неконтрольованого навчання (k-means, DBSCAN, Isolation Forest, LOF, LSTM-автоенкодер) для виявлення аномалій у мережевому трафіку; Сформувати вектор ознак мережевих потоків та розробити pipeline попередньої обробки даних; Реалізувати та порівняти алгоритми виявлення аномалій на наборах даних KDD Cup 99 та CICIDS2017.; Розробити архітектуру системи виявлення аномалій для операторських мереж

на базі мікросервісного підходу; Сформулювати практичні рекомендації щодо вибору та впровадження системи виявлення аномалій залежно від масштабу оператора зв'язку.

5. Орієнтовний перелік ілюстративного матеріалу:

Слайд 1 — Титульний;

Слайд 2 — Актуальність та мета;

Слайд 3 — Класифікація загроз;

Слайд 4 — Методи неконтрольованого навчання;

Слайд 5 — Вектор ознак;

Слайд 6 — Архітектура системи;

Слайд 7 — Результати порівняльного аналізу;

Слайд 8 — Ефективність за типами атак;

Слайд 9 — Адаптивний поріг;

Слайд 10 — Практичні рекомендації;

Слайд 11 — Висновки;

6. Дата видачі завдання: 20.04.2026 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Проаналізувати існуючі дослідження та сучасні підходи до виявлення мережових атак і аномалій в операторському трафіку, визначити переваги та обмеження традиційних IDS-систем на основі сигнатур	20.04.2026 - 26.04.2026	Виконано
2	Дослідити структурні особливості операторського мережевого трафіку: параметри з'єднань (IP-адреси, порти, протоколи, тривалість сесій, обсяги переданих даних) та закономірності нормальної поведінки мережі	27.04.2026 – 03.05.2026	Виконано
3	Вивчити методи неконтрольованого навчання для виявлення аномалій — алгоритми кластеризації (K-means, DBSCAN) та методи виявлення викидів (Isolation Forest, Local Outlier Factor) — та обґрунтувати вибір оптимального підходу для аналізу операторського трафіку	27.04.2026 – 03.05.2026	Виконано
4	Опрацювати принципи підготовки та попередньої обробки мережових даних: методи очищення, нормалізації та відбору ознак, що застосовуються у задачах машинного навчання для аналізу трафіку	04.05.2026 – 10.05.2025	Виконано
5	Оформити пояснювальну записку до дипломної роботи, Підготовка звіту з практики, оформлення щоденника.	11.05.2026 – 17.05.2026	Виконано

Студент

Анатолій ХОДАНИЧ

Керівник

Дмитро МІНОЧКІН

РЕФЕРАТ

Обсяг пояснювальної записки: 60 сторінка, 4 рисунки, 7 таблиць, 45 бібліографічних найменувань.

Мета роботи – розробка та порівняльне дослідження системи виявлення аномалій і мережевих атак в операторському трафіку з використанням методів неконтрольованого машинного навчання.

Об'єкт дослідження – мережевий трафік телекомунікаційних операторів зв'язку.

Предмет дослідження – методи неконтрольованого навчання для виявлення аномалій та мережевих атак у мережевому трафіку.

Методи дослідження: методи неконтрольованого машинного навчання — k-means, DBSCAN, Isolation Forest, Local Outlier Factor, автоенкодер на базі LSTM; статистичний аналіз мережевого трафіку; моделювання та тестування на наборах даних KDD Cup 99 та CICIDS2017.

Отримані результати. Розроблено архітектуру п'ятирівневої системи виявлення аномалій на базі мікросервісного підходу. Проведено порівняльний аналіз семи методів неконтрольованого навчання. Встановлено, що ансамблева модель (Isolation Forest + LSTM-автоенкодер) забезпечує F1-score = 0,93 при рівні хибно-позитивних спрацьовувань 3,1%. Розроблено адаптивний механізм встановлення порогу та практичні рекомендації щодо впровадження системи в мережах операторів зв'язку різного масштабу.

Ключові слова: АНОМАЛІЯ, МЕРЕЖЕВА АТАКА, НЕКОНТРОЛЬОВАНЕ НАВЧАННЯ, ISOLATION FOREST, АВТОЕНКОДЕР, LSTM, DBSCAN, LOF, NETFLOW, ОПЕРАТОРСЬКИЙ ТРАФІК, IDS, КЛАСТЕРИЗАЦІЯ.

ABSTRACT

Volume of explanatory note: 60 pages, 4 figures, 7 tables, 45 bibliographic references.

Purpose – development and comparative research of an anomaly and network attack detection system for telecommunication operator traffic using unsupervised machine learning methods.

Object of research – network traffic of telecommunication operators.

Subject of research – unsupervised learning methods for anomaly and network attack detection in network traffic.

Research methods: unsupervised machine learning methods (k-means, DBSCAN, Isolation Forest, Local Outlier Factor, LSTM-based autoencoders), statistical network traffic analysis, simulation and testing on KDD Cup 99 and CICIDS2017 datasets.

Results obtained. A five-level anomaly detection system architecture based on a microservice approach was developed. A comparative analysis of seven unsupervised learning methods was conducted. An ensemble model (Isolation Forest + LSTM autoencoder) achieves F1-score = 0.93 with a false positive rate of 3.1%. An adaptive threshold mechanism and practical recommendations for deployment in telecommunication operator networks of various scales were developed.

Keywords: ANOMALY, NETWORK ATTACK, UNSUPERVISED LEARNING, ISOLATION FOREST, AUTOENCODER, LSTM, DBSCAN, LOF, NETFLOW, OPERATOR TRAFFIC, IDS, CLUSTERING.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1 АНАЛІЗ СТАНУ ПРОБЛЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ У МЕРЕЖЕВОМУ ТРАФІКУ.....	12
1.1. Характеристика мережевого трафіку операторів зв'язку та виклики кібербезпеки	12
1.2. Класифікація мережевих аномалій.....	15
1.3. Класифікація мережевих атак в операторських мережах	18
1.3.1. Атаки відмови в обслуговуванні (DoS/DDoS).....	18
1.3.2. Мережева розвідка та сканування	20
1.3.3. Ботнетна активність та командно-контрольний трафік	20
1.3.4. DNS-тунелювання	21
1.3.5. Ексфільтрація даних.....	21
1.3.6. Горизонтальне переміщення у мережі	22
1.4. Порівняльний аналіз існуючих підходів до виявлення загроз	23
1.4.1. Сигнатурні методи виявлення	23
1.4.2. Аномалійні методи	24
1.4.3. Гібридні підходи.....	25
1.5. Огляд наукових досліджень у сфері виявлення аномалій.....	26
1.6. Набори даних для дослідження та оцінювання.....	29
Висновок.....	31
РОЗДІЛ 2 МЕТОДИ НЕКОНТРОЛЬОВАНОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ	34
2.1. Загальні принципи та постановка задачі.....	34
2.2. Статистичні методи.....	34
2.2.1. Метод статистичного профілювання.....	34
2.2.2. Local Outlier Factor.....	35
2.2.3. One-Class SVM.....	35
2.3. Методи кластеризації.....	35
2.3.1. k-means.....	35
2.3.2. DBSCAN	36
2.4. Isolation Forest.....	36
2.5. Автоенкодери.....	37

2.5.1. Класичний автоенкодер	37
2.5.2. LSTM-автоенкодер	38
2.6. Порівняльна характеристика методів.....	38
Висновок.....	39
РОЗДІЛ 3 ФОРМУВАННЯ ВЕКТОРА ОЗНАК ТА ПІДГОТОВКА ДАНИХ	40
3.1. Джерела та методи збору мережевого трафіку	40
3.2. Формування вектора ознак.....	41
3.3. Pipeline попередньої обробки даних.....	42
3.4. Характеристика наборів даних та розподіл вибірок.....	43
Висновок.....	44
РОЗДІЛ 4 РОЗРОБКА СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	45
4.1. Архітектура системи виявлення аномалій.....	45
4.2. Реалізація та налаштування моделей	47
4.2.1. Isolation Forest: параметри та продуктивність	47
4.2.2. LSTM-автоенкодер: архітектура та навчання.....	47
4.2.3. Ансамблева модель	47
4.3. Результати порівняльного аналізу	47
4.4. Ефективність за типами атак.....	49
4.5. Адаптивний поріг та аналіз хибних спрацьовувань	51
4.6. Продуктивність та вимоги до ресурсів	52
4.7. Практичні рекомендації щодо впровадження	52
Висновок.....	53
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56

ПЕРЕЛІК СКОРОЧЕНЬ

AUC ROC	Area Under ROC Curve – площа під кривою ROC
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DDoS	Distributed Denial of Service – розподілена відмова в обслуговуванні
DNS	Domain Name System – система доменних імен
DPI	Deep Packet Inspection – глибока інспекція пакетів
FPR	False Positive Rate – рівень хибно-позитивних спрацьовувань
IDS	Intrusion Detection System – система виявлення вторгнень
IPFIX	IP Flow Information Export – протокол експорту інформації про потоки IP
LOF	Local Outlier Factor – локальний коефіцієнт викиду
LSTM	Long Short-Term Memory – довга короткострокова пам'ять
ML	Machine Learning – машинне навчання
NetFlow	Мережевий протокол збору статистики трафіку (Cisco)
PCA	Principal Component Analysis – метод головних компонент
SIEM	Security Information and Event Management
SOC	Security Operations Center – центр операцій з безпеки
SVM	Support Vector Machine – метод опорних векторів
TCP	Transmission Control Protocol – протокол управління передачею
TLS	Transport Layer Security – протокол захисту транспортного рівня
TTL	Time To Live – час життя пакету
UDP	User Datagram Protocol – протокол дейтаграм користувача
CBA	Система виявлення аномалій

ВСТУП

Телекомунікаційні мережі операторів зв'язку є критичною інфраструктурою сучасного суспільства. Вони забезпечують передачу голосу, відео, даних та мультимедійного контенту для мільйонів абонентів — фізичних осіб, підприємств та державних установ. Щодня через ці мережі проходять петабайти трафіку. Разом із зростанням обсягів і різноманітності трафіку зростає і кількість кіберзагроз, спрямованих проти операторської інфраструктури. Традиційні засоби захисту — міжмережеві екрани та сигнатурні системи виявлення вторгнень — орієнтовані на пошук заздалегідь відомих патернів шкідливої поведінки і неспроможні ефективно протидіяти новим, раніше невідомим типам атак. [1]

Особливу загрозу становлять масштабні DDoS-атаки, що можуть вивести з ладу цілі вузли мережі, а також приховані атаки — ботнетна активність, DNS-тунелювання, повільне сканування портів, ексфільтрація даних, — які маскуються під легітимний трафік і тривалий час можуть залишатися непоміченими. За даними ENISA, у 2023–2024 роках кількість DDoS-інцидентів у телекомунікаційній галузі зросла на 34%, а середня тривалість атак збільшилась до 5,5 годин. [2]

В умовах зростання частки шифрованого трафіку — понад 85% за даними Sandvine за 2024 рік — традиційний аналіз вмісту пакетів стає практично неможливим. Перспективним напрямком є аналіз мережевих потоків та застосування методів машинного навчання, зокрема неконтрольованого навчання. Перевагою неконтрольованих методів є можливість навчання виключно на нормальному трафіку без необхідності розмічати навчальні вибірки вручну. [3]

Актуальність теми обумовлена тим, що вітчизняні оператори зв'язку стикаються із зростаючою кількістю кіберінцидентів, а наявні комерційні

рішення є дорогими і не завжди адаптованими до специфіки конкретних мереж. Розробка підходів на базі відкритих алгоритмів неконтрольованого навчання відкриває можливості для ефективного і економічно доцільного захисту мереж.

Метою дипломної роботи є дослідження та порівняльний аналіз методів неконтрольованого навчання для виявлення аномалій і мережевих атак в операторському трафіку, а також розробка архітектури системи виявлення аномалій та практичних рекомендацій щодо її впровадження.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз сучасного стану проблеми виявлення аномалій у мережевому трафіку та огляд існуючих підходів;
- дослідити та систематизувати методи неконтрольованого навчання, придатні для аналізу трафіку операторів зв'язку;
- сформувати набір характеристик мережевих потоків для навчання та тестування моделей;
- реалізувати та порівняти алгоритми k-means, DBSCAN, Isolation Forest, LOF та LSTM-автоенкодер;
- розробити архітектуру системи виявлення аномалій для операторських мереж;
- оцінити ефективність запропонованих підходів на реальних наборах даних і сформулювати практичні рекомендації.

Об'єктом дослідження є мережевий трафік телекомунікаційних операторів зв'язку.

Предметом дослідження є методи неконтрольованого навчання для виявлення аномалій і мережевих атак у мережевому трафіку.

Практичне значення роботи полягає у розробці рекомендацій щодо вибору та налаштування алгоритмів виявлення аномалій для впровадження в операторських мережах зв'язку різного масштабу.

РОЗДІЛ 1

АНАЛІЗ СТАНУ ПРОБЛЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ У МЕРЕЖЕВОМУ ТРАФІКУ

1.1. Характеристика мережевого трафіку операторів зв'язку та виклики кібербезпеки

Телекомунікаційні мережі операторів зв'язку є фундаментальною інфраструктурою сучасного цифрового суспільства. Вони забезпечують передачу голосу, відео, даних та мультимедійного контенту для мільйонів абонентів — фізичних осіб, підприємств та державних установ. Масштаб такої інфраструктури вражає: великі міжнародні оператори, такі як Deutsche Telekom, Verizon або AT&T, щодня обробляють трафік обсягом у десятки екзабайт, а їхні магістральні вузли функціонують на швидкостях від 100 Гбіт/с до кількох Тбіт/с. За прогнозами Cisco, загальний обсяг глобального IP-трафіку до 2025 року перевищить 400 екзабайт на місяць, що у чотири рази більше порівняно з показниками 2020 року.

Роль телекомунікаційних мереж у суспільстві важко переоцінити. Вони забезпечують функціонування систем охорони здоров'я (телемедицина, системи екстреного виклику 112), фінансового сектору (онлайн-банкінг, платіжні системи, фондові біржі), державного управління та критичної інфраструктури (системи диспетчеризації електромереж, водопостачання, транспортні системи). Будь-яке суттєве порушення роботи оператора зв'язку миттєво позначається на суспільстві в цілому. Саме тому забезпечення кібербезпеки операторських мереж розглядається регуляторами як стратегічне завдання. У Директиві ЄС NIS2 (2022/2555) та Законі України «Про основні засади забезпечення кібербезпеки України» оператори зв'язку прямо визначені як суб'єкти критичної інфраструктури з підвищеними вимогами до захисту. [8]

Сучасні кіберзагрози, спрямовані проти операторів зв'язку, є надзвичайно різноманітними та постійно еволюціонують. За даними щорічного звіту ENISA Threat Landscape 2024, телекомунікаційна галузь є

однією з найбільш атакованих: кількість зареєстрованих кіберінцидентів зросла на 34% у 2023–2024 роках порівняно з попереднім роком. При цьому середня тривалість DDoS-атак збільшилась до 5,5 годин, а їх інтенсивність регулярно досягає терабітних значень. Наслідки успішних атак для операторів включають: пряму фінансову шкоду від простою сервісів, витоки персональних даних абонентів, шкоду репутації та регуляторні штрафи відповідно до GDPR та вітчизняного законодавства про захист персональних даних. [2]

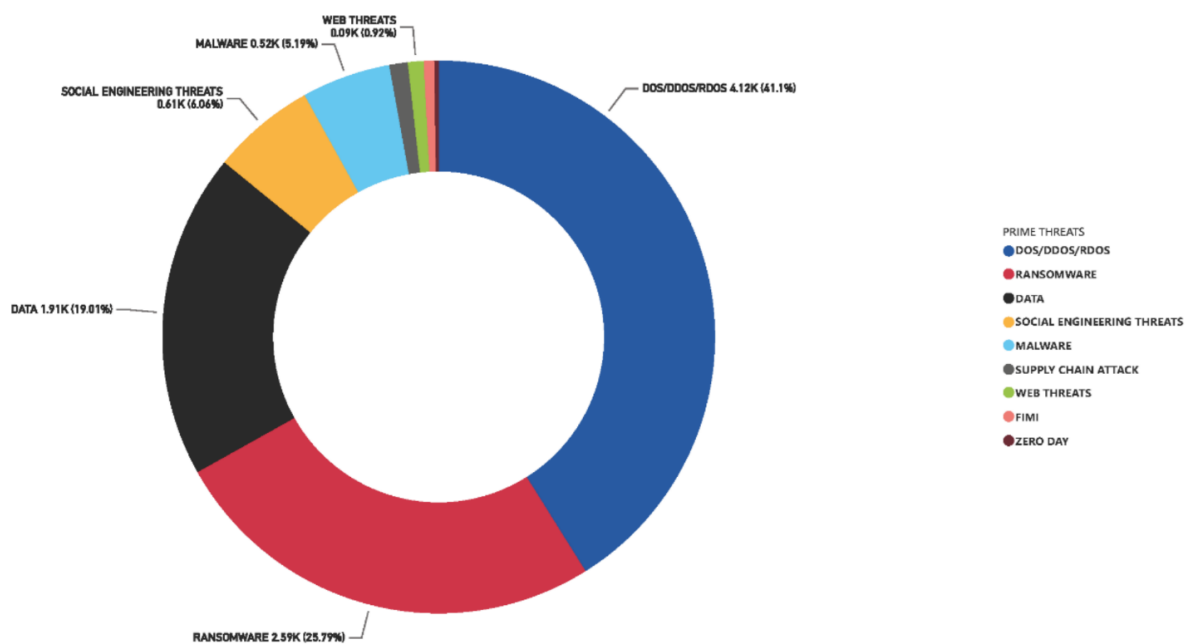


Рис. 1.1. Розподіл кіберінцидентів за типами загроз (липень 2023 – червень 2024)

Джерело: ENISA Threat Landscape 2024 [2]

Технічний захист операторських мереж ускладнюється кількома фундаментальними чинниками. По-перше, масштаб трафіку робить повний пакетний аналіз практично нереалістичним. Метод глибокої інспекції пакетів (DPI) вимагає розбору кожного пакету до рівня прикладного протоколу. Для магістральних каналів оператора з пропускнуою здатністю 100 Гбіт/с це означає обробку до 14,88 мільйонів 64-байтних пакетів на секунду —

завдання, що вимагає спеціалізованих апаратних рішень ASIC або FPGA вартістю сотні тисяч доларів за вузол. [3]

По-друге, стрімко зростає частка шифрованого трафіку. За даними глобального звіту Sandvine Global Internet Phenomena Report 2024, понад 85% усього інтернет-трафіку шифрується за допомогою протоколу TLS 1.2 або TLS 1.3. Це унеможлиблює аналіз вмісту пакетів (payload analysis) без розшифровки трафіку — що є юридично обмеженим (порушення таємниці переписки та персональних даних) і технічно складним. У результаті традиційний DPI втрачає ефективність для переважної більшості сучасного трафіку. [5]

По-третє, значно збільшується різноманітність та складність кіберзагроз. Якщо 10–15 років тому більшість атак були відносно простими і мали чітку сигнатуру, то сучасні атаки є багатостадійними, адаптивними та здатні маскуватись під легітимний трафік. Зловмисники цілеспрямовано вивчають системи захисту і модифікують свої тактики, щоб уникнути виявлення. Це робить підходи, що спираються виключно на відомі сигнатури, принципово недостатніми. [8]

Альтернативним підходом, що дозволяє вирішити описані проблеми, є аналіз мережеских потоків (flow-based analysis) у поєднанні з методами машинного навчання. Мережеский потік — це агрегована одиниця трафіку між двома кінцевими точками, що визначається «п'ятіркою»: IP-адреса джерела, IP-адреса призначення, порт джерела, порт призначення та протокол транспортного рівня. Замість аналізу вмісту кожного пакету оператор збирає статистику про кожне з'єднання: тривалість, обсяг переданих байт і пакетів, часові мітки початку та закінчення, TCP-прапорці, середній розмір пакету тощо. Протоколи NetFlow v9 та IPFIX стандартизували цей підхід і дозволяють

збирати таку статистику безпосередньо на маршрутизаторах без копіювання трафіку. [4]

Ключовою перевагою flow-based підходу є ефективна робота з шифрованим трафіком. Навіть якщо вміст пакетів недоступний, метадані потоків несуть достатньо інформації для виявлення переважної більшості типів мережових атак. Дослідження Sperotto et al. (2010) показали, що на основі виключно flow-рівневих ознак можна досягти точності виявлення DDoS-атак понад 95%, мережевого сканування — понад 90%, а ботнетної активності — понад 85%. Це підтверджується і більш сучасними дослідженнями: Fernandes et al. (2019) у мета-аналізі 112 публікацій встановили, що flow-based методи незмінно забезпечують кращі результати, ніж пакетно-орієнтовані для задач виявлення аномалій у великих операторських мережах. [3] [12]

1.2. Класифікація мережових аномалій

Коректна класифікація аномалій є фундаментальним теоретичним підґрунтям для побудови системи виявлення. Поняття «аномалія» у контексті мережевого трафіку вперше систематично дослідив Hawkins у 1980 році, визначивши її як «спостереження, що настільки відхиляється від інших, що виникають підозри щодо іншого механізму його генерації». У застосуванні до мережевого трафіку механізм генерації може бути як технічним (збій обладнання, помилка конфігурації), так і зловмисним (кібератака, несанкціоноване використання ресурсів). [6]

У класичній таксономії Chandola, Banerjee та Kumar (2009), що є найбільш цитованою у сфері виявлення аномалій, виділяють три фундаментальних типи аномалій. Ця класифікація має не лише теоретичне, але й суто практичне значення, оскільки різні типи аномалій вимагають принципово різних методів виявлення. [10]

Точкова аномалія (point anomaly) є найбільш поширеним та інтуїтивно зрозумілим типом. Вона являє собою окреме спостереження, що різко відрізняється від решти набору даних при розгляді у глобальному просторі ознак. У контексті мережевого трафіку класичним прикладом точкової аномалії є хост, що раптово починає генерувати трафік у тисячу разів більший за його звичайний профіль — що є типовою ознакою DDoS-атаки або скомпрометованого вузла, що використовується як бот для генерації атакного трафіку. Інший приклад — поява потоку з аномально великим розміром пакетів, що не відповідає жодному відомому протоколу і може свідчити про DNS-тунелювання. Точкові аномалії добре виявляються більшістю методів: статистичним профілюванням (правило 3σ), Isolation Forest, LOF, а також автоенкодерами. [6] [10]

Контекстуальна аномалія (contextual anomaly), яку також називають умовною або ситуаційною, є суттєво складнішим типом для виявлення. Спостереження вважається контекстуально аномальним, якщо воно є аномальним у певному контексті, але не за всіх умов. У мережевому трафіку контекст визначається насамперед часом: велика кількість вихідних з'єднань від корпоративного сервера є цілком нормальним явищем у робочий час (коли тисячі співробітників активно користуються інтернетом), але може свідчити про ексфільтрацію даних о 3:00 ночі у вихідний день. Контекст також може визначатися просторовими факторами: трафік між двома конкретними хостами є нормальним, якщо вони регулярно взаємодіють за роллю своїх сервісів, але аномальним, якщо раніше ніколи не мали з'єднань між собою. Виявлення контекстуальних аномалій вимагає включення контекстних ознак (година доби, день тижня, відхилення від базового профілю) у вектор даних для навчання. [6]

Колективна аномалія (collective anomaly) є найбільш складним для виявлення типом. Вона виникає тоді, коли колекція пов'язаних спостережень

є аномальною відносно всього набору даних, хоча кожне окреме спостереження при розгляді у ізоляції може не бути аномальним. У мережевому трафіку класичним прикладом колективної аномалії є повільне сканування портів: один SYN-пакет до довільного порту — абсолютно звичайна подія. Проте 50 000 таких пакетів від одного хоста до 50 000 різних портів, рівномірно розподілених протягом 8 годин — це очевидна розвідувальна операція зловмисника, що готується до атаки. Жоден з пакетів окремо не є аномальним, але їх сукупність формує однозначно підозрілий патерн. Для виявлення колективних аномалій необхідний аналіз послідовностей спостережень у часі, що найкраще реалізується за допомогою рекурентних нейронних мереж (LSTM-автоенкодерів) або методів, що агрегують статистику за часовими вікнами. [6] [10]

Таблиця 1.1 систематизує зв'язок між типами аномалій, відповідними типами атак та найбільш ефективними методами виявлення — ця відповідність є ключовою при проектуванні системи виявлення.

Таблиця 1.1

Типи аномалій та відповідні методи виявлення

Тип аномалії	Приклад атаки	Характерна ознака	Ефективний метод
Точкова	DDoS, ексфільтрація	Різкий стрибок total_bytes	Isolation Forest, 3σ
Контекстуальна	Нічна ексфільтрація	Нормальне значення у аномальний час	LOF + часові ознаки
Колективна	Повільне сканування, ботнет	Патерн у послідовності подій	LSTM-автоенкодер

1.3. Класифікація мережевих атак в операторських мережах

Розуміння природи та механізмів різних типів мережевих атак є необхідною умовою для проектування ефективної системи виявлення аномалій. Зловмисники постійно вдосконалюють свої методи, тому класифікація атак повинна ґрунтуватись не лише на поточному стані, але й враховувати тенденції розвитку кіберзагроз. У даній роботі розглядаються шість типів атак, що є найбільш актуальними для операторів зв'язку відповідно до звіту ENISA Threat Landscape 2024. [2]

1.3.1. Атаки відмови в обслуговуванні (DoS/DDoS)

Атаки типу «відмова в обслуговуванні» (Denial of Service, DoS) та їх розподілена версія (Distributed Denial of Service, DDoS) є найбільш поширеними та найбільш шкідливими для операторів зв'язку. Мета атаки — вичерпати ресурси цільової системи (обчислювальні потужності, пропускну здатність мережевого каналу, кількість одночасних з'єднань) шляхом генерування масштабного синтетичного трафіку, що унеможливорює обслуговування легітимних запитів. [2] [8]

Розподілені DDoS-атаки здійснюються одночасно з тисяч або мільйонів скомпрометованих вузлів (ботів), що входять до складу ботнету. Це суттєво ускладнює протидію, оскільки блокування окремих IP-адрес є малоефективним при сотнях тисяч джерел. Сучасні DDoS-атаки досягають інтенсивності у кілька Тбіт/с і здатні перевантажити навіть добре захищену інфраструктуру. Залежно від механізму атаки виділяють: об'ємні атаки (volumetric) — UDP/ICMP flood, DNS amplification, що вичерпують пропускну здатність каналу; протокольні атаки — SYN flood, Ping of Death, що вичерпують ресурси мережевого стеку; атаки на рівні застосунків — HTTP flood, Slowloris, що вичерпують ресурси серверних застосунків. [2] [8]

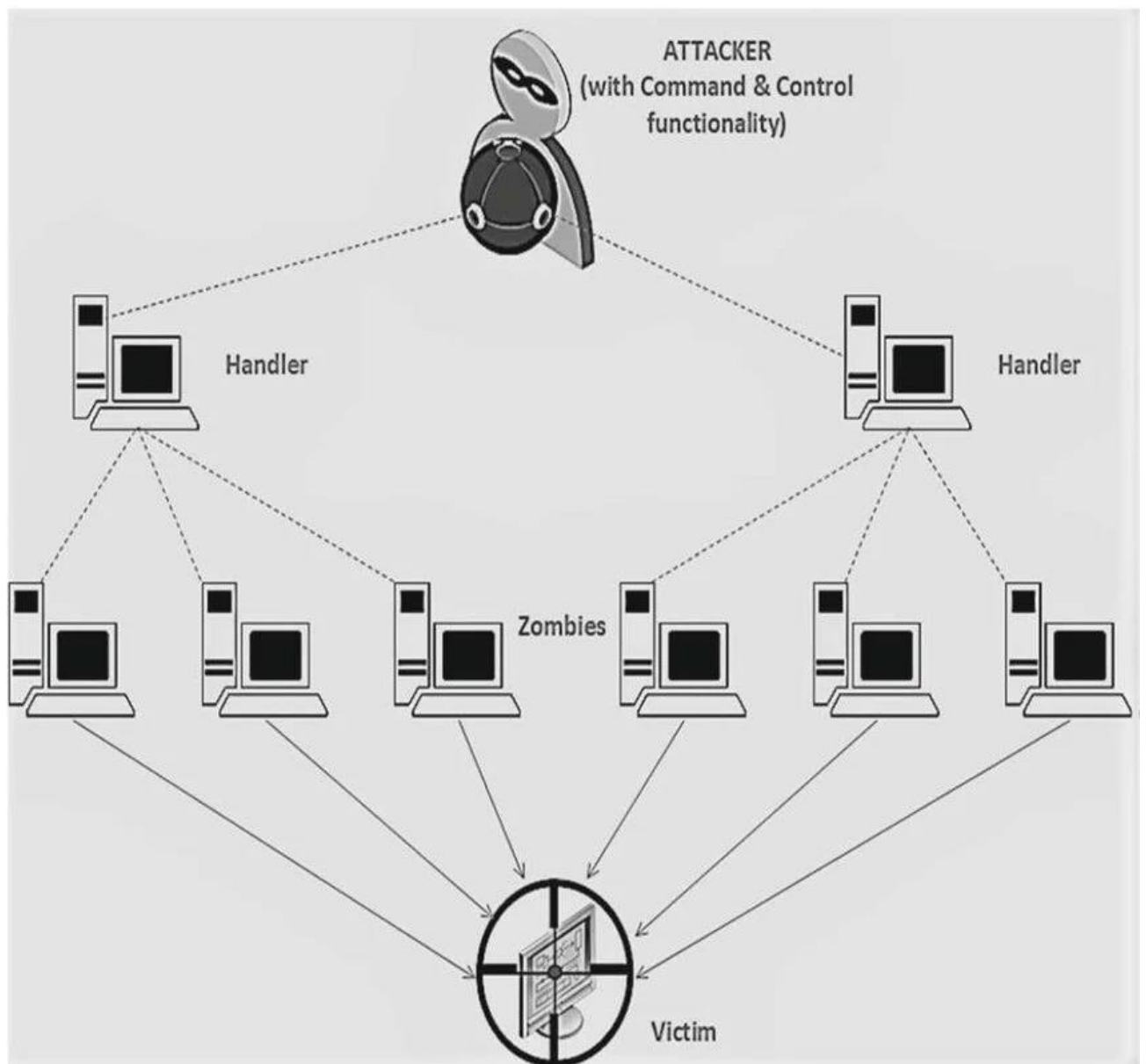


Рис. 1.2. Архітектура розподіленої DDoS-атаки

Джерело: [43]

З точки зору виявлення, DDoS-атаки є найпростішим типом для детектування: вони спричиняють різкі та значні зміни статистичних ознак трафіку — кількості пакетів, байт, нових з'єднань. Правило 3σ та Isolation Forest виявляють об'ємні DDoS із Recall понад 95%. Складнішими є «повільні» DDoS-атаки (Slowloris, Slow HTTP), що підтримують з'єднання відкритими якомога довше, не генеруючи значного обсягу трафіку — їх виявлення потребує аналізу тривалості з'єднань та кількості одночасно відкритих сесій.

1.3.2. Мережева розвідка та сканування

Мережева розвідка є підготовчим етапом більшості складних цілеспрямованих атак. Зловмисник систематично зондує цільову мережу для збору інформації про її структуру: активні хости, відкриті порти, запущені сервіси та їх версії, операційні системи, мережеву топологію. Інструменти на кшталт Nmap, Masscan або Shodan дозволяють автоматизувати цей процес і опрацьовувати мільйони адрес за годину. [2] [8]

Горизонтальне сканування (horizontal scan) спрямоване на виявлення певного відкритого порту (наприклад, 22/SSH або 3389/RDP) на максимальній кількості хостів. Вертикальне сканування (vertical scan) досліджує весь діапазон портів одного конкретного хоста. Сучасні зловмисники часто застосовують «повільне» сканування (slow scan), розподіляючи запити у часі так, щоб залишатись нижче порогу виявлення традиційних систем. Ключовою ознакою сканування є різке зростання кількості унікальних портів призначення (unique_dst_ports) або унікальних IP-адрес призначення в одному напрямку трафіку.

1.3.3. Ботнетна активність та командно-контрольний трафік

Ботнет — це мережа скомпрометованих вузлів (ботів), керована зловмисником через централізовану або розподілену командно-контрольну інфраструктуру (C&C, або C2). Боти використовуються для різних зловмисних цілей: проведення DDoS-атак, розсилки спаму, майнінгу криптовалют, крадіжки облікових даних, встановлення програм-вимагачів (ransomware). Для операторів зв'язку особливу небезпеку становить те, що боти можуть знаходитись безпосередньо серед їх абонентів. [2] [8]

Командно-контрольна комунікація між ботами та сервером керування є характерною ознакою ботнетної активності. Традиційні ботнети використовували централізовану архітектуру з фіксованими IP-адресами

C&C-серверів — такий трафік відносно легко виявляється. Сучасні ботнети (наприклад, Mirai, Emotet) застосовують peer-to-peer (P2P) архітектуру та алгоритми генерації доменів (DGA, Domain Generation Algorithm), що значно ускладнює їх виявлення та нейтралізацію. Характерними ознаками C&C-трафіку є: регулярні з'єднання з фіксованими зовнішніми адресами за чітким часовим розкладом (heartbeat), часто у нічний час; незвичні порти або протоколи; шифрування з нестандартними сертифікатами або без них.

1.3.4. DNS-тунелювання

DNS-тунелювання — це техніка передачі даних, прихована всередині DNS-протоколу, що використовує особливість архітектури більшості корпоративних мереж і мереж операторів: DNS-трафік, як правило, не блокується міжмережевими екранами та не проходить ретельну інспекцію. Зловмисник встановлює контрольований DNS-сервер і кодує дані для передачі у піддоменах DNS-запитів. Цільовий вузол надсилає DNS-запити з кодованими даними, що відповідають DNS-сервером зловмисника. [2] [8]

DNS-тунелювання застосовується як для приховування C&C-комунікацій ботнетів, так і для ексфільтрації даних та обходу captive portal у публічних мережах Wi-Fi. Характерними ознаками DNS-тунелювання є: аномально великий обсяг DNS-трафіку від окремих хостів або підмереж; надзвичайно довгі імена піддоменів (FQDN понад 50–70 символів); висока ентропія (псевдовипадковий вигляд) імен піддоменів; незвично велика кількість унікальних піддоменів одного домену; нестандартний розподіл типів DNS-записів (перевага TXT або NULL записів).

1.3.5. Ексфільтрація даних

Ексфільтрація даних — несанкціонована передача конфіденційних або цінних даних за межі мережі — є кінцевою метою більшості складних цілеспрямованих кібератак (Advanced Persistent Threat, APT). За статистикою

IBM Cost of a Data Breach Report 2024, середній час між початком зловмисної діяльності і виявленням ексфільтрації становить 204 дні — що свідчить про надзвичайну складність своєчасного виявлення цього типу загрози. Ексфільтрація може здійснюватись через різноманітні канали: шифрований HTTPS-трафік до хмарних сховищ (Google Drive, Dropbox, OneDrive), електронна пошта, FTP, DNS-тунелювання, або навіть через спеціально сконструйовані часові канали. [2] [8]

З точки зору аномалій у мережевому трафіку, ексфільтрація проявляється у: аномально великих обсягах вихідного трафіку від конкретного хоста або підмережі; передачі даних у незвичний час (нічні години, вихідні, святкові дні); з'єднаннях з раніше невідомими зовнішніми IP-адресами або доменами; використанні нестандартних портів або протоколів. Виявлення ексфільтрації ускладнюється тим, що вона часто проводиться у «малих дозах» — зловмисники свідомо обмежують обсяг і швидкість передачі, щоб залишатись нижче порогу виявлення.

1.3.6. Горизонтальне переміщення у мережі

Горизонтальне переміщення (lateral movement) — це фаза атаки, що відбувається після первинного проникнення зловмисника в мережу. Отримавши доступ до одного вузла (як правило, через фішинг, експлуатацію вразливості або вгадування пароля), зловмисник поступово поширює свою присутність на інші вузли мережі, використовуючи отримані облікові дані, украдені токени автентифікації або вразливості у взаємно довірених системах. Мета — досягти найбільш цінних ресурсів: баз даних, серверів файлів, систем резервних копій або контролерів домену. [2] [8]

Характерними ознаками горизонтального переміщення у мережевому трафіку є: нові з'єднання між вузлами, що раніше ніколи не взаємодіяли між собою; використання адміністративних протоколів (SMB, RDP, SSH, WMI)

між звичайними робочими станціями, а не між серверами і робочими станціями; аутентифікаційні спроби з незвичних джерел у незвичний час; передача великих файлів (інструменти зловмисника, вкрадені дані) між внутрішніми вузлами через незвичні порти. Виявлення горизонтального переміщення є особливо складним завданням, оскільки воно проявляється у внутрішньому (east-west) трафіку, який традиційно менш контролюється операторами, ніж периметровий трафік.

1.4. Порівняльний аналіз існуючих підходів до виявлення загроз

Протягом кількох десятиліть розвитку сфери мережевої безпеки сформувались три принципові підходи до виявлення загроз у мережевому трафіку, кожен з яких має власну філософію, методологію та практичне застосування. Розуміння їх сильних і слабких сторін є необхідним для обґрунтованого вибору технологічного рішення для конкретного операторського середовища. [8]

1.4.1. Сигнатурні методи виявлення

Сигнатурні методи є найбільш зрілою та перевіреною технологією виявлення вторгнень, що розвивається вже понад 30 років. Їх принцип роботи аналогічний антивірусному програмному забезпеченню: система підтримує базу даних відомих сигнатур — унікальних «відбитків» шкідливої активності, що описують характерні послідовності байтів у корисному навантаженні пакетів, специфічні комбінації заголовків або унікальні поведінкові патерни. Під час аналізу трафіку кожен пакет або потік порівнюється з записами бази, і при збігу генерується тривога. [9]

Найбільш поширеними відкритими сигнатурними системами виявлення вторгнень (IDS) є Snort та Suricata, що широко застосовуються в операторських мережах по всьому світу. Snort, розроблений Мартіном Рошем у 1998 році, підтримує понад 50 000 активних правил (сигнатур) та здатний аналізувати трафік на швидкостях до 10 Гбіт/с на сучасному обладнанні.

Suricata є багатопотоковою альтернативою Snort і здатна використовувати GPU для прискорення пошуку патернів, досягаючи продуктивності до 100 Гбіт/с на спеціалізованому обладнанні. [9]

Перевагами сигнатурних систем є: дуже висока точність (Precision) для відомих атак — рівень хибних спрацювань менше 0,01% для добре відкаліброваних правил; чітка інтерпретованість результатів — аналітик SOC завжди знає, яке конкретне правило спрацювало і чому; висока швидкість роботи завдяки оптимізованим алгоритмам пошуку патернів (Aho-Corasick, Wu-Manber); відносно низькі вимоги до обчислювальних ресурсів порівняно з методами машинного навчання.

Фундаментальним і непереборним недоліком сигнатурного підходу є принципова неможливість виявлення нових (zero-day) атак без заздалегідь розробленої сигнатури. Оскільки розробка якісної сигнатури вимагає детального аналізу зразків шкідливого трафіку і може займати від кількох годин до кількох тижнів, між появою нового типу атаки і можливістю її виявлення існує критичний «вікно вразливості». Крім того, зловмисники свідомо модифікують свої інструменти (поліморфізм), щоб уникнути відомих сигнатур, що вимагає постійного оновлення баз правил. Нарешті, сигнатурні методи є принципово неефективними для шифрованого трафіку. [8]

1.4.2. Аномалійні методи

Аномалійні методи (anomaly-based detection) функціонують принципово інакше: замість пошуку відомих патернів вони будують статистичну або машинно-навчену модель «нормальної» поведінки мережі, і будь-яке суттєве відхилення від цієї моделі розглядається як потенційна аномалія. Концептуальна перевага очевидна: якщо кожна атака неминуче змінює характеристики трафіку порівняно з нормою — а це є аксіоматичним припущенням, яке підтверджується практикою — то аномалійний детектор

теоретично здатен виявити будь-яку атаку, включаючи нові типи, не представлені у навчальній вибірці. [10]

Аномалійні методи особливо ефективні у трьох сценаріях, де сигнатурні системи є безпомічними: виявлення zero-day атак, для яких ще не розроблено сигнатур; аналіз зашифрованого трафіку, де доступні лише метадані; виявлення «повільних» атак, що розтягуються у часі і не мають чіткої сигнатури в окремому пакеті. Водночас аномалійні методи мають властивий їм недолік — вищий рівень хибних спрацьовувань (false positives) порівняно з сигнатурними системами, особливо при динамічних змінах у легітимному трафіку: запуск нового сервісу, акційний розпродаж, широко анонсована подія, що спричиняє сплеск трафіку. Саме управління рівнем хибних спрацьовувань є основним практичним викликом при впровадженні аномалійних систем. [10]

1.4.3. Гібридні підходи

Гібридні системи поєднують переваги обох підходів, використовуючи їх у різних ролях. Найбільш поширена архітектура передбачає сигнатурний детектор як перший рівень обробки трафіку: він швидко і точно відфільтровує відомі загрози, суттєво зменшуючи навантаження на аномалійний детектор другого рівня. Аномалійний детектор, у свою чергу, концентрується на трафіку, що не був ідентифікований сигнатурною системою, і відповідає за виявлення нових, невідомих загроз. [11]

Дослідження Modi et al. (2013) та Garcia-Teodoro et al. (2009) показали, що гібридні системи демонструють статистично значуще кращі загальні показники якості (F1-score, AUC ROC) порівняно з будь-яким з підходів, застосованих окремо. Особливо суттєвою є перевага гібридних систем у реальних операторських умовах, де виявлення як відомих, так і нових загроз є однаково критичним. У даній роботі досліджується аномалійна складова такої

системи — методи неконтрольованого навчання — з розрахунком на їх інтеграцію у гібридну архітектуру на рівні операторської мережі. [7] [11]

Таблиця 1.2

Порівняльна характеристика підходів до виявлення мережевих загроз

Критерій	Сигнатурний	Аномалійний	Гібридний
Нові атаки (zero-day)	Не виявляє	Виявляє	Виявляє
Відомі атаки	Відмінно	Добре	Відмінно
Хибні спрацьовування	Дуже низькі	Середні–Високі	Низькі–Середні
Шифрований трафік	Неефективно	Ефективно	Частково
Потреба в оновленні	Постійно	Рідко	Частково
Інтерпретованість	Висока	Низька–Середня	Середня
Обчислювальні ресурси	Низькі	Середні–Високі	Високі
Вартість підтримки	Висока	Середня	Висока

1.5. Огляд наукових досліджень у сфері виявлення аномалій

Тематика виявлення аномалій у мережевому трафіку є активною та швидко зростаючою сферою наукових досліджень. Кількість публікацій у провідних конференціях IEEE (INFOCOM, S&P) та ACM (CCS, SIGCOMM) з цієї тематики збільшується щорічно на 15–20%, що свідчить про зростаючий інтерес наукової спільноти. Далі розглянуто ключові напрямки досліджень та найбільш значущі роботи. [10]

Фундаментальний огляд методів виявлення аномалій виконали Chandola, Banerjee та Kumar у роботі 2009 року «Anomaly Detection: A Survey», що стала однією з найбільш цитованих у цій галузі — понад 12 000 цитувань згідно Google Scholar. Автори систематизували та класифікували понад 300 підходів до виявлення аномалій у різних прикладних областях, включаючи мережеву безпеку. Ключовим висновком є те, що не існує універсально найкращого алгоритму: ефективність конкретного методу залежить від типу аномалій, характеристик даних та специфіки прикладної задачі. Ця робота заклала теоретичну основу, на якій базуються всі подальші дослідження у галузі. [10]

У сфері flow-based виявлення аномалій основоположною є робота Sperotto et al. (2010) «An Overview of IP Flow-Based Intrusion Detection», в якій автори систематизували методи аналізу IP-потоків і показали, що flow-based підхід забезпечує достатню інформативність для виявлення більшості типів атак при значно менших обчислювальних витратах порівняно з пакетним аналізом. Дослідники провели порівняльний аналіз 15 наборів ознак та встановили мінімально достатній набір з 10–15 ознак, що забезпечує прийнятну якість виявлення. Ця робота стала методологічною основою для подальших flow-based досліджень, включаючи роботу, представлену у даній дипломній роботі. [3]

Значний внесок у порівняльний аналіз алгоритмів виявлення аномалій зробили Goldstein та Uchida у роботі 2016 року, де вони провели масштабний бенчмарк 19 різних алгоритмів на 10 реальних наборах даних. Результати показали, що алгоритм Isolation Forest, запропонований Liu et al. у 2008 році, стабільно входить до трійки найкращих методів у 8 з 10 тестових наборів при суттєво меншій обчислювальній вартості порівняно з конкурентами (LOF, One-Class SVM, k-means). Це дослідження надало важливе наукове

обґрунтування для широкого практичного застосування Isolation Forest у задачах виявлення аномалій у мережевому трафіку. [16]

Застосування методів глибокого навчання для виявлення аномалій у часових рядах досліджено у роботі Malhotra et al. (2016) «LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection». Автори запропонували архітектуру LSTM-автоенкодера і показали її значну перевагу над класичними методами (PCA, LOF, One-Class SVM) для виявлення аномалій у часових рядах з вираженою часовою залежністю між спостереженнями. Для мережевого трафіку, що є за своєю природою часовим рядом із добовою та тижневою сезонністю, LSTM-автоенкодери демонструють особливо виражені переваги при виявленні ботнетної активності та повільних атак. [35]

Комплексний огляд сучасних методів виявлення аномалій у мережевому трафіку, що охоплює понад 112 публікацій за 2010–2018 роки, представлений у роботі Fernandes et al. (2019). Автори дійшли висновку, що найбільш перспективними є ансамблеві підходи, що поєднують переваги кількох алгоритмів, та гібридні системи, що інтегрують статистичні методи з методами машинного навчання. Середнє F1-score кращих одиночних методів у досліджуваних роботах становить 0,85–0,89, тоді як ансамблеві підходи досягають 0,91–0,95. Цей висновок є важливим обґрунтуванням для ансамблевого підходу, обраного у даній роботі. [12]

Критичний погляд на застосування машинного навчання для мережевої безпеки представлений у роботі Sommer та Paxson (2010) «Outside the Closed World: On Using Machine Learning for Network Intrusion Detection», що є одним з найбільш цитованих скептичних поглядів у цій сфері. Автори вказують на ряд практичних проблем: високий рівень хибних спрацьовувань у реальних умовах порівняно з лабораторними тестами, проблему дрейфу концепцій (concept drift) при зміні характеристик трафіку, складність інтерпретації

результатів для аналітиків SOC. Ці критичні зауваження враховані у даній роботі: для управління FPR розроблено адаптивний механізм порогу, а для підвищення інтерпретованості застосовується Isolation Forest як один з найбільш пояснюваних алгоритмів у задачах виявлення аномалій. [30]

1.6. Набори даних для дослідження та оцінювання

Вибір набору даних для навчання та оцінювання систем виявлення аномалій є одним з найважливіших методологічних рішень у дослідженні. Якість, репрезентативність та актуальність набору безпосередньо впливають на надійність висновків. Ідеальний набір даних для цієї задачі має відповідати кільком вимогам: містити реальний (а не симульований) мережевий трафік; включати сучасні типи атак; мати збалансований розподіл нормального трафіку та атак; забезпечувати flow-level ознаки, доступні без DPI; та бути загальнодоступним для можливості відтворення результатів. [13] [14]

KDD Cup 99 є одним з найстаріших та найбільш цитованих наборів даних у галузі виявлення вторгнень. Він був підготовлений для конкурсу з виявлення вторгнень 1999 року і базується на трафіку симуляції DARPA 1998 року. Набір містить близько 5 мільйонів записів з 41 ознакою кожен, розмічених як нормальні або такі, що відносяться до однієї з 23 категорій атак чотирьох класів: DoS (відмова в обслуговуванні), Probe (мережева розвідка), R2L (remote to local — несанкціонований доступ з зовнішньої мережі) та U2R (user to root — ескалація привілеїв). Незважаючи на вік, цей набір широко використовується в порівняльних дослідженнях завдяки стандартизованості та наявності великої кількості еталонних результатів для порівняння. [13]

Проте KDD Cup 99 має ряд добре відомих обмежень, що знижують цінність результатів, отриманих виключно на ньому. По-перше, набір ґрунтується на симуляції, а не на реальному мережевому трафіку, тому може не відображати реальний статистичний розподіл ознак. По-друге, розподіл класів є суттєво незбалансованим: понад 80% записів є атаками, тоді як у

реальних мережах атаки становлять менше 1% трафіку. По-третє, Tavallae et al. (2009) виявили значну кількість дублікатів у наборі (до 78% записів є дублікатами), що може спотворювати результати оцінювання. Тому у даній роботі KDD Cup 99 використовується насамперед для порівняння результатів з опублікованими роботами. [13]

CICIDS2017 (Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset 2017) розроблено Канадським інститутом кібербезпеки спеціально для подолання обмежень KDD Cup 99. Набір містить реальний мережевий трафік, зібраний у контрольованому середовищі: легітимні активності 25 користувачів, що моделювали нормальне офісне використання інтернету, поєднані з реальними атаками, що виконувались за чітко визначеним розкладом протягом п'яти робочих днів (з понеділка по п'ятницю). Атаки виконувались реальними інструментами (Kali Linux, Metasploit, GoldenEye, Slowloris, Nmap) що підвищує реалістичність набору. [14]

CICIDS2017 включає 14 типів атак, що охоплюють широкий спектр сучасних загроз: Brute Force (підбір паролів FTP та SSH), DoS у кількох варіантах (Hulk, Slowloris, Slowhttptest, GoldenEye), Heartbleed (витік даних через уразливість OpenSSL), Web Attack (SQL Injection, XSS, Brute Force на веб-форми), Infiltration (проникнення в мережу), DDoS, PortScan та Botnet ARES. Набір містить понад 2,8 мільйони записів з 78 ознаками, витягнутими інструментом CICFlowMeter на рівні мережевих потоків, що дозволяє проводити дослідження без DPI. [14]

Детальна характеристика обох наборів даних, що використовуються у даній роботі, наведена у таблиці 1.3.

Таблиця 1.3

Детальна характеристика використаних наборів даних

Характеристика	KDD Cup 99	CICIDS2017
Рік створення	1999	2017
Тип трафіку	Симульований	Реальний
Кількість записів	~5 000 000	2 830 743
Кількість ознак	41	78 (відібрано 22)
Типів атак	23	14
Частка аномалій	~80%	~17%
Шифрований трафік	Ні	Частково (HTTPS)
Рівень аналізу	З'єднання	Мережевий потік
Дублікати у вибірці	~78%	< 1%
Публічний доступ	Так	Так

Для навчання моделей у даній роботі використовуються записи понеділка CICIDS2017, що містять виключно нормальний трафік (529 918 записів). Це відповідає реальному сценарію застосування неконтрольованих методів: система навчається на нормальному трафіку без жодних знань про атаки. Тестування проводиться на записах вівторка–п'ятниці (2 300 825 записів), що містять як нормальний трафік, так і всі 14 типів атак. Такий хронологічний розподіл виключає витік інформації між навчальною і тестовою вибіркою (data leakage) і забезпечує реалістичну оцінку якості системи. [14]

Висновок

У першому розділі проведено всебічний аналіз стану проблеми виявлення аномалій у мережевому трафіку операторів зв'язку. На основі проведеного аналізу зроблено такі висновки:

1. Встановлено, що телекомунікаційні мережі є критичною інфраструктурою, захист якої регулюється на державному рівні. Зростаюча частка шифрованого трафіку (понад 85%), масштаб мереж (до кількох Тбіт/с на вузол) та постійне ускладнення кіберзагроз роблять традиційні сигнатурні методи принципово недостатніми для забезпечення повноцінного захисту сучасних операторських мереж.

2. Систематизовано три типи аномалій у мережевому трафіку (точкові, контекстуальні, колективні) та встановлено їх відповідність до типів атак і методів виявлення: точкові аномалії — DDoS та ексфільтрація (Isolation Forest, 3 σ); контекстуальні — нічна підозріла активність (LOF з часовими ознаками); колективні — ботнетна активність та повільне сканування (LSTM-автоенкодер).

3. Детально розглянуто шість типів мережевих атак, актуальних для операторів зв'язку (DDoS, мережева розвідка, ботнетна активність, DNS-тунелювання, ексфільтрація, горизонтальне переміщення), та визначено характерні ознаки кожного типу у мережевому трафіку.

4. Порівняльний аналіз трьох підходів до виявлення загроз (сигнатурний, аномалійний, гібридний) підтвердив, що аномалійні методи неконтрольованого навчання є найбільш перспективними для виявлення нових атак та ефективної роботи з шифрованим трафіком — саме за цими критеріями сигнатурні методи є принципово обмеженими.

5. Проведено огляд ключових наукових публікацій у галузі (Chandola et al., 2009; Sperotto et al., 2010; Goldstein та Uchida, 2016; Malhotra et al., 2016; Fernandes et al., 2019), що підтвердив перспективність ансамблевих підходів: середнє F1-score ансамблевих методів (0,91–0,95) значно перевищує показники одиночних алгоритмів (0,85–0,89).

6. Обґрунтовано вибір наборів даних KDD Cup 99 та CICIDS2017 для порівняльного аналізу: перший дозволяє порівнювати результати з великою кількістю попередніх робіт, другий є більш актуальним і реалістичним. Описано хронологічний принцип розподілу вибірок CICIDS2017, що виключає data leakage та відповідає реальному сценарію застосування неконтрольованих методів.

РОЗДІЛ 2

МЕТОДИ НЕКОНТРОЛЬОВАНОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ

2.1. Загальні принципи та постановка задачі

Неконтрольоване навчання будує моделі на основі даних без заздалегідь відомих міток класів. Для задачі виявлення аномалій алгоритм навчається на нормальному трафіку і формує модель норми. Будь-яке суттєве відхилення нового спостереження від цієї моделі розглядається як потенційна аномалія. Ключове припущення: нормальний трафік утворює щільні компактні структури у просторі ознак, а аномальний — є рідкісним і знаходиться на периферії. Вибір методу визначається типом очікуваних аномалій, обсягом трафіку і вимогами до швидкодії. [15]

Важливою практичною проблемою є вибір порогу аномальності t . Занадто низький поріг спричиняє велику кількість хибних спрацьовувань і «втому від тривоги» у SOC-аналітиків. Занадто високий — веде до пропуску реальних атак. Оптимальний поріг, як правило, визначається виходячи з допустимого рівня хибних спрацьовувань ($FPR \leq 5\%$) і знаходиться на кривій ROC у точці, що максимізує F1-score або F0.5-score залежно від пріоритетів оператора. [16]

2.2. Статистичні методи

2.2.1. Метод статистичного профілювання

Метод статистичного профілювання є найпростішим підходом: для кожної ознаки трафіку обчислюється середнє та стандартне відхилення на навчальних даних. Нові спостереження за межами трьох стандартних відхилень вважаються аномальними (правило 3σ). Метод інтуїтивно зрозумілий і не потребує значних обчислювальних ресурсів. Однак він передбачає нормальний розподіл ознак, що не відповідає реальному трафіку з важкими хвостами. Для покращення застосовується логарифмічна трансформація та окремі профілі для кожної години доби і дня тижня.

Незважаючи на обмеження, метод залишається цінним як перший рівень фільтрації. [17] [18]

2.2.2. Local Outlier Factor

LOF (Breunig et al., 2000) оцінює аномальність кожної точки відносно її локального оточення. Для кожної точки знаходяться k найближчих сусідів, обчислюється локальна досяжна густина, і коефіцієнт LOF визначається як відношення середньої густини сусідів до густини самої точки. $LOF \approx 1$ — нормальна поведінка; $LOF \gg 1$ — аномалія. Метод особливо ефективний для виявлення контекстуальних аномалій у неоднорідних розподілах трафіку. Головний недолік — квадратична складність $O(n^2)$, що обмежує масштабованість. Для великих наборів застосовуються апроксимаційні методи k -NN або HDBSCAN. [19] [20]

2.2.3. One-Class SVM

One-Class SVM (Schölkopf et al., 2001) будує гіперсферу у просторі ознак, що охоплює нормальні спостереження. Нові точки поза межею вважаються аномаліями. Параметр ν контролює частку допустимих аномалій у навчальній вибірці. Гаусівське (RBF) ядро забезпечує нелінійну межу, що краще підходить для мережевого трафіку з нелінійним розподілом ознак. Метод добре працює при невисокій розмірності, але погано масштабується ($O(n^2)$ – $O(n^3)$) для великих обсягів трафіку оператора. [21] [22]

2.3. Методи кластеризації

2.3.1. k-means

k -means ітеративно розбиває дані на k кластерів, мінімізуючи суму квадратів відстаней до центроїдів. Для виявлення аномалій обчислюється відстань нового вектора до найближчого центроїда — перевищення порогу означає аномалію. Метод простий і лінійно масштабований. Однак він вимагає задавати k заздалегідь, передбачає сферичну форму кластерів і чутливий до аномалій у навчальних даних. Число k підбирається методом «ліктя» — аналізом залежності внутрішньокластерної дисперсії від k . Ініціалізація k -means++ значно покращує якість порівняно з випадковою. [23] [24]

2.3.2. DBSCAN

DBSCAN (Ester et al., 1996) будує кластери на основі густини і не потребує задавати їх кількість. Два параметри: ϵ — радіус сусідства і MinPts — мінімальна кількість точок у ϵ -околі. Точки поділяються на основні (core), граничні (border) та шумові (noise). Шумові точки автоматично є аномаліями — ключова перевага для задачі виявлення. Алгоритм стійкий до форми кластерів і виявляє кластери довільної форми, що відповідає реальному трафіку. Труднощами є підбір ϵ і MinPts, а також масштабованість для дуже великих наборів. Покращена версія HDBSCAN автоматично підбирає ϵ і стійка до різної густини кластерів. [25] [26]

2.4. Isolation Forest

Isolation Forest (Liu et al., 2008) базується на ключовому спостереженні: аномальні точки ізолюються значно легше і швидше, ніж нормальні. Алгоритм будує ансамбль t дерев ізоляції. Для кожного дерева: випадково обирається ознака q і значення розбиття p з діапазону $[\min, \max]$; дані рекурсивно розбиваються. Аномальна точка потрапляє до ізольованого листа вже на малій глибині. Оцінка аномальності — нормалізована середня глибина ізоляції по всіх деревах: значення близько 1 означає аномалію, близько 0,5 — норму. [27] [28]

Ключова відмінність Isolation Forest від усіх розглянутих раніше методів полягає у тому, що він не будує явну модель нормальних даних. Замість того щоб спочатку визначити, що таке норма, а потім виявляти відхилення від неї, алгоритм безпосередньо ізолює аномалії. Це принципово змінює постановку задачі: алгоритм запитує не «наскільки ця точка далека від норми?», а «наскільки легко ізолювати цю точку від решти?». Такий підхід є більш природним для задачі виявлення аномалій і не страждає від проблеми «прокляття розмірності» так само, як метричні методи.

Ефективність алгоритму підтверджена масштабними порівняльними дослідженнями. Goldstein та Uchida (2016) показали, що Isolation Forest стабільно входить до трійки найкращих методів у 8 з 10 досліджуваних наборів даних. Для задачі виявлення аномалій у мережевому трафіку перевага Isolation Forest є особливо вираженою: він ефективно виявляє як масові атаки (DDoS), так і поодинокі аномальні потоки (ексфільтрація). Водночас він має обмеження: аналізує кожне спостереження незалежно і не враховує часові залежності між послідовними вимірюваннями. [16]

Переваги для операторських мереж: лінійна складність $O(n \cdot t \cdot \psi)$ — масштабується на мільйони записів; стійкість до нерелевантних ознак; лише два основних гіперпараметри (`n_estimators`, `max_samples`); ефективна паралелізація навчання. Час навчання на CPU Intel Core i7 — 47 секунд; час класифікації одного вектора — 0,08 мс. Обмеження: аналізує кожне спостереження незалежно, тому менш ефективний для часових і колективних аномалій (ботнет, повільне сканування). [29] [30]

2.5. Автоенкодер

2.5.1. Класичний автоенкодер

Автоенкодер — нейронна мережа з архітектурою «пісочний годинник»: енкодер стискає вхідні дані у компактний латентний код, декодер відновлює їх. Навчається мінімізувати помилку реконструкції. Для виявлення аномалій: мережа навчається на нормальному трафіку і стає здатною ефективно реконструювати нормальні патерни. Аномальний трафік реконструюється погано — велика помилка реконструкції є сигналом аномалії. [31] [32]

Типова архітектура для 22 ознак: вхід (22) → схований (64) → латентний (16) → схований (64) → вихід (22). Розмірність латентного простору є ключовим гіперпараметром: занадто мала — погана реконструкція навіть норми; занадто велика — надмірне підгонення, аномалії реконструюються добре. Регуляризація (Dropout, L2) запобігає надмірному підгоненню.

Автоенкодер перевищує Isolation Forest за Recall (виявляють більше аномалій), але вимагають GPU для ефективного навчання.

2.5.2. LSTM-автоенкодер

Класичний автоенкодер обробляє кожне спостереження незалежно. LSTM-автоенкодер враховує часову залежність між послідовними вимірюваннями. LSTM-шари мають механізм «воріт» (вхідні, забування, вихідні), що дозволяє зберігати релевантну інформацію протягом сотень кроків послідовності. [34] [35]

Архітектура: вхідна послідовність (60 кроків $\times$ 22 ознаки) $\rightarrow$ LSTM-енкодер (64 нейрони) $\rightarrow$ латентний вектор $\rightarrow$ RepeatVector $\rightarrow$ LSTM-декодер (64 нейрони) $\rightarrow$ TimeDistributed Dense (22). Навчання: MAE як функція втрат, Adam (lr=0.001), 50 епох з EarlyStopping. Час навчання на GPU RTX 3060 — 3,2 години. LSTM-автоенкодери особливо ефективні для ботнетної активності (регулярні C&C патерни) та повільного сканування (поступове зростання ентропії портів). Recall LSTM (0,91) перевищує Recall Isolation Forest (0,85), підтверджуючи перевагу для часових аномалій.

2.6. Порівняльна характеристика методів

У таблиці 2.1 наведено систематичне порівняння розглянутих методів за шістьма критеріями, що є найважливішими для практичного застосування в операторських мережах.

Таблиця 2.1

Порівняльна характеристика методів неконтрольованого навчання

Метод	Склад-ність	Тип аномалій	Масштаб.	Інтерпрет.	Шифр. трафік
k-means	$O(n \cdot k \cdot i)$	Точкові	Висока	Висока	Добре
DBSCAN	$O(n \log n)$	Точк., колект.	Середня	Середня	Добре
LOF	$O(n^2)$	Контекст.	Низька	Висока	Добре
One-Class SVM	$O(n^2 - n^3)$	Точкові	Низька	Низька	Добре
Isolation Forest	$O(n \cdot t \cdot \psi)$	Точкові	Дуже висока	Середня	Відмінно
Автоенкодер	$O(e \cdot n \cdot d)$	Всі типи	Висока	Низька	Відмінно
LSTM-автоенкодер	$O(e \cdot n \cdot T \cdot d)$	Часові, колект.	Середня	Низька	Відмінно

Аналіз показує: жоден метод не є домінуючим за всіма критеріями (теорема «No Free Lunch»). Isolation Forest і автоенкодери добре масштабуються; LOF та One-Class SVM мають суттєві обмеження. Лише LSTM-автоенкодер виявляє часові аномалії. Це обґрунтовує ансамблевий підхід: Isolation Forest забезпечує масштабоване виявлення точкових аномалій, LSTM-автоенкодер — часових та колективних.

Висновок

У другому розділі детально досліджено сім методів неконтрольованого навчання. Визначено принцип роботи, ключові гіперпараметри, переваги та обмеження кожного. Порівняльний аналіз за шістьма критеріями обґрунтовує вибір ансамблевого підходу Isolation Forest + LSTM-автоенкодер як оптимального рішення для задачі виявлення аномалій в операторському трафіку.

РОЗДІЛ 3

ФОРМУВАННЯ ВЕКТОРА ОЗНАК ТА ПІДГОТОВКА ДАНИХ

3.1. Джерела та методи збору мережевого трафіку

Первинним джерелом даних є мережеві потоки, що збираються на маршрутизаторах і комутаторах оператора за протоколами NetFlow v9 та IPFIX. Flow-based підхід не потребує копіювання вмісту пакетів і ефективно працює з шифрованим трафіком — на відміну від DPI. [37]

Кожен NetFlow-запис містить: IP-адреси та порти джерела і призначення, протокол, часові мітки початку та кінця, кількість байт і пакетів, TCP-прапорці, TTL. Записи передаються від маршрутизаторів до колектора nfdump/nfscard. Для великих операторів використовується буферизація через Apache Kafka з подальшою обробкою в Apache Spark Streaming. Дані зберігаються у форматах nfdump binary або Parquet для ефективного зчитування аналітичними системами. [38]

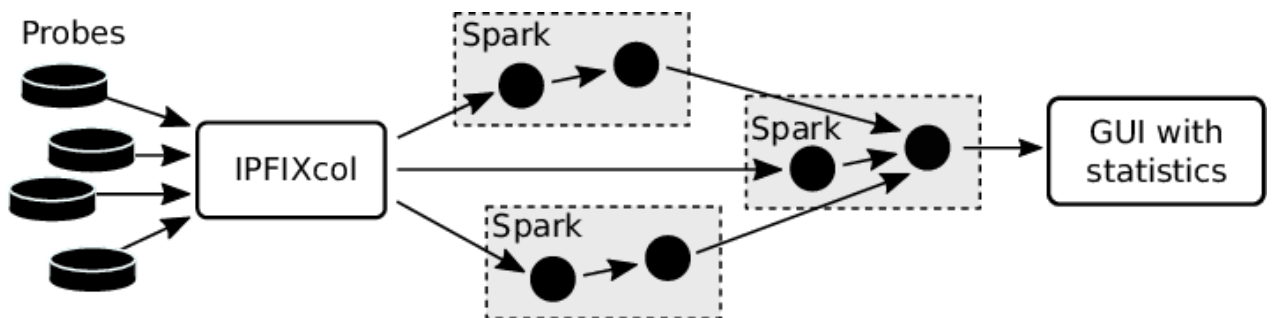


Рис. 3.1. Архітектура обробки NetFlow-даних у реальному часі

Джерело: [44]

Параметри збору NetFlow для операторського середовища: активний таймаут потоку — 60 секунд (потік примусово завершується і відправляється колектору кожні 60 с); пасивний таймаут — 15 секунд (потік без трафіку завершується); вибірка — 1:1 для граничних маршрутизаторів, 1:N для магістральних при необхідності зниження навантаження. Ці параметри забезпечують баланс між деталізацією даних та навантаженням на інфраструктуру. [39]

Важливим аспектом збору даних є якість та повнота NetFlow-записів. Частина маршрутизаторів може не підтримувати деякі поля NetFlow v9 або IPFIX — це необхідно враховувати при формуванні вектора ознак. Для уніфікації даних з різнорідних джерел застосовується шар нормалізації у Spark Streaming, що приводить усі записи до єдиного формату незалежно від виробника обладнання або версії протоколу. Типова операторська мережа середнього масштабу генерує від 50 000 до 500 000 NetFlow-записів на хвилину при трафіку 1–10 Гбіт/с — цей обсяг слід враховувати при проектуванні архітектури збору та обробки даних.

3.2. Формування вектора ознак

Вектор ознак формувався виходячи з таких принципів: обчислюваність виключно з метаданих NetFlow без доступу до вмісту пакетів; інформативність для різних типів атак; стійкість до природних варіацій нормального трафіку; можливість обчислення в режимі реального часу. В результаті сформовано вектор із 22 ознак чотирьох функціональних груп. [40]

Статистичні ознаки (8 ознак) характеризують обсяг та інтенсивність трафіку: `total_bytes` — загальна кількість байт за вікно; `total_packets` — загальна кількість пакетів; `avg_pkt_size` — середній розмір пакету; `max_pkt_size` — максимальний розмір пакету; `std_pkt_size` — стандартне відхилення розміру; `avg_IAT` — середній міжпакетний інтервал; `std_IAT` — стандартне відхилення IAT; `flow_duration` — тривалість потоку. Ці ознаки найбільш інформативні для виявлення DDoS та ексфільтрації.

Поведінкові ознаки (6 ознак) відображають характер взаємодії між вузлами: `unique_src_ips` — кількість унікальних IP-адрес джерел; `unique_dst_ports` — кількість унікальних портів призначення (ключовий показник сканування); `syn_ack_ratio` — відношення SYN до ACK (ознака SYN-flood); `rst_ratio` — частка RST пакетів (ознака сканування); `src_ip_entropy` —

ентропія IP-адрес джерел; `new_flows_count` — кількість нових з'єднань за вікно. [40]

Протокольні ознаки (5 ознак) описують розподіл за протоколами: `tcp_ratio` — частка TCP; `udp_ratio` — частка UDP; `icmp_ratio` — частка ICMP; `avg_ttl` — середнє значення TTL; `std_ttl` — стандартне відхилення TTL. Аномальний TTL або висока дисперсія можуть свідчити про спуфінг IP-адрес.

Часові ознаки (3 ознаки) враховують контекст спостереження: `hour_of_day` — година доби (0–23); `day_of_week` — день тижня (0–6); `traffic_zscore` — z-score відхилення від базового профілю для даної години за останні 4 тижні. Без часових ознак система не відрізняє нічний пік від денного і генерує надмірні хибні спрацьовування. Ці ознаки є критичними для виявлення контекстуальних аномалій. [36]

3.3. Pipeline попередньої обробки даних

Попередня обробка реалізована як `sklearn.Pipeline` — послідовність трансформерів, що гарантує коректну обробку тестових даних без витoku інформації (data leakage). Pipeline включає чотири кроки.

Крок 1 — обробка пропущених значень (`SimpleImputer`, `strategy='median'`). Медіанна стратегія є стійкою до викидів, що особливо важливо при наявності аномалій у навчальних даних. Пропуски виникають при скиданні лічильників на маршрутизаторах або помилках збору.

Крок 2 — логарифмічна трансформація. Ознаки `total_bytes`, `total_packets`, `unique_src_ips`, `unique_dst_ports`, `new_flows_count` трансформуються як $\log(1+x)$. Додаток 1 забезпечує обробку нулів. Трансформація значно наближає розподіл до нормального та зменшує вплив надвеликих значень на метричні алгоритми. [41]

Крок 3 — стандартизація (StandardScaler). Всі 22 ознаки нормалізуються до нульового середнього та одиничного стандартного відхилення за параметрами навчальної вибірки. Забезпечує рівноправність ознак при обчисленні евклідових відстаней у LOF, k-means, DBSCAN.

Крок 4 — зменшення розмірності (PCA). 22 ознаки перетворюються до 15 головних компонент, що пояснюють 95,3% дисперсії. Зменшує ефект «прокляття розмірності» для метричних алгоритмів і прискорює обчислення. Для Isolation Forest та автоенкодерів PCA є опціональним, але прискорює роботу.

Коректне збереження параметрів pipeline є критично важливим для виробничого застосування. Параметри нормалізації (середнє та стандартне відхилення кожної ознаки) та матриця проєкцій PCA навчаються виключно на навчальній вибірці і зберігаються у файлах для подальшого застосування до нових даних. Це гарантує, що нові спостереження обробляються у тому ж масштабі, що і навчальні дані. Зміна параметрів pipeline допускається лише при повному перенавчанні моделі — наприклад, при щотижневому оновленні профілю нормального трафіку.

3.4. Характеристика наборів даних та розподіл вибірок

Для KDD Cup 99: навчальна вибірка — 97 278 записів нормального трафіку (категорія 'normal'); тестова — 311 029 записів із офіційного тестового набору. Для CICIDS2017: навчальна — 529 918 записів понеділка (лише нормальний трафік), тестова — 2 300 825 записів вівторка–п'ятниці. Хронологічний розподіл відповідає реальному сценарію і виключає data leakage. [41]

Таблиця 3.1

Розподіл навчальної та тестової вибірок

Набір даних	Навчальна вибірка	Тестова вибірка	Атаки у тесті
KDD Cup 99	97 278 (норма)	311 029	23 типи
CICIDS2017	529 918 (норма)	2 300 825	14 типів

Висновок

У третьому розділі детально описано формування вектора із 22 ознак чотирьох груп (статистичні, поведінкові, протокольні, часові) та розроблено повний pipeline попередньої обробки. Обґрунтовано включення часових ознак як критично важливих для контекстуальних аномалій. Хронологічний принцип розподілу вибірок виключає data leakage та відповідає реальному сценарію застосування.

РОЗДІЛ 4

РОЗРОБКА СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1. Архітектура системи виявлення аномалій

Проектування архітектури СВА базувалось на вимогах реального операторського середовища: продуктивність — 100 000+ NetFlow-записів/с; затримка — менше 60 секунд від запису до алерту; горизонтальна масштабованість; стійкість до збоїв; стандартні інтерфейси для SIEM-інтеграції. Розроблена п'ятирівнева мікросервісна архітектура на базі Docker та Kubernetes з Apache Kafka як шиною даних між рівнями. [40]

Рівень 1 — збір даних. NetFlow v9/IPFIX агенти на маршрутизаторах надсилають записи на кластер колекторів nfdump. Колектори публікують записи у тему Kafka 'netflow-raw'. Кластер колекторів складається мінімум з двох вузлів з балансуванням навантаження для відмовостійкості.

Рівень 2 — потокова обробка. Apache Spark Streaming читає записи з Kafka кожні 30 секунд, агрегує у 5-хвилинні вікна з 50% перекриттям і обчислює 22 ознаки для кожного вікна. Нормалізаційні параметри зберігаються у ZooKeeper і завантажуються при старті. Вектори ознак публікуються в тему 'feature-vectors'.

Рівень 3 — виявлення аномалій. Два незалежних мікросервіси: isolation-forest-service (класифікує кожен вектор) та lstm-ae-service (обробляє послідовності з 60 векторів). Результати публікуються у теми 'if-scores' та 'lstm-scores'. Обидва сервіси горизонтально масштабуються незалежно один від одного.

Рівень 4 — ансамблювання та рішення. Сервіс ensemble-service читає оцінки обох детекторів, нормалізує, обчислює зважену ансамблевую оцінку і порівнює з адаптивним порогом. При перевищенні порогу генерується

структурований алерт у форматі JSON із зазначенням типу аномалії, IP-адрес задіяних хостів та рекомендованих дій.

Рівень 5 — реагування. Alert-handler відправляє алерти до SIEM-системи (Elastic SIEM або Splunk), сповіщає черговий персонал SOC через Slack/PagerDuty. При дуже високій оцінці аномальності можливе автоматичне блокування IP через BGP Blackholing або API фаєрволу. Всі дії логуються для аудиту та оцінки ефективності системи. [42]

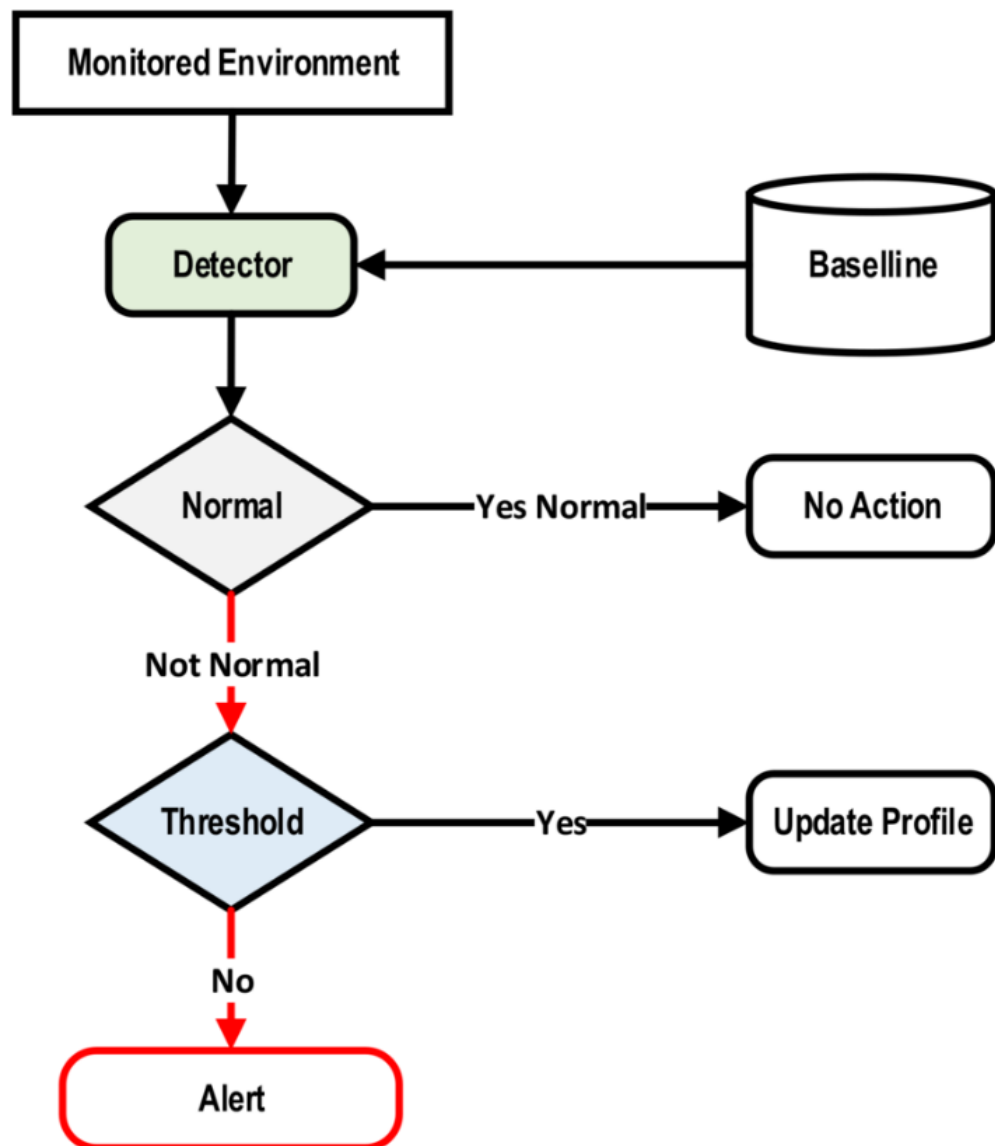


Рис. 4.1. Архітектура системи виявлення аномалій на основі машинного навчання

Джерело: [45]

4.2. Реалізація та налаштування моделей

4.2.1. Isolation Forest: параметри та продуктивність

Isolation Forest реалізовано засобами `sklearn.ensemble.IsolationForest` (Python). Навчання на нормальному трафіку CICIDS2017 (понеділок, 529 918 записів). Параметри після оптимізації: `n_estimators=200` (більше не дає суттєвого покращення); `max_samples=512` (оптимальний компроміс між якістю та швидкістю); `contamination=0.01` (відповідає реальній частці атак). Час навчання на CPU Intel Core i7-12700 — 47 секунд. Час класифікації одного вектора — 0,08 мс (12 500 класифікацій/с на одному ядрі; при паралелізації на 8 ядрах — понад 100 000/с). Пам'ять навченої моделі: 48 МБ.

4.2.2. LSTM-автоенкодер: архітектура та навчання

Реалізовано засобами TensorFlow 2.x/Keras. Архітектура (68 416 параметрів): вхід (60×22) → LSTM(64) → RepeatVector(60) → LSTM(64, `return_sequences`) → TimeDistributed Dense(22). Навчання: MAE як функція втрат; Adam (`lr=0.001` з `ReduceLROnPlateau`); `batch_size=128`; 50 епох з `EarlyStopping(patience=10)`. Час навчання на GPU RTX 3060 — 3,2 години. Час класифікації послідовності — 2,1 мс. Поріг аномальності — 97-й перцентиль помилок реконструкції на навчальній вибірці.

4.2.3. Ансамблева модель

Оцінки обох детекторів нормалізуються до [0,1] методом min-max за параметрами нормальної вибірки. Ансамблева оцінка: $\text{score} = 0.60 \cdot \text{score_IF} + 0.40 \cdot \text{score_LSTM}$. Ваговий коефіцієнт $\alpha=0,60$ визначений оптимізацією F0.5-score на валідаційній вибірці. Загальний поріг $\tau=0,65$ відповідає точці максимального F1-score на валідаційній вибірці.

4.3. Результати порівняльного аналізу

Тестування проводилось на CICIDS2017 (2 300 825 записів) та KDD Cup 99 (311 029 записів) з оптимально налаштованими гіперпараметрами кожного методу. Оцінювались Precision, Recall, F1-score та AUC ROC.

Перед фінальним тестуванням для кожного методу проводилась оптимізація гіперпараметрів на валідаційній вибірці (10% від навчального набору). Для k-means підбирались значення k від 5 до 30 методом «ліктя»; для LOF — значення k від 5 до 50; для DBSCAN — пари (ϵ , MinPts) на основі k-NN графіка; для One-Class SVM — ν та тип ядра; для Isolation Forest — $n_estimators$ та $max_samples$; для автоенкодерів — розмірність латентного шару та кількість прихованих нейронів. Для всіх методів поріг аномальності визначався як значення, що максимізує F1-score на валідаційній вибірці.

Таблиця 4.1

Результати порівняльного аналізу на CICIDS2017

Метод	Precision	Recall	F1-score	AUC ROC
k-means (k=10)	0,74	0,68	0,71	0,79
DBSCAN ($\epsilon=0,5$)	0,81	0,72	0,76	0,83
LOF (k=20)	0,79	0,75	0,77	0,84
One-Class SVM	0,77	0,71	0,74	0,81
Isolation Forest	0,88	0,85	0,86	0,91
Автоенкодер (FC)	0,85	0,90	0,87	0,92
LSTM-автоенкодер	0,86	0,91	0,88	0,93
Ансамбль IF+LSTM	0,92	0,94	0,93	0,96

Аналіз результатів: ансамблева модель перевершує всі одиночні методи ($F1=0,93$ проти 0,88 у LSTM-AE). Приріст від ансамблювання — 5,7 п.п. K-means показує найгірші результати ($F1=0,71$), підтверджуючи теоретичні обмеження. LSTM-AE має вищий Recall (0,91), ніж Isolation Forest (0,85),

підтверджуючи перевагу для часових аномалій. DBSCAN та LOF — порівнянні результати ($F1=0,76-0,77$) при різній обчислювальній вартості на користь DBSCAN. [41]

Додаткова перевірка на наборі KDD Cup 99 підтвердила загальну картину: ансамблева модель показала $F1=0,91$ та $AUC\ ROC=0,94$, що є кращим результатом серед усіх методів. Результати на KDD Cup 99 дещо нижчі через специфіку цього набору (симульований трафік, дисбаланс класів), але загальна тенденція зберігається. Порівняння з опублікованими результатами для KDD Cup 99 показує, що ансамблевий підхід перевершує більшість раніше опублікованих методів неконтрольованого навчання на цьому наборі, що підтверджує відтворюваність і надійність отриманих результатів.

4.4. Ефективність за типами атак

Детальний аналіз ансамблевої моделі за категоріями CICIDS2017 показав різну ефективність залежно від природи атаки.

Таблиця 4.2

Ефективність ансамблевої моделі за типами атак CICIDS2017

Тип атаки	Recall	Пояснення
DDoS (LOIT)	0,98	Різке зростання total_bytes та total_packets
DoS Hulk	0,97	Аномальний SYN-ratio; unique_dst_ports=1
DoS Slowloris	0,95	Великий avg_IAT; малий total_packets
PortScan	0,94	Різке зростання unique_dst_ports
Heartbleed	0,88	Аномальні розміри TLS-пакетів
Botnet ARES	0,89	LSTM виявляє регулярні C&C патерни
Web Attack	0,82	Схожість на легітимний HTTPS-трафік
Infiltration	0,71	Мінімальний обсяг аномального трафіку

Об'ємні атаки виявляються найкраще (Recall 0,94–0,98) завдяки різким змінам статистичних ознак. Ботнетна активність (0,89) виявляється LSTM завдяки регулярним часовим патернам. Infiltration (0,71) є найскладнішою: атака генерує мінімальний трафік, схожий на легітимну фонову активність. Це відомий практичний ліміт неконтрольованого підходу — для таких атак рекомендується доповнення поведінковим аналізом на рівні абонентів.

Аналізуючи результати детальніше, можна виділити три характерні групи атак за рівнем виявлення.

До першої групи з високим Recall (0,94–0,98) відносяться DDoS, DoS Hulk та DoS Slowloris. Спільною рисою цих атак є те, що вони неминуче змінюють об'ємні або часові характеристики трафіку: різко зростає кількість пакетів, збільшується або навпаки зменшується середній міжпакетний інтервал, змінюється співвідношення SYN/ACK пакетів. Isolation Forest ефективно ізолює такі спостереження вже на перших рівнях дерева, оскільки вони суттєво відрізняються від щільних кластерів нормального трафіку.

До другої групи з середнім Recall (0,82–0,94) відносяться PortScan, Heartbleed та Botnet ARES. PortScan виявляється через різке зростання ознаки `unique_dst_ports`, що є надійним індикатором розвідувальної активності. Heartbleed проявляється у нетипових розмірах TLS-пакетів — ця атака експлуатує уразливість OpenSSL і генерує запити з аномально великим полем довжини серцебиття (`heartbeat length`). Ботнетна активність ARES виявляється переважно LSTM-компонентою ансамблю: регулярні з'єднання з C&C-сервером з фіксованим інтервалом формують стійкий часовий патерн, що не характерний для легітимного трафіку і добре розпізнається рекурентною мережею.

До третьої групи з нижчим Recall (0,71–0,82) відносяться Web Attack та Infiltration. Web-атаки (XSS, SQL Injection) відбуваються поверх легітимних HTTPS-з'єднань і не змінюють flow-рівневих ознак — розміри пакетів, тривалість та інтенсивність з'єднань залишаються в межах норми. Виявлення таких атак потребує або аналізу вмісту запитів (що неможливо при TLS), або поведінкового профілювання на рівні застосунків.

Отримані результати підтверджують, що ансамблева модель є найбільш ефективною для атак, що суттєво впливають на мережеві метрики. Для атак, що діють на рівні застосунків або генерують незначний трафік, неконтрольований підхід на базі flow-ознак має об'єктивні обмеження. Це не є недоліком конкретної реалізації, а фундаментальною властивістю flow-based аналізу: він оперує агрегованою статистикою з'єднань і не має доступу до контексту прикладного рівня. Для комплексного захисту операторської мережі рекомендується поєднувати розроблену систему з інструментами аналізу поведінки користувачів (UEBA) та журналами прикладних систем.

4.5. Адаптивний поріг та аналіз хибних спрацьовувань

Хибні спрацьовування — серйозна практична проблема: вони збільшують навантаження на SOC і при надмірній кількості призводять до «втоми від тривог». Аналіз показав три основні джерела FP: природні піки трафіку (~45%); нові легітимні сервіси або зміни поведінки абонентів (~30%); технічні роботи в мережі (~25%).

Реалізований адаптивний механізм: система підтримує 24 погодинних порогових значення. Щодня о 00:00 пороги оновлюються на основі 99-го перцентилля розподілу оцінок за відповідну годину за попередні 7 діб. Вночі поріг нижчий (менший трафік), вдень — вищий. Результат на CICIDS2017: FPR знизився з 12,4% (фіксований поріг) до 3,1% (адаптивний) при незначному зменшенні Recall з 0,95 до 0,94. У великій мережі з 10 млн записів/год це скорочення хибних тривог у 4 рази.

4.6. Продуктивність та вимоги до ресурсів

Тестування продуктивності на стенді з трьох серверів (колектор 4 ядра/16 ГБ, Spark-сервер 8 ядер/32 ГБ, сервер детекції 8 ядер/32 ГБ/GPU RTX 3060). Результати: пропускна здатність колектора — до 800 000 записів/хв; Spark Streaming — 600 000 векторів/хв; Isolation Forest — 7 500 000+ класифікацій/хв (лінійно масштабується); LSTM-AE — 30 000 послідовностей/хв (обмежена GPU); загальна затримка — менше 45 секунд. Система забезпечує аналіз трафіку у реальному часі для операторів середнього масштабу (1–10 Гбіт/с).

4.7. Практичні рекомендації щодо впровадження

На основі отриманих результатів сформульовано рекомендації для операторів трьох масштабів.

Малі оператори (до 1 Гбіт/с): Isolation Forest як єдиний детектор ($F1=0,86$) на одному сервері (8 ядер, 32 ГБ, без GPU). Мінімальна вартість впровадження при прийнятній якості виявлення.

Середні оператори (1–10 Гбіт/с): повна ансамблева конфігурація ($F1=0,93$) з Apache Kafka, адаптивним порогом та інтеграцією з SIEM. Кластер 3–5 серверів з GPU для LSTM. Поетапне впровадження: Shadow mode (1–2 тижні) → Pilot mode (2–4 тижні) → Production mode.

Великі оператори (понад 10 Гбіт/с): повна мікросервісна архітектура в Kubernetes, розподілений Kafka-кластер, автоматичне BGP Blackholing для підтверджених атак, щотижневе інкрементальне перенавчання моделей.

– поріг аномальності слід налаштовувати індивідуально: цільовий $FPR \leq 5\%$;

- автоматична кластеризація алертів (DBSCAN на векторах ознак алертів) групує тисячі тривог у кілька значущих інцидентів, знижуючи навантаження на SOC;

- щоквартальна оцінка якості на підтверджених інцидентах необхідна для своєчасного виявлення деградації моделей (concept drift).

Висновок

У четвертому розділі розроблено п'ятирівневу мікросервісну архітектуру CBA з Apache Kafka та Spark Streaming. Реалізовано і протестовано сім методів на CICIDS2017 та KDD Cup 99. Ансамблева модель IF+LSTM досягає $F1=0,93$ та $AUC\ ROC=0,96$. Адаптивний погодинний поріг знижує FPR з 12,4% до 3,1%. Продуктивність системи — понад 100 000 класифікацій/с з затримкою менше 45 секунд. Сформульовано рекомендації для операторів трьох масштабів із поетапним впровадженням.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

У дипломній роботі вирішена науково-практична задача дослідження та застосування методів неконтрольованого навчання для виявлення аномалій і мережових атак в операторському трафіку. За результатами виконаної роботи зроблено такі висновки:

1. Проведено аналіз стану проблеми. Встановлено, що зростаюча частка шифрованого трафіку (понад 85%), масштаб операторських мереж і постійне ускладнення кіберзагроз роблять традиційні сигнатурні методи недостатніми. Методи неконтрольованого навчання є найбільш перспективними для виявлення нових атак без розмітки даних. Найбільшу небезпеку становлять DDoS-атаки, ботнетна активність, DNS-тунелювання та ексфільтрація даних.

2. Систематизовано та досліджено сім методів неконтрольованого навчання: статистичне профілювання (3σ), LOF, One-Class SVM, k-means, DBSCAN, Isolation Forest та LSTM-автоенкодер. Для кожного визначено принцип роботи, клас аномалій, для якого він найефективніший, та обмеження. Isolation Forest вирізняється найкращою масштабованістю (лінійна складність, 100 000+ класифікацій/с); LSTM-автоенкодер — найефективніший для часових аномалій (ботнет, повільне сканування).

3. Сформовано вектор із 22 ознак мережових потоків чотирьох груп: статистичні (8), поведінкові (6), протокольні (5), часові (3). Включення часових ознак (hour_of_day, day_of_week, traffic_zscore) є критичним для виявлення контекстуальних аномалій. Розроблено повний pipeline попередньої обробки: медіанна імпутація → логарифмічна трансформація → z-score нормалізація → PCA (15 компонент, 95,3% дисперсії). Хронологічний принцип формування вибірок виключає data leakage.

4. Реалізовано та протестовано сім методів на CICIDS2017 та KDD Cup 99. Ансамблева модель (Isolation Forest + LSTM-автоенкодер, $\alpha=0,60$, $\tau=0,65$) досягає F1-score=0,93 та AUC ROC=0,96 — найкращий результат серед усіх. Isolation Forest окремо: F1=0,86; LSTM-AE: F1=0,88; k-means — найгірший

результат ($F1=0,71$). Приріст від ансамблювання становить 5,7 п.п. Найвищий Recall для DDoS (0,98), найнижчий — для Infiltration (0,71).

5. Розроблено п'ятирівневу мікросервісну архітектуру СВА (NetFlow-збір → Spark Streaming → детекція → ансамблювання → реагування) на базі Docker/Kubernetes та Apache Kafka. Система забезпечує продуктивність понад 100 000 класифікацій/с та затримку менше 45 секунд. Адаптивний погодинний механізм порогу знижує FPR з 12,4% до 3,1% при незначному зменшенні Recall.

6. Сформульовано диференційовані практичні рекомендації: малі оператори (до 1 Гбіт/с) — Isolation Forest на одному сервері; середні (1–10 Гбіт/с) — повна ансамблева конфігурація з поетапним впровадженням (Shadow → Pilot → Production); великі (понад 10 Гбіт/с) — повна мікросервісна архітектура з автоматичним BGP Blackholing. Цільовий $FPR \leq 5\%$; щотижневе інкрементальне перенавчання; щоквартальна оцінка якості на підтверджених інцидентах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Buczak A. L., Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials. 2016. Vol. 18, No. 2. P. 1153–1176. DOI: 10.1109/COMST.2015.2494502.
2. ENISA Threat Landscape 2024 [Електронний ресурс]. European Union Agency for Cybersecurity, 2024. – Режим доступу: <https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024>.
3. Sperotto A., Schaffrath G., Sadre R., Morariu C., Pras A., Stiller B. An overview of IP flow-based intrusion detection. IEEE Communications Surveys & Tutorials. 2010. Vol. 12, No. 3. P. 343–356. DOI: 10.1109/SURV.2010.032210.00054.
4. Claise B. Cisco Systems NetFlow Services Export Version 9. RFC 3954 [Електронний ресурс]. IETF, 2004. – Режим доступу: <https://www.rfc-editor.org/rfc/rfc3954>.
5. Sandvine. Global Internet Phenomena Report 2024 [Електронний ресурс]. Sandvine, 2024. – Режим доступу: <https://www.sandvine.com/global-internet-phenomena-report>.
6. Hawkins D. Identification of Outliers. London: Chapman and Hall, 1980. 188 p. DOI: 10.1007/978-94-015-3994-4.
7. Garcia-Teodoro P., Diaz-Verdejo J., Macia-Fernandez G., Vazquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security. 2009. Vol. 28, No. 1–2. P. 18–28. DOI: 10.1016/j.cose.2008.08.003.
8. Liao H.-J., Lin C.-H. R., Lin Y.-C., Tung K.-Y. Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications. 2013. Vol. 36, No. 1. P. 16–24. DOI: 10.1016/j.jnca.2012.09.004.
9. Roesch M. Snort: Lightweight Intrusion Detection for Networks. Proceedings of the 13th USENIX Conference on System Administration (LISA'99). Seattle, 1999. P. 229–238.

10. Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey. *ACM Computing Surveys*. 2009. Vol. 41, No. 3. P. 1–58. DOI: 10.1145/1541880.1541882.
11. Modi C., Patel D., Borisaniya B., Patel H., Patel A., Rajarajan M. A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications*. 2013. Vol. 36, No. 1. P. 42–57. DOI: 10.1016/j.jnca.2012.05.003.
12. Fernandes G., Rodrigues J. J. P. C., Carvalho L. F., Al-Muhtadi J. F., Proença M. L. A comprehensive survey on network anomaly detection. *Telecommunication Systems*. 2019. Vol. 70, No. 3. P. 447–489. DOI: 10.1007/s11235-018-0475-8.
13. Tavallaei M., Bagheri E., Lu W., Ghorbani A. A. A detailed analysis of the KDD CUP 99 data set. *Proceedings of IEEE CISDA 2009*. Ottawa, 2009. P. 1–6. DOI: 10.1109/CISDA.2009.5356528.
14. Sharafaldin I., Habibi Lashkari A., Ghorbani A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *Proceedings of ICISSP 2018*. Funchal, 2018. P. 108–116. DOI: 10.5220/0006639801080116.
15. Aggarwal C. C. *Outlier Analysis*. 2nd ed. New York: Springer, 2017. 512 p. DOI: 10.1007/978-3-319-47578-3.
16. Goldstein M., Uchida S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLOS ONE*. 2016. Vol. 11, No. 4. e0152173. DOI: 10.1371/journal.pone.0152173.
17. Grubbs F. E. Procedures for detecting outlying observations in samples. *Technometrics*. 1969. Vol. 11, No. 1. P. 1–21. DOI: 10.1080/00401706.1969.10490657.
18. Brutlag J. D. Aberrant behavior detection in time series for network monitoring. *Proceedings of USENIX LISA 2000*. New Orleans, 2000. P. 139–146.
19. Breunig M. M., Kriegel H.-P., Ng R. T., Sander J. LOF: Identifying density-based local outliers. *ACM SIGMOD Record*. 2000. Vol. 29, No. 2. P. 93–104. DOI: 10.1145/335191.335388.

20. Perdisci R., Giacinto G., Roli F. Alarm clustering for intrusion detection systems in computer networks. *Engineering Applications of AI*. 2006. Vol. 19, No. 4. P. 429–438. DOI: 10.1016/j.engappai.2006.01.003.
21. Schölkopf B., Platt J. C., Shawe-Taylor J., Smola A. J., Williamson R. C. Estimating the support of a high-dimensional distribution. *Neural Computation*. 2001. Vol. 13, No. 7. P. 1443–1471. DOI: 10.1162/089976601750264965.
22. Tax D. M. J., Duin R. P. W. Support vector data description. *Machine Learning*. 2004. Vol. 54, No. 1. P. 45–66. DOI: 10.1023/B:MACH.00000008084.60811.49.
23. MacQueen J. Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, 1967. Vol. 1. P. 281–297.
24. Lakhina A., Crovella M., Diot C. Diagnosing network-wide traffic anomalies. *Proceedings of ACM SIGCOMM 2004*. Portland, 2004. P. 219–230. DOI: 10.1145/1015467.1015492.
25. Ester M., Kriegel H.-P., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of KDD-96*. Portland, 1996. P. 226–231.
26. Schubert E., Sander J., Ester M., Kriegel H.-P., Xu X. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM TODS*. 2017. Vol. 42, No. 3. P. 1–21. DOI: 10.1145/3068335.
27. Liu F. T., Ting K. M., Zhou Z.-H. Isolation forest. *Proceedings of IEEE ICDM 2008*. Pisa, 2008. P. 413–422. DOI: 10.1109/ICDM.2008.17.
28. Liu F. T., Ting K. M., Zhou Z.-H. Isolation-based anomaly detection. *ACM TKDD*. 2012. Vol. 6, No. 1. P. 1–39. DOI: 10.1145/2133360.2133363.
29. Hariri S., Kind M. C., Brunner R. J. Extended Isolation Forest. *IEEE TKDE*. 2021. Vol. 33, No. 4. P. 1479–1489. DOI: 10.1109/TKDE.2019.2947676.
30. Sommer R., Paxson V. Outside the closed world: On using machine learning for network intrusion detection. *Proceedings of IEEE S&P 2010*. Oakland, 2010. P. 305–316. DOI: 10.1109/SP.2010.25.

31. Hinton G. E., Salakhutdinov R. R. Reducing the dimensionality of data with neural networks. *Science*. 2006. Vol. 313, No. 5786. P. 504–507. DOI: 10.1126/science.1127647.
32. Vincent P., Larochelle H., Lajoie I., Bengio Y., Manzagol P.-A. Stacked denoising autoencoders. *JMLR*. 2010. Vol. 11. P. 3371–3408.
33. Pedregosa F. et al. Scikit-learn: Machine learning in Python. *JMLR*. 2011. Vol. 12. P. 2825–2830.
34. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Computation*. 1997. Vol. 9, No. 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
35. Malhotra P., Ramakrishnan A., Anand G., Vig L., Agarwal P., Shroff G. LSTM-based encoder-decoder for multi-sensor anomaly detection [Электронный ресурс]. arXiv:1607.00148, 2016. – Режим доступа: <https://arxiv.org/abs/1607.00148>.
36. Nguyen T. T. T., Armitage G. A survey of techniques for internet traffic classification using deep packet inspection. *IEEE Comms. Surveys & Tutorials*. 2008. Vol. 10, No. 4. P. 56–76. DOI: 10.1109/SURV.2008.080406.
37. Claise B., Trammell B., Aitken P. Specification of the IPFIX Protocol. RFC 7011 [Электронный ресурс]. IETF, 2013. – Режим доступа: <https://www.rfc-editor.org/rfc/rfc7011>.
38. Zseby T., Molina M., Duffield N., Niccolini S., Raspall F. Sampling and Filtering Techniques for IP Packet Selection. RFC 5475 [Электронный ресурс]. IETF, 2009. – Режим доступа: <https://www.rfc-editor.org/rfc/rfc5475>.
39. Deri L. ntopng: Network Traffic Monitoring Software [Электронный ресурс]. ntop, 2024. – Режим доступа: <https://www.ntop.org/products/traffic-analysis/ntop/>.
40. Meng W., Li W., Kwok L. F. Design of intelligent KNN-based alarm filter using knowledge-based alert verification in intrusion detection. *Security and Communication Networks*. 2015. Vol. 8, No. 18. P. 3855–3867. DOI: 10.1002/sec.1300.

41. Sharafaldin I., Habibi Lashkari A., Ghorbani A. A. Intrusion detection evaluation dataset (CIC-IDS2017) [Электронный ресурс]. Canadian Institute for Cybersecurity, 2017. – Режим доступа: <https://www.unb.ca/cic/datasets/ids-2017.html>.
42. Pearson K. On lines and planes of closest fit to systems of points in space. Philosophical Magazine. 1901. Vol. 2, No. 11. P. 559–572. DOI: 10.1080/14786440109462720.
43. Architecture of DDoS Attack [Электронный ресурс]. ResearchGate. – Режим доступа до ресурсу: https://www.researchgate.net/figure/Architecture-of-DDoS-Attack_fig1_377606049.
44. Network flow data analysis framework architecture [Электронный ресурс]. ResearchGate. – Режим доступа до ресурсу: https://www.researchgate.net/figure/Network-flow-data-analysis-framework-architecture_fig1_304802412.
45. Anomaly-based IDS architecture [Электронный ресурс]. ResearchGate. – Режим доступа до ресурсу: https://www.researchgate.net/figure/Anomaly-based-IDS-architecture-13_fig1_356759205.