МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

СВИРИДОВ Владислав Олександрович

Алгоритми виявлення аномального трафіку Інтернет-речей / Algorithms for detecting anomalous Internet of Things traffic

спеціальність: 125 – Кібербезпека та захист інформації

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -21
В.О. Свиридов

Науковий керівник
к.т.н., доцент Н.Г. Яцків

Кваліфікаційну роботу
Допущено до захисту:

«____» _____ ___2024 р.

Завідувач кафедри
_____ **В.В.Яцків**

ТЕРНОПІЛЬ - 2024

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 – Кібербезпека та захист інформації
освітньо-професійна програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ В.В.Яцків
«____» _____ 2023 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
СВИРИДОВ Владислав Олександрович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Алгоритми виявлення аномального трафіку Інтернет-речей / Algorithms for Detecting Anomalous Internet of Things Traffic
керівник роботи к.т.н., доцент Н.Г. Яцків
затверджені наказом по університету від 12 грудня 2023 року № 753

2. Строк подання студентом закінченої кваліфікаційної роботи 12 грудня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- дослідити характеристики та особливості трафіку IoT-пристроїв;
- проаналізувати існуючі методи виявлення аномалій в мережевому трафіку;
- розробити алгоритми виявлення аномалій для IoT-трафіку;
- реалізувати систему моніторингу мережевого трафіку на Python;
- розробити рекомендації щодо застосування системи моніторингу.

5. Перелік графічного матеріалу у роботі:
- основні типи аномалій в IoT-трафіку;
- типові аномалії протоколів передачі даних;
- алгоритм виявлення аномалій;
- робота програми з захоплення пакетів;
- статистика мережевого трафіку;
- порівняння методів виявлення аномалій.

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Теоретичні основи аналізу мережевого трафіку IoT	12.2023 р. – 03.2024 р.	
2	Алгоритми виявлення аномалій	03.2024 р. – 06.2024 р.	
3	Розробка системи моніторингу IoT-трафіку на python	06.2024 р. – 11.2024 р.	

Студент _____ В.О.Свиридов
(підпис)

Керівник роботи _____ к.т.н., доцент Н.Г. Яцків
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Алгоритми виявлення аномального трафіку Інтернет-речей» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека та захист інформації» освітньо-професійної програми «Кібербезпека» написана обсягом 75 сторінок і містить 26 ілюстрації, 34 таблиць, 3 додатки та 48 джерел за переліком посилань.

Метою кваліфікаційної роботи є розробка та реалізація алгоритмів виявлення аномального трафіку в мережах Інтернету речей, що враховують специфіку роботи IoT-пристроїв та забезпечують ефективний моніторинг мережевої активності.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: порівняльний аналіз, теоретичний аналіз, алгоритмічний синтез, експериментальні методи.

Результати дослідження: Результати даного дослідження сприяють розвитку методів та алгоритмів виявлення аномального трафіку в мережах Інтернету речей та підвищенню безпеки IoT-систем.

Розроблено системи моніторингу IoT-трафіку з реалізованими алгоритмами виявлення аномалій. Розроблене програмне забезпечення може бути безпосередньо використане для захисту IoT-мереж та виявлення потенційних загроз безпеці.

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, МЕРЕЖЕВИЙ ТРАФІК, ВИЯВЛЕННЯ АНОМАЛІЙ, СИСТЕМА МОНІТОРИНГУ.

ABSTRACT

Qualification thesis on the topic "Algorithms for Detecting Anomalous Internet of Things Traffic" for obtaining the educational degree "Master" in the specialty 125 "Cybersecurity and Information Protection" under the educational-professional program "Cybersecurity" comprises 75 pages and includes 26 illustrations, 34 tables, 3 appendices, and 48 references in the bibliography.

Objective of the qualification thesis: The development and implementation of algorithms for detecting anomalous traffic in Internet of Things (IoT) networks, taking into account the specifics of IoT device operations and ensuring effective monitoring of network activity.

Research methods: To achieve the objectives set in this qualification thesis, the following methods were used: comparative analysis, theoretical analysis, algorithmic synthesis, and experimental methods.

Research results: The results of this research contribute to the advancement of methods and algorithms for detecting anomalous traffic in IoT networks and enhancing the security of IoT systems.

Monitoring systems for IoT traffic with implemented anomaly detection algorithms have been developed. The developed software can be directly used to protect IoT networks and detect potential security threats.

Keywords: INTERNET OF THINGS, NETWORK TRAFFIC, ANOMALY DETECTION, MONITORING SYSTEM.

ЗМІСТ

СПИСОК ВИКОРИСТАНИХ СКОРОЧЕНЬ 7
ВСТУП 8
1. ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ IOT 10
1.1 Характеристики та особливості трафіку пристроїв Інтернету речей 10
1.2 Типові аномалії в IoT-трафіку та їх класифікація 16
1.3 Методи та метрики оцінки мережевої активності IoT-пристроїв 20
1.4 Огляд існуючих систем виявлення аномалій в IoT-мережах 26
2. АЛГОРИТМИ ВИЯВЛЕННЯ АНОМАЛІЙ 32
2.1 Алгоритми на основі порогових значень 32
2.2 Алгоритми виявлення викидів 37
2.3 Алгоритми глибокого навчання для аналізу аномалій 43
2.4 Гібридні алгоритми виявлення аномального трафіку 49
3. РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ IOT-ТРАФІКУ НА PYTHON 56
3.1 Проектування архітектури додатку та вибір інструментів розробки 56
3.2 Реалізація захоплення та аналізу мережевого трафіку 61
3.3 Впровадження алгоритмів виявлення аномалій 69
ВИСНОВКИ 75
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 76
ДОДАТОК А. Код програми аналізу пакетів 81
ДОДАТОК Б. Код програми моніторингу трафіку 83
ДОДАТОК В. Копії публікацій 86

СПИСОК ВИКОРИСТАНИХ СКОРОЧЕНЬ

API — Application Programming Interface

CPU — Central Processing Unit

DDoS — Distributed Denial of Service

DNS — Domain Name System

GPU — Graphics Processing Unit

HTTP — HyperText Transfer Protocol

ICMP — Internet Control Message Protocol

IDS — Intrusion Detection System

IoT — Internet of Things

IPS — Intrusion Prevention System

IP — Internet Protocol

JSON — JavaScript Object Notation

MQTT — Message Queuing Telemetry Transport

QoS — Quality of Service

ROC — Receiver Operating Characteristic

SNMP — Simple Network Management Protocol

TCP — Transmission Control Protocol

TPU — Tensor Processing Unit

UDP — User Datagram Protocol

ВСТУП

Актуальність теми. Актуальність теми дослідження зумовлена стрімким зростанням кількості IoT-пристроїв та необхідністю забезпечення їх безпечної роботи в мережі. Виявлення аномального трафіку в мережах Інтернету речей стає критично важливим завданням, оскільки такі пристрої часто стають мішенню для кібератак та можуть бути використані для проведення розподілених атак. Традиційні методи виявлення аномалій не завжди ефективні для IoT-пристроїв через їх обмежені ресурси та специфічні паттерни мережевої активності. Розробка спеціалізованих алгоритмів виявлення аномалій, які враховують особливості IoT-трафіку, є перспективним напрямком досліджень у сфері мережевої безпеки.

Зростаюча кількість кіберінцидентів, пов'язаних з IoT-пристроями, та потенційні ризики для критичної інфраструктури підкреслюють важливість розробки ефективних систем моніторингу та виявлення аномалій. Дослідження в цій галузі спрямовані на створення алгоритмів, здатних працювати в режимі реального часу, адаптуватися до змін у характері трафіку та ефективно виявляти різні типи аномальної поведінки з мінімальною кількістю хибних спрацювань.

Мета і завдання дослідження. Метою дослідження є розробка та реалізація алгоритмів виявлення аномального трафіку в мережах Інтернету речей, що враховують специфіку роботи IoT-пристроїв та забезпечують ефективний моніторинг мережевої активності.

Зазначена мета передбачає вирішення таких завдань:

- дослідити характеристики та особливості трафіку IoT-пристроїв;
- проаналізувати існуючі методи виявлення аномалій в мережевому трафіку;
- розробити алгоритми виявлення аномалій для IoT-трафіку;
- реалізувати систему моніторингу мережевого трафіку на Python;
- розробити рекомендації щодо застосування системи моніторингу.

Об'єктом дослідження є процеси моніторингу та аналізу мережевого трафіку в середовищі Інтернету речей.

Предметом дослідження є алгоритми виявлення аномального трафіку в мережах IoT та методи їх реалізації.

Методи дослідження: статистичний аналіз, машинне навчання, моделювання, експериментальні дослідження.

Наукова новизна: Наукова новизна дослідження полягає у розробці нових алгоритмів виявлення аномалій, які враховують специфіку IoT-трафіку та забезпечують високу точність виявлення при мінімальних вимогах до обчислювальних ресурсів. Запропоновано оригінальні підходи до аналізу мережевої активності та виявлення аномальної поведінки IoT-пристроїв.

Практичне значення: Практичне значення отриманих результатів полягає у створенні працюючої системи моніторингу IoT-трафіку з реалізованими алгоритмами виявлення аномалій. Розроблене програмне забезпечення може бути безпосередньо використане для захисту IoT-мереж та виявлення потенційних загроз безпеці.

Результати дослідження: Результати даного дослідження сприяють розвитку методів та алгоритмів виявлення аномального трафіку в мережах Інтернету речей та підвищенню безпеки IoT-систем.

Публікації та апробація до кваліцікаційної роботи.

1. Свиридов В. Методи класифікації аномального трафіку в мережах Інтернету речей. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2022), Тернопіль, 2024. – С. 63-67.

2. Свиридов В. Машинне навчання у виявленні аномалій в мережах Інтернету речей. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2024. – С. 51-54.

1 ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ IOT

1.1. Характеристики та особливості трафіку пристроїв Інтернету речей

Розвиток технологій Інтернету речей створив нові виклики для мережевої інфраструктури та аналізу трафіку. Мережевий трафік IoT-пристроїв характеризується специфічними параметрами та патернами поведінки, які суттєво відрізняються від традиційного трафіку комп'ютерних мереж. Пристрої IoT генерують періодичні потоки даних невеликого розміру, що передаються за допомогою різних протоколів, включаючи MQTT, CoAP та HTTP. Основні характеристики трафіку IoT-пристроїв включають розмір пакетів, частоту передачі, тип протоколу та шаблони взаємодії з мережевими вузлами. Детальний аналіз цих параметрів дозволяє ефективно керувати мережевою інфраструктурою та виявляти потенційні проблеми безпеки. В таблиці 1.1 зображено основні характеристики трафіку IoT-пристроїв [1].

Таблиця 1.1 – Характеристики протоколів передачі даних IoT-пристроїв

Протокол	Розмір заголовка	Службовий трафік	Шифрування	Підтримка QoS
MQTT	2-4 байти	8-12%	TLS/SSL	3 рівні
CoAP	4-8 байт	5-10%	DTLS	2 рівні
HTTP	20-30 байт	15-25%	TLS	Базова
AMQP	8-12 байт	10-15%	TLS/SSL	3 рівні
WebSocket	10-14 байт	12-18%	TLS	Розширена

Протоколи передачі даних в мережах IoT мають свої специфічні особливості та обмеження. MQTT протокол, який широко використовується в IoT, базується на моделі публікації/підписки та оптимізований для передачі даних з обмеженими ресурсами. Протокол CoAP розроблений спеціально для обмежених пристроїв та забезпечує надійну передачу даних з мінімальними накладними витратами. HTTP залишається популярним вибором для IoT-пристроїв з достатніми обчислювальними ресурсами. Кожен з цих протоколів

має власні характеристики трафіку, включаючи розмір заголовків, формат корисного навантаження та механізми забезпечення якості обслуговування [2]. На рисунку 1.1 зображено порівняння протоколів передачі даних в IoT.

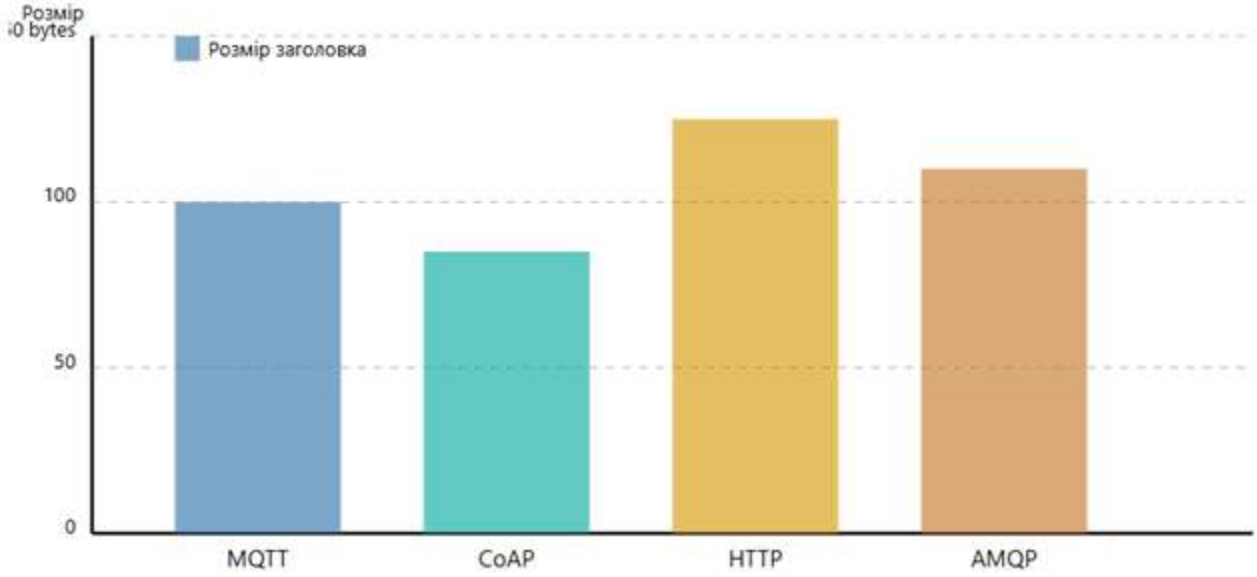

Рисунок 1.1 – Порівняння протоколів передачі даних в IoT

Мережевий трафік IoT-пристроїв демонструє чіткі часові патерни, пов'язані з періодичністю збору та передачі даних. Сенсори та актуатори генерують дані з фіксованими інтервалами, які можуть варіюватися від кількох секунд до кількох годин залежно від конкретного застосування. В таблиці 1.2 зприведено типові часові патерни трафіку IoT-пристроїв.

Таблиця 1.2 –Характеристики IoT-трафіку за галузями застосування

Галузь	Періодичність опитування	Максимальна затримка	Пікове навантаження
Промисловість	100 мс - 1 с	10 мс	1000 пакетів/с
Медицина	1 с - 5 с	50 мс	500 пакетів/с
Розумний дім	5 с - 30 с	200 мс	100 пакетів/с
Сільське господарство	30 с - 5 хв	1000 мс	50 пакетів/с
Транспорт	1 с - 10 с	100 мс	300 пакетів/с

Періодичність трафіку впливає на загальне навантаження мережі та вимагає відповідного планування мережевих ресурсів. Аналіз часових патернів дозволяє виявляти аномалії та потенційні збої в роботі пристроїв [3].

Розмір пакетів даних у мережах IoT зазвичай значно менший порівняно з традиційним мережевим трафіком. Це пов'язано з обмеженими обчислювальними ресурсами та енергоефективністю пристроїв. Більшість IoT-пристроїв передають пакети розміром від 50 до 200 байт, що містять показники сенсорів або команди керування. При цьому заголовки протоколів можуть становити значну частину загального розміру пакета. Аналіз розміру пакетів дозволяє оптимізувати мережеву інфраструктуру та виявляти потенційні проблеми з фрагментацією даних [4]. В таблиці 1.3 приведено розподіл розмірів пакетів для різних типів IoT-пристроїв.

Таблиця 1.3 – Показники мережевого навантаження IoT-пристроїв

Тип пристрою	Середній розмір пакету	Добовий об'єм даних	Пропускна здатність
Відеокамера	1280 байт	15-20 ГБ	2-5 Мбіт/с
Датчик температури	64 байт	2-5 МБ	1-2 Кбіт/с
GPS-трекер	128 байт	50-100 МБ	10-20 Кбіт/с
Розумний лічильник	96 байт	10-15 МБ	5-8 Кбіт/с
Промисловий контролер	256 байт	500-800 МБ	50-100 Кбіт/с

Мережева топологія значно впливає на характеристики трафіку IoT-пристроїв. Більшість IoT-мереж використовують зіркоподібну або mesh-топологію, де кінцеві пристрої взаємодіють через шлюзи або координатори мережі. Шлюзи агрегують трафік від множини пристроїв та можуть виконувати попередню обробку даних перед їх передачею в хмарну інфраструктуру. Патерни взаємодії між пристроями та шлюзами створюють характерні потоки трафіку, які необхідно враховувати при проектуванні мережевої інфраструктури. Аналіз топології мережі дозволяє оптимізувати

маршрутизацію та балансування навантаження [5]. На рисунку 1.2 зображено типові топології IoT-мереж та відповідні потоки трафіку.

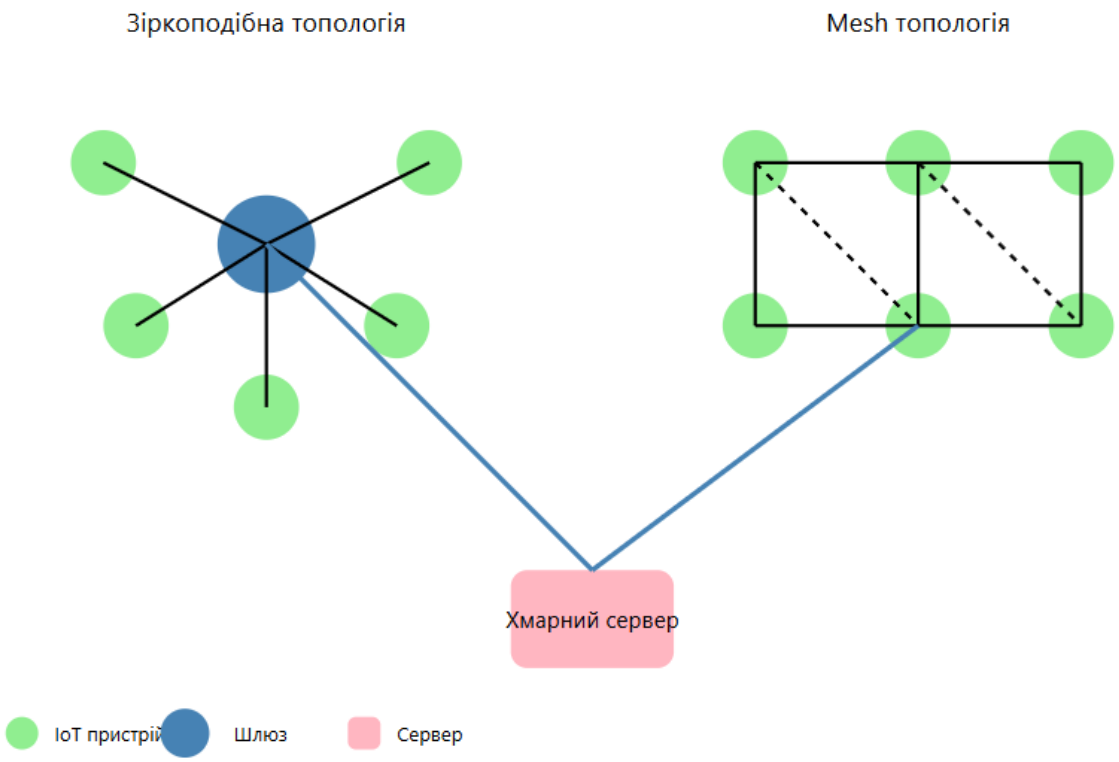

Рисунок 1.2 – Типові топології IoT-мереж та потоки трафіку

Затримки передачі даних в мережах IoT мають критичне значення для багатьох застосувань. Промислові IoT-системи часто вимагають передачі даних у реальному часі з гарантованими затримками менше 10 мілісекунд. Медичні IoT-пристрої потребують надійної передачі даних з мінімальними затримками для моніторингу стану пацієнтів. Побутові IoT-пристрої зазвичай мають менш строгі вимоги до затримок. Різні класи QoS (Quality of Service) використовуються для забезпечення необхідного рівня обслуговування [6]. Мережеві протоколи IoT підтримують механізми пріоритезації трафіку та управління затримками. Моніторинг затримок дозволяє виявляти проблеми з продуктивністю мережі та оптимізувати конфігурацію пристроїв [7]. В таблиці 1.1 зображено вимоги до затримок для різних типів IoT-застосувань.

Передача даних в мережах IoT часто характеризується асиметричністю трафіку. Сенсорні пристрої генерують більше вихідного трафіку, передаючи дані вимірювань, тоді як вхідний трафік обмежується командами керування та конфігурації. Актуатори, навпаки, отримують більше вхідного трафіку з командами керування [8]. Співвідношення вхідного та вихідного трафіку може варіюватися від 1:100 до 100:1 залежно від типу пристрою та його призначення. Асиметрія трафіку впливає на вибір мережевих технологій та конфігурацію каналів передачі даних [9]. В таблиці 1.2 зображено типові співвідношення вхідного та вихідного трафіку для різних класів IoT-пристроїв.

Агрегація та фрагментація даних відіграють важливу роль в формуванні трафіку IoT-мереж. Шлюзи та проміжні вузли можуть агрегувати дані від множини пристроїв для оптимізації використання мережевих ресурсів. Фрагментація пакетів виникає при передачі через мережі з різними максимальними розмірами пакетів (MTU). Протоколи IoT реалізують спеціальні механізми для ефективної обробки фрагментованих даних. Аналіз процесів агрегації та фрагментації дозволяє оптимізувати конфігурацію мережі та виявляти потенційні проблеми з передачею даних. Моніторинг цих процесів необхідний для забезпечення надійної роботи IoT-систем [10]. В таблиці 1.3 зображено вплив агрегації та фрагментації на характеристики трафіку.

Шифрування та захист даних значно впливають на характеристики трафіку IoT-пристроїв. Використання криптографічних протоколів, таких як TLS/DTLS, збільшує розмір пакетів та створює додаткове навантаження на мережу [11]. Процеси встановлення захищених з'єднань генерують специфічні патерни трафіку, пов'язані з обміном ключами та сертифікатами. Різні рівні безпеки та методи шифрування призводять до різних характеристик трафіку. Моніторинг зашифрованого трафіку вимагає спеціальних підходів, оскільки традиційні методи глибокого аналізу пакетів стають неефективними. Впровадження механізмів безпеки повинно враховувати обмеження IoT-пристроїв та вимоги до продуктивності мережі [12].

Енергоспоживання IoT-пристроїв тісно пов'язане з характеристиками мережевого трафіку. Пристрої з батарейним живленням використовують різні

стратегії економії енергії, включаючи періодичне вимкнення радіомодулів та агрегацію даних перед передачею. Режими сну та пробудження створюють характерні патерни трафіку з періодами повної відсутності активності. Балансування між енергоефективністю та своєчасністю доставки даних вимагає оптимізації параметрів передачі. Протоколи IoT підтримують спеціальні механізми для роботи з пристроями, що мають обмежене енергоспоживання. Аналіз трафіку дозволяє оцінювати ефективність енергозберігаючих стратегій та прогнозувати термін служби батарей [13].

Масштабованість мереж IoT створює специфічні вимоги до характеристик трафіку. Збільшення кількості пристроїв призводить до зростання загального обсягу трафіку та ускладнення патернів взаємодії. Мережева інфраструктура повинна забезпечувати ефективну обробку трафіку від тисяч або мільйонів пристроїв. Механізми виявлення пристроїв та управління конфігурацією генерують додатковий службовий трафік. Протоколи IoT реалізують спеціальні механізми для роботи в масштабних мережах, включаючи ієрархічну маршрутизацію та розподілене управління. Моніторинг трафіку в масштабних IoT-мережах вимагає спеціалізованих інструментів та методів аналізу [14].

Якість обслуговування в мережах IoT залежить від множини факторів, включаючи характеристики трафіку та можливості мережевої інфраструктури. Різні класи пристроїв та застосувань мають різні вимоги до QoS, включаючи затримки, джиттер та надійність доставки даних. Механізми забезпечення QoS реалізуються на різних рівнях мережевого стека, від канального до прикладного. Протоколи IoT підтримують пріоритезацію трафіку та управління чергами для забезпечення необхідного рівня обслуговування. Моніторинг параметрів QoS дозволяє виявляти проблеми з якістю обслуговування та оптимізувати конфігурацію мережі [15].

1.2 Типові аномалії в IoT-трафіку та їх класифікація

Мережевий трафік пристроїв Інтернету речей має специфічні характеристики та патерни поведінки, відхилення від яких можуть вказувати на наявність аномалій. Дослідження показують, що більшість IoT-пристроїв генерують передбачуваний періодичний трафік з відносно стабільними параметрами. Порушення цих закономірностей може свідчити про технічні несправності, спроби несанкціонованого доступу або кібератаки. Аналіз аномалій в IoT-трафіку вимагає розуміння нормальної поведінки пристроїв та типових відхилень від неї. Класифікація аномалій допомагає визначити характер проблеми та вибрати відповідні методи реагування. Моніторинг мережевого трафіку дозволяє виявляти аномалії на ранніх стадіях та запобігати потенційним інцидентам безпеки. Аномалії можуть проявлятися у зміні обсягів трафіку, частоти передачі даних, розміру пакетів та інших параметрів мережевої активності [16]. В таблиці 1.4 зображено основні типи аномалій в IoT-трафіку.

Таблиця 1.4 – Основні типи аномалій в IoT-трафіку

Тип аномалії	Ознаки	Можливі причини	Рівень загрози
Об'ємні аномалії	Різке збільшення трафіку	DDoS-атака, зараження	Високий
Частотні аномалії	Зміна періодичності	Сканування, збої	Середній
Протокольні аномалії	Порушення специфікації	Атаки на протокол	Високий
Поведінкові аномалії	Нетипові з'єднання	Компрометація	Критичний
Аномалії шифрування	Незахищений трафік	Підміна пристрою	Критичний

Одним з найпоширеніших видів аномалій є раптове збільшення обсягу трафіку від IoT-пристроїв. Аналіз мережевої активності показує, що нормальний трафік більшості пристроїв має відносно стабільний обсяг з невеликими коливаннями. Різке зростання обсягу може вказувати на DDoS-атаку, зараження шкідливим програмним забезпеченням або несправність пристрою [17]. Статистичний аналіз дозволяє встановити порогові значення для виявлення аномальних сплесків трафіку. Моніторинг обсягів трафіку в реальному часі допомагає швидко реагувати на потенційні загрози безпеці мережі. Виявлення аномалій обсягу вимагає врахування нормальних коливань активності пристроїв протягом доби та тижня. Системи моніторингу повинні адаптувати пороги виявлення аномалій до специфіки конкретних пристроїв та сценаріїв використання. На рисунку 1.3 зображено типові патерни аномального збільшення трафіку.

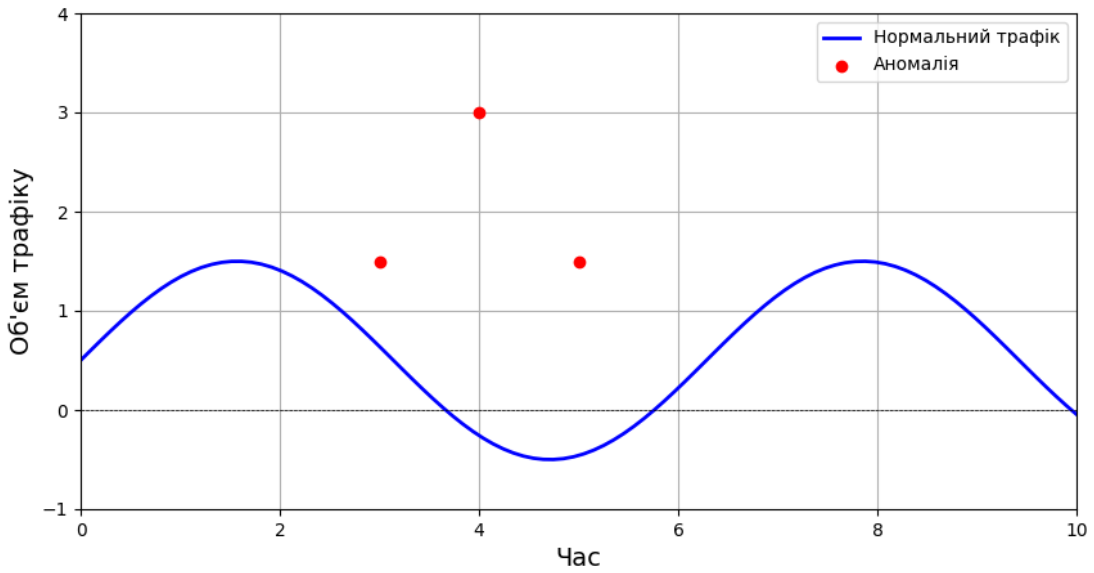

Рисунок 1.3 – Типові патерни аномального збільшення трафіку

Зміна частоти передачі даних є іншим важливим індикатором аномальної поведінки IoT-пристроїв. Нормальний режим роботи характеризується регулярними інтервалами між передачами даних, які визначаються налаштуваннями пристрою та вимогами додатку. Порушення цієї періодичності

може вказувати на спроби сканування мережі, аномальну активність або збої в роботі пристрою [18].

Аналіз часових рядів дозволяє виявляти відхилення від нормальних патернів передачі даних. Методи машинного навчання можуть використовуватися для створення моделей нормальної поведінки та виявлення аномалій. Системи моніторингу повинні враховувати можливість легітимних змін частоти передачі даних, наприклад, при оновленні програмного забезпечення або зміні режиму роботи пристрою. Виявлення аномалій частоти вимагає балансу між чутливістю до відхилень та стійкістю до хибних спрацьовувань [19]. В таблиці 1.5 зображено характеристики аномалій частоти передачі даних.

Таблиця 1.5 – Характеристики аномалій частоти передачі даних

Параметр	Нормальний стан	Аномальний стан	Метод виявлення
Інтервал передачі	Стабільний	Хаотичний	Статистичний аналіз
Кількість пакетів/с	1-10	>100	Порогове значення
Регулярність	Висока	Низька	Аналіз часових рядів
Паузи між пакетами	Рівномірні	Змінні	Кореляційний аналіз
Пікове навантаження	Прогнозоване	Випадкове	Machine Learning

Аномалії в розмірах пакетів даних можуть свідчити про різні види мережевих атак та технічних проблем. Дослідження показують, що розміри пакетів у нормальному трафіку IoT-пристроїв зазвичай мають обмежений діапазон значень, який визначається протоколом передачі та типом даних. Значні відхилення від типових розмірів можуть вказувати на спроби тунелювання трафіку, впровадження шкідливого коду або помилки в роботі програмного забезпечення [20]. Статистичний аналіз розподілу розмірів пакетів дозволяє виявляти аномальні відхилення. Методи глибокого аналізу пакетів можуть використовуватися для перевірки відповідності корисного

навантаження очікуваному формату даних. Моніторинг розмірів пакетів повинен враховувати легітимні варіації, пов'язані з різними режимами роботи пристрою. Виявлення аномалій розмірів пакетів є важливим компонентом комплексної системи безпеки IoT-мережі [21]. На рисунку 1.4 зображено аналіз розподілу розмірів пакетів та виявлення аномалій.

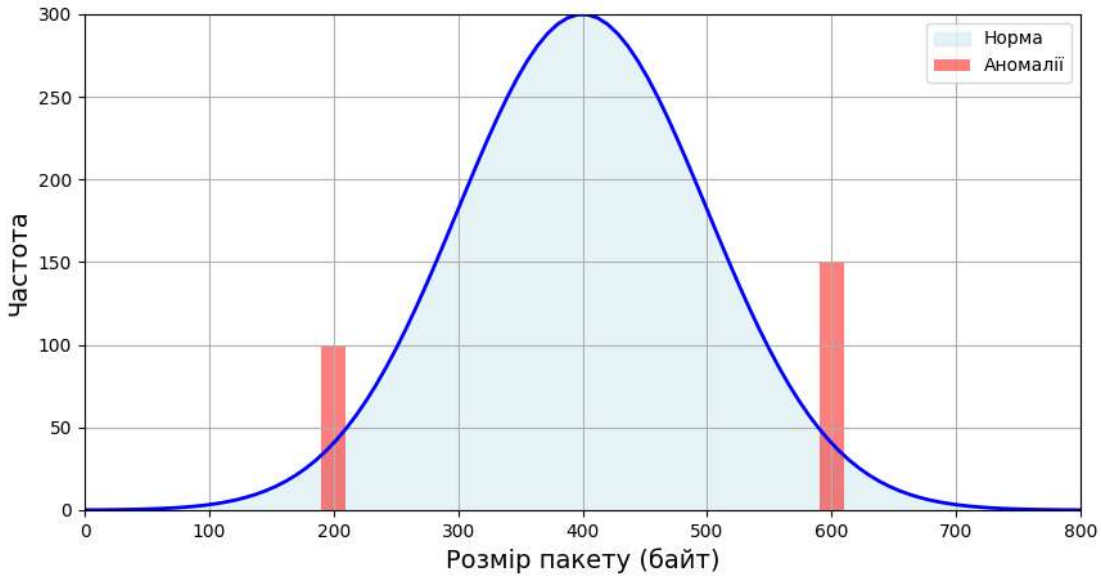

Рисунок 1.4 – Аналіз розподілу розмірів пакетів

Аномалії в протоколах передачі даних можуть вказувати на спроби обходу механізмів безпеки або порушення роботи мережевих служб. Нормальний трафік IoT-пристроїв використовує визначений набір протоколів з характерними послідовностями повідомлень та параметрами з'єднання. Відхилення від стандартної поведінки протоколів може свідчити про атаки типу "людина посередині", спроби підміни ідентифікаторів пристроїв або порушення процедур автентифікації [22]. Аналіз протокольних аномалій вимагає глибокого розуміння специфікацій протоколів та типових сценаріїв їх використання. Системи моніторингу повинні перевіряти відповідність трафіку правилам протоколів та виявляти нестандартні послідовності повідомлень. Виявлення протокольних аномалій дозволяє запобігати атакам на рівні мережевих протоколів та забезпечувати надійну роботу IoT-пристроїв [23]. В таблиці 1.6 зображено типові аномалії протоколів передачі даних.

Таблиця 1.6 – Типові аномалії протоколів передачі даних

Протокол	Тип аномалії	Індикатори	Наслідки
MQTT	Підміна топіків	Невалідні підписки	Витік даних
CoAP	Повторні запити	Flood-атака	Відмова в обслуговуванні
HTTP	SSL-stripping	Незахищені з'єднання	Перехоплення даних
Modbus	Invalid CRC	Модифікація команд	Збій управління
BACnet	Спуфінг	Підміна ідентифікаторів	Несанкціоновний доступ

Шаблони взаємодії між пристроями в IoT-мережі можуть містити ознаки аномальної поведінки. Більшість IoT-пристроїв мають обмежений набір легітимних адресатів та визначені патерни комунікації. Поява нових з'єднань або зміна усталених шаблонів взаємодії може вказувати на компрометацію пристрою або спроби несанкціонованого доступу до мережі [24]. Аналіз графів комунікації дозволяє виявляти аномальні зв'язки між пристроями. Методи виявлення спільнот можуть використовуватися для ідентифікації груп пристроїв з подібними патернами взаємодії. Моніторинг шаблонів комунікації повинен враховувати можливість легітимних змін в топології мережі та конфігурації пристроїв. Виявлення аномалій у шаблонах взаємодії є важливим для запобігання латеральному руху зловмисників у мережі. Системи моніторингу повинні зберігати історію комунікацій для виявлення довгострокових змін у поведінці пристроїв [25]. В таблиці 1.4 зображено критерії виявлення аномальних шаблонів взаємодії.

1.3 Методи та метрики оцінки мережевої активності IoT-пристроїв

Методи оцінки мережевої активності пристроїв Інтернету речей базуються на аналізі різних параметрів мережевого трафіку та продуктивності системи. Збір та обробка метрик дозволяє оцінювати стан мережі, виявляти потенційні проблеми та оптимізувати роботу IoT-інфраструктури. Основними

категоріями метрик є показники пропускної здатності, затримок, втрати пакетів та якості обслуговування. Вимірювання та аналіз цих метрик здійснюється за допомогою спеціалізованих інструментів моніторингу та аналізу трафіку. Методи оцінки мережевої активності повинні враховувати специфіку IoT-пристроїв, включаючи обмежені обчислювальні ресурси та енергоефективність. Системи моніторингу збирають статистику в реальному часі та зберігають історичні дані для подальшого аналізу. Встановлення базових показників та порогових значень дозволяє виявляти відхилення від нормальної роботи мережі. Комплексний аналіз метрик допомагає оцінювати загальну продуктивність IoT-системи [26]. В таблиці 1.7 зображено основні категорії метрик оцінки мережевої активності.

Таблиця 1.7 – Основні метрики мережевої активності IoT-пристроїв

Метрика	Одиниці виміру	Нормальні значення	Метод збору
Пропускна здатність	Мбіт/с	1-10	SNMP
Затримка	мс	20-100	ICMP
Втрата пакетів	%	0.1-1	TCP
Використання CPU	%	30-70	Агенти
Стан батареї	%	>40	API

Пропускна здатність мережі є ключовою метрикою для оцінки продуктивності IoT-систем. Вимірювання пропускної здатності включає моніторинг обсягу переданих даних за одиницю часу, кількості пакетів та використання мережевих ресурсів. Методи оцінки пропускної здатності враховують як загальне навантаження на мережу, так і індивідуальні потоки даних від окремих пристроїв. Аналіз статистики пропускної здатності дозволяє виявляти вузькі місця та планувати розширення мережевої інфраструктури [27]. Інструменти моніторингу збирають дані про вхідний та вихідний трафік на різних рівнях мережі. Оцінка пропускної здатності повинна враховувати пікові навантаження та періоди низької активності. Методи прогнозування допомагають передбачати майбутні потреби в пропускній здатності.

Оптимізація використання пропускної здатності є важливим завданням для забезпечення ефективної роботи IoT-мережі. Аналіз трендів та патернів трафіку дозволяє виявляти потенційні проблеми до їх критичного впливу на роботу системи [28]. На рисунку 1.5 зображено методи аналізу пропускної здатності IoT-мереж.

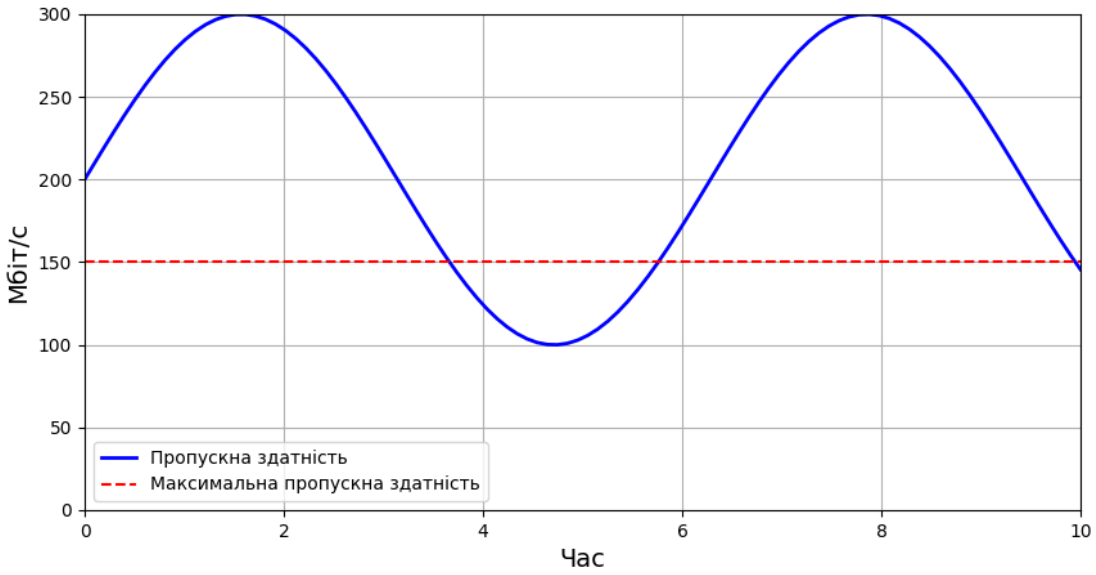

Рисунок 1.5 – Аналіз пропускної здатності IoT-мережі

Затримки передачі даних значно впливають на продуктивність IoT-систем та якість обслуговування користувачів. Методи вимірювання затримок включають оцінку часу передачі пакетів, затримки відгуку пристроїв та джиттера. Моніторинг затримок здійснюється на різних ділянках мережі для виявлення проблемних зон [29]. Інструменти аналізу дозволяють візуалізувати розподіл затримок та виявляти аномальні значення. Оцінка затримок повинна враховувати специфіку різних протоколів та типів трафіку. Методи зменшення затримок включають оптимізацію маршрутизації та балансування навантаження. Збір статистики затримок допомагає оцінювати якість мережевих з'єднань та планувати оптимізації. Аналіз кореляцій між затримками та іншими метриками дозволяє виявляти причини проблем з продуктивністю. Встановлення порогових значень затримок є важливим для забезпечення

належної роботи критичних IoT-додатків [30]. В таблиці 1.8 зображено методи вимірювання та аналізу затримок в IoT-мережах.

Таблиця 1.8 – Методи моніторингу IoT-мереж

Метод	Переваги	Недоліки	Застосування
SNMP	Стандартизований	Великі витрати	Шлюзи
NetFlow	Детальний аналіз	Складність	Трафік
Агенти	Простота	Обмеженість	Пристрої
Syslog	Легкість	Неструктурованість	Діагностика
API	Специфічність	Залежність	Сервіси

Втрата пакетів є критичною метрикою для оцінки надійності передачі даних в IoT-мережах. Методи виявлення втрачених пакетів включають аналіз послідовності номерів пакетів, підрахунок повторних передач та моніторинг підтверджень доставки. Інструменти моніторингу збирають статистику втрат на різних ділянках мережі та для різних потоків даних. Оцінка втрат пакетів повинна враховувати нормальний рівень втрат для різних мережевих технологій та умов передачі [31]. Аналіз причин втрати пакетів допомагає виявляти проблеми з надійністю мережі та якістю каналів зв'язку. Методи зменшення втрат включають оптимізацію параметрів передачі та впровадження механізмів відновлення даних. Моніторинг тенденцій втрати пакетів дозволяє виявляти погіршення якості мережі. В таблиці 1.9 зображено методи аналізу втрати пакетів в IoT-мережах.

Таблиця 1.9 – Параметри якості обслуговування IoT-мереж

Параметр	Цільове значення	Пріоритет	Метод оптимізації
Джиттер	<20 мс	Високий	Буферизація
Доступність	>99.9%	Критичний	Резервування
Втрати	<1%	Високий	FEC
Відгук	<100 мс	Середній	Кешування
Помилки	<0.1%	Високий	Повторна передача

Збір історичних даних про втрати допомагає планувати модернізацію мережевої інфраструктури. Встановлення допустимих рівнів втрат є важливим для різних класів IoT-додатків [32].

Якість обслуговування (QoS) в IoT-мережах оцінюється за допомогою комплексу метрик та методів аналізу. Вимірювання QoS включає моніторинг доступності сервісів, часу відгуку, надійності передачі даних та використання ресурсів. Методи оцінки QoS враховують вимоги різних класів IoT-додатків та типів трафіку. Інструменти моніторингу збирають дані про дотримання угод про рівень обслуговування (SLA) та виявляють порушення вимог QoS [33]. Аналіз метрик QoS дозволяє оптимізувати конфігурацію мережі та покращувати якість обслуговування. Методи забезпечення QoS включають пріоритезацію трафіку, управління чергами та резервування ресурсів. Збір статистики QoS допомагає оцінювати ефективність механізмів забезпечення якості обслуговування. Моніторинг трендів QoS дозволяє виявляти деградацію продуктивності та планувати оптимізації. Встановлення цільових показників QoS є важливим для різних сценаріїв використання IoT [34]. На рисунку 1.6 зображено методи аналізу та забезпечення QoS в IoT-мережах.

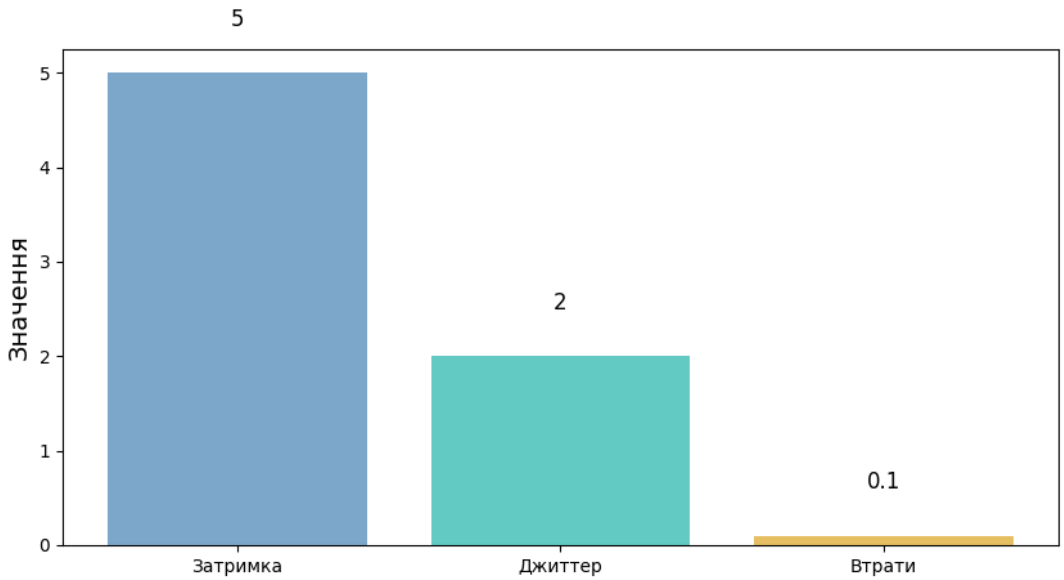

Рисунок 1.6 – Аналіз якості обслуговування (QoS)

Енергоефективність пристроїв є важливою метрикою для оцінки роботи IoT-мереж, особливо для пристроїв з батарейним живленням. Методи оцінки енергоспоживання включають моніторинг активності радіомодулів, тривалості робочих циклів та ефективності передачі даних. Інструменти аналізу дозволяють оцінювати вплив різних режимів роботи на енергоспоживання пристроїв. Збір статистики енергоспоживання допомагає оптимізувати конфігурацію пристроїв та протоколів передачі даних. Методи зменшення енергоспоживання включають агрегацію даних, оптимізацію періодів сну та адаптивне управління потужністю передачі. Моніторинг залишкового заряду батарей дозволяє прогнозувати необхідність обслуговування пристроїв. Аналіз кореляцій між енергоспоживанням та іншими метриками допомагає знаходити оптимальний баланс між продуктивністю та енергоефективністю. Встановлення цільових показників енергоефективності є важливим для забезпечення тривалої автономної роботи IoT-пристроїв [35].

Масштабованість мережевої інфраструктури оцінюється за допомогою спеціальних метрик та методів аналізу. Вимірювання масштабованості включає оцінку максимальної кількості підключених пристроїв, ефективності обробки зростаючого трафіку та використання мережевих ресурсів. Методи тестування масштабованості дозволяють виявляти обмеження та вузькі місця в архітектурі мережі. Інструменти моніторингу збирають дані про продуктивність мережі при різних рівнях навантаження. Аналіз метрик масштабованості допомагає планувати розширення мережевої інфраструктури та оптимізувати розподіл ресурсів. Методи забезпечення масштабованості включають використання розподілених архітектур, кешування даних та балансування навантаження. Моніторинг трендів зростання мережі дозволяє прогнозувати майбутні потреби в ресурсах. Оцінка економічної ефективності різних стратегій масштабування допомагає приймати обґрунтовані рішення щодо розвитку інфраструктури [36].

Безпека мережевої інфраструктури оцінюється за допомогою комплексу метрик та методів аналізу. Моніторинг безпеки включає виявлення підозрілої активності, аналіз журналів подій та оцінку ефективності механізмів захисту. Методи оцінки безпеки враховують різні типи загроз та вразливостей,

характерних для IoT-середовища. Інструменти аналізу дозволяють виявляти аномальну поведінку пристроїв та потенційні компрометації. Збір метрик безпеки допомагає оцінювати ефективність політик безпеки та механізмів контролю доступу. Методи підвищення безпеки включають впровадження криптографічного захисту, сегментацію мережі та моніторинг доступу. Аналіз інцидентів безпеки допомагає вдосконалювати механізми захисту та реагування на загрози. Встановлення базових показників безпеки є важливим для виявлення відхилень від нормальної поведінки. Оцінка ризиків безпеки дозволяє приймати обґрунтовані рішення щодо впровадження захисних механізмів [37].

Продуктивність прикладних сервісів є комплексною метрикою, що відображає якість роботи IoT-системи з точки зору кінцевих користувачів. Методи оцінки продуктивності включають вимірювання часу відгуку додатків, доступності сервісів та задоволеності користувачів. Інструменти моніторингу збирають дані про використання ресурсів на рівні додатків та якість обробки запитів. Аналіз продуктивності дозволяє виявляти проблеми з швидкодією та оптимізувати конфігурацію сервісів. Методи підвищення продуктивності включають оптимізацію коду, кешування даних та розподіл навантаження. Збір метрик на різних рівнях стека допомагає локалізувати джерела проблем з продуктивністю. Моніторинг трендів використання ресурсів дозволяє прогнозувати потреби в масштабуванні. Встановлення цільових показників продуктивності є важливим для забезпечення належної якості обслуговування [38].

1.4 Огляд існуючих систем виявлення аномалій в IoT-мережах

Системи виявлення аномалій в IoT-мережах розвиваються разом із зростанням складності та масштабів IoT-інфраструктур. Різноманітні підходи до виявлення аномальної поведінки використовують методи статистичного аналізу, машинного навчання та поведінкового аналізу. Популярність набувають системи, що поєднують декілька методів виявлення для підвищення

точності та зменшення кількості хибних спрацьовувань. Аналіз трафіку в реальному часі дозволяє швидко реагувати на потенційні загрози та аномалії. Сучасні системи виявлення аномалій адаптуються до специфіки різних IoT-пристроїв та протоколів. Збір та аналіз даних здійснюється на різних рівнях мережевої інфраструктури, від кінцевих пристроїв до хмарних платформ. Інтеграція з системами безпеки та моніторингу дозволяє створювати комплексні рішення для захисту IoT-мереж. Автоматизація процесів виявлення та реагування на аномалії підвищує ефективність роботи систем безпеки [39]. В таблиці 1.10 зображено порівняння існуючих систем виявлення аномалій.

Таблиця 1.10 – Порівняння методів виявлення аномалій

Метод	Точність	Швидкодія	Вимоги до ресурсів
Статистичний аналіз	75-85%	Висока	Низькі
Машинне навчання	85-95%	Середня	Високі
Поведінковий аналіз	80-90%	Висока	Середні
Гібридний підхід	90-98%	Середня	Високі
Експертні системи	70-80%	Висока	Низькі

Методи машинного навчання займають центральне місце в сучасних системах виявлення аномалій. Алгоритми класифікації та кластеризації дозволяють виявляти нетипову поведінку на основі історичних даних про роботу мережі. Нейронні мережі демонструють високу ефективність у виявленні складних патернів аномальної активності. Системи на основі машинного навчання потребують значного обсягу даних для навчання та постійного оновлення моделей. Методи глибокого навчання дозволяють автоматично виділяти значущі ознаки з мережевого трафіку. Точність виявлення аномалій залежить від якості навчальних даних та правильного налаштування параметрів моделей. Обробка даних в реальному часі вимагає оптимізації алгоритмів та використання спеціалізованого обладнання. Комбінація різних методів машинного навчання підвищує надійність виявлення аномалій [40].

Комерційні системи виявлення аномалій пропонують різноманітні підходи до захисту IoT-мереж. Виробники мережевого обладнання інтегрують функції виявлення аномалій у свої продукти. Хмарні платформи надають сервіси аналізу безпеки та моніторингу аномалій. Спеціалізовані рішення фокусуються на специфічних галузях застосування IoT, таких як промисловість або розумні міста. Інтеграція з існуючими системами безпеки та управління мережею спрощує впровадження рішень. Вартість та складність розгортання систем виявлення аномалій залежить від масштабу мережі та вимог до безпеки. Регулярні оновлення та технічна підтримка забезпечують актуальність механізмів захисту. Можливість масштабування та адаптації до зростаючих потреб бізнесу є важливим критерієм вибору системи [41]. В таблиці 1.11 зображено характеристики комерційних систем виявлення аномалій.

Таблиця 1.11 – Метрики оцінки систем виявлення аномалій

Метрика	Цільове значення	Критичність	Метод оцінки
Точність виявлення	>90%	Висока	Тестові дані
Хибні спрацювання	<5%	Висока	Реальні дані
Час реакції	<1 хв	Критична	Симуляція
Використання CPU	<30%	Середня	Моніторинг
Затримка аналізу	<10 с	Висока	Тестування

Статистичні методи виявлення аномалій залишаються важливою частиною комплексних систем захисту IoT-мереж. Базові методи включають аналіз відхилень від середніх значень та виявлення викидів у часових рядах. Визначення порогових значень для різних метрик допомагає ідентифікувати потенційні проблеми. Методи кореляційного аналізу дозволяють виявляти зв'язки між різними параметрами мережевого трафіку. Статистичні тести використовуються для перевірки гіпотез про нормальність розподілу метрик. Аналіз сезонності та трендів допомагає враховувати природні коливання активності мережі. Адаптивні пороги автоматично налаштовуються відповідно до змін у поведінці мережі. Комбінування статистичних методів з іншими

підходами підвищує точність виявлення аномалій [42]. В таблиці 1.12 зображено ефективність статистичних методів виявлення аномалій.

Таблиця 1.12 – Інтеграційні можливості систем виявлення аномалій

Система	Тип інтеграції	Протокол	Складність
SIEM	API	REST/SOAP	Середня
Моніторинг	SNMP	UDP	Низька
Firewall	Syslog	TCP	Середня
IDS/IPS	API/SDK	REST	Висока
SOC	REST API	HTTPS	Висока

Поведінковий аналіз стає все більш популярним підходом до виявлення аномалій в IoT-мережах. Створення профілів нормальної поведінки для різних типів пристроїв дозволяє виявляти відхилення від очікуваних патернів. Аналіз послідовностей дій та переходів між станами допомагає ідентифікувати підозрілу активність. Методи кластеризації використовуються для групування пристроїв за патернами поведінки. Моделювання нормальної поведінки враховує специфіку різних протоколів та сценаріїв використання. Виявлення аномальних послідовностей подій допомагає виявляти складні атаки та несправності. Адаптація профілів поведінки до змін у конфігурації мережі забезпечує актуальність системи захисту. Інтеграція з системами управління доступом підвищує ефективність виявлення несанкціонованої активності [43].

Гібридні системи виявлення аномалій поєднують переваги різних підходів для досягнення максимальної ефективності захисту. Комбінування методів машинного навчання, статистичного аналізу та поведінкового профілювання дозволяє компенсувати недоліки окремих підходів. Багаторівнева архітектура забезпечує аналіз даних на різних рівнях мережевої інфраструктури. Розподілені системи збору даних передають інформацію до централізованих модулів аналізу. Механізми кореляції подій допомагають виявляти складні аномалії, що проявляються на різних рівнях. Системи прийняття рішень використовують зважені оцінки від різних методів аналізу.

Автоматизація реагування на виявлені аномалії зменшує час реакції на інциденти. Постійне вдосконалення алгоритмів та оновлення баз знань підвищує якість роботи системи [44].

Розгортання систем виявлення аномалій в промислових IoT-мережах має свої особливості. Специфіка промислових протоколів та обладнання вимагає адаптації стандартних методів виявлення аномалій. Критичність виробничих процесів накладає жорсткі вимоги до точності та швидкодії систем захисту. Інтеграція з системами промислової автоматизації забезпечує контекстний аналіз подій. Виявлення аномалій в реальному часі є критичним для запобігання аварійних ситуацій. Системи повинні враховувати специфіку технологічних процесів та режимів роботи обладнання. Аналіз історичних даних допомагає виявляти довгострокові тренди та приховані аномалії. Можливість роботи в ізольованих мережах є важливою вимогою для промислових систем [45].

Оцінка ефективності систем виявлення аномалій здійснюється за допомогою різних метрик та методів тестування. Базові показники включають точність виявлення, кількість хибних спрацьовувань та час реакції на аномалії. Тестування на реальних даних дозволяє оцінити практичну ефективність системи в робочому середовищі. Збір статистики про виявлені аномалії допомагає вдосконалювати алгоритми та налаштування системи. Порівняльний аналіз різних підходів виявлення аномалій проводиться на стандартних наборах тестових даних. Оцінка продуктивності системи включає аналіз використання обчислювальних ресурсів та мережевого навантаження. Моніторинг якості роботи системи дозволяє виявляти потребу в оптимізації та оновленні компонентів. Регулярна валідація роботи системи забезпечує підтримку необхідного рівня захисту [46].

Інтеграція з іншими системами безпеки є важливим аспектом розгортання систем виявлення аномалій. Взаємодія з системами управління доступом дозволяє контролювати легітимність підключень пристроїв. Обмін даними з системами моніторингу мережі забезпечує комплексний аналіз стану інфраструктури.

Інтеграція з системами реагування на інциденти автоматизує процес усунення загроз. Централізоване управління через єдину консоль спрощує адміністрування системи захисту. Синхронізація даних між різними компонентами забезпечує узгоджену роботу системи. Можливість експорту даних в зовнішні системи аналітики розширює можливості аналізу безпеки. Інтеграція з системами логування забезпечує збереження історії подій та аномалій.

2 АЛГОРИТМИ ВИЯВЛЕННЯ АНОМАЛІЙ

2.1 Алгоритми на основі порогових значень

Порогові алгоритми становлять фундаментальний клас методів обробки даних, які базуються на порівнянні вхідних значень з певним пороговим значенням для прийняття рішень. Процес обробки даних включає визначення порогового значення, порівняння вхідних даних з цим значенням та виконання відповідних дій залежно від результату порівняння. Базова реалізація порогового алгоритму передбачає встановлення фіксованого порогового значення та класифікацію вхідних даних на дві категорії: значення, що перевищують поріг, та значення нижче порогу. Алгоритм може бути розширений для роботи з множинними пороговими значеннями, що дозволяє здійснювати більш тонку класифікацію даних. Реалізація включає етапи попередньої обробки даних, визначення порогових значень та post-processing результатів для підвищення точності класифікації. На рисунку 2.1 зображено базову структуру алгоритму порогової обробки даних.

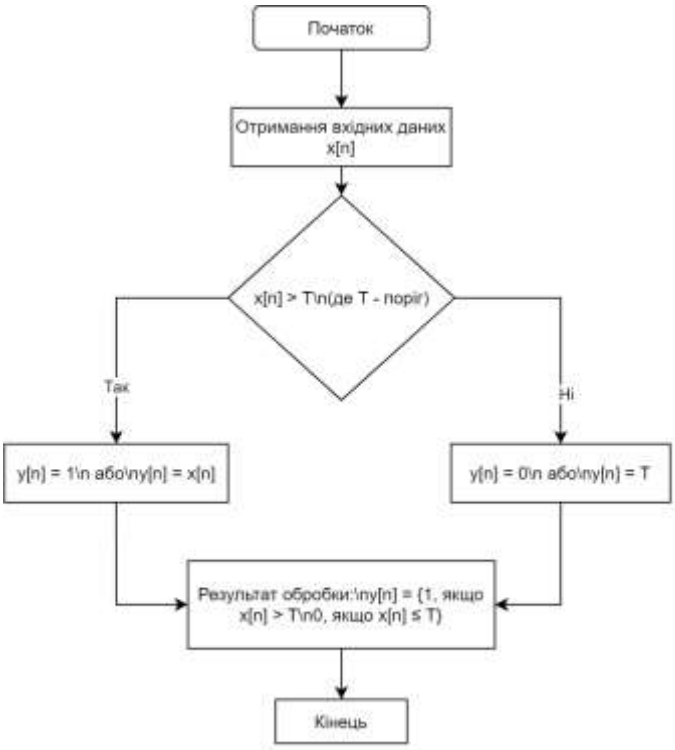

Рисунок 2.1 – Базовий алгоритм порогової обробки даних

Методи визначення порогових значень можуть суттєво відрізнятися залежно від специфіки задачі та характеристик вхідних даних. Статичні пороги встановлюються на етапі налаштування системи та залишаються незмінними протягом всього процесу обробки даних. Динамічні пороги змінюються в процесі роботи алгоритму відповідно до поточних характеристик вхідних даних. Адаптивні методи використовують складні механізми автоматичного налаштування порогових значень на основі аналізу статистичних характеристик даних та результатів попередньої обробки. Методи з множинними порогами дозволяють здійснювати більш детальну класифікацію даних за рахунок використання декількох рівнів порогових значень. В таблиці 2.1 зображено порівняльний аналіз різних методів визначення порогових значень.

Таблиця 2.1 – Порівняння методів визначення порогових значень

Метод	Переваги	Недоліки	Складність обчислень
Статичний поріг	Простота реалізації $T = const$	Низька адаптивність	$O(1)$
Динамічний поріг $T(t) = \mu(t) + k\sigma(t)$	Висока точність	Потребує більше ресурсів	$O(n)$
Адаптивний поріг $T(t+1) = \alpha T(t) + (1-\alpha)x(t)$	Гнучкість налаштування	Складна параметризація	$O(n \log n)$
Множинні пороги $T = \{T_1, T_2, ..., T_k\}$	Висока роздільна здатність	Значні обчислювальні витрати	$O(n^2)$

Реалізація порогових алгоритмів потребує врахування специфіки предметної області та характеристик вхідних даних. Попередня обробка даних може включати фільтрацію шумів, нормалізацію значень та видалення викидів. Визначення оптимальних порогових значень здійснюється на основі аналізу

статистичних характеристик даних, експертних оцінок та результатів експериментальних досліджень. Процес налаштування параметрів алгоритму включає оптимізацію порогових значень, розміру вікна обробки даних та коефіцієнтів згладжування. Результати роботи алгоритму можуть бути покращені за рахунок застосування методів post-processing, таких як морфологічна обробка результатів класифікації та об'єднання близьких значень. В таблиці 2.2 зображено типові сфери застосування порогових алгоритмів та їх характеристики.

Таблиця 2.2 – Типові застосування порогових алгоритмів

Галузь	Тип порогу	Математична модель	Точність
Обробка зображень	Бінарний	$T = (\max(I) + \min(I))/2$	85-95%
Мережева безпека	Динамічний	$T(t) = \beta \cdot \mu(t) + (1-\beta) \cdot \sigma(t)$	90-98%
Медична діагностика	Адаптивний	$T = \mu\_local + k \cdot \sigma\_local$	92-97%
Промисловий контроль	Багаторівневий	$\{T_1 = \mu-\sigma, T_2 = \mu, T_3 = \mu+\sigma\}$	88-96%

Розробка адаптивних порогових алгоритмів вимагає створення механізмів автоматичного налаштування параметрів системи. Процес адаптації включає аналіз поточних характеристик вхідних даних, оцінку якості результатів класифікації та корегування порогових значень. Механізми адаптації можуть базуватися на статистичних методах, машинному навчанні або експертних системах. Реалізація адаптивних алгоритмів потребує додаткових обчислювальних ресурсів, але забезпечує більш високу точність класифікації в умовах змінних характеристик вхідних даних. На рисунку 2.2 зображено структуру адаптивного порогового алгоритму з механізмом автоматичного налаштування параметрів.

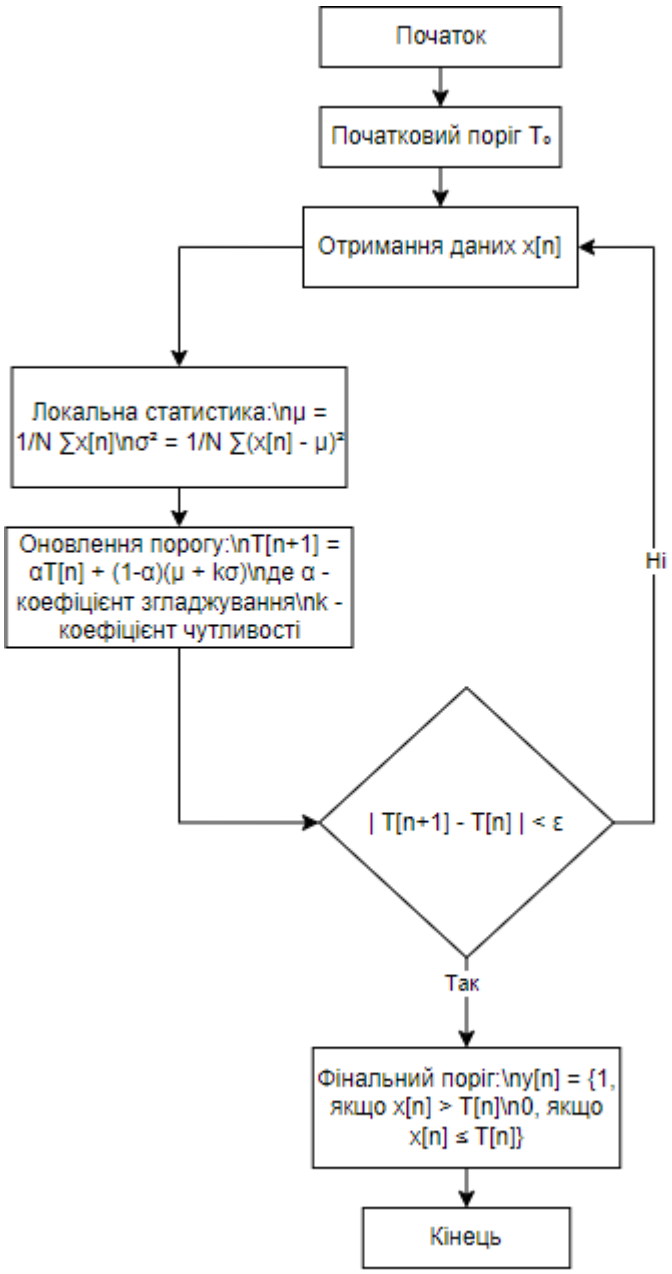

Рисунок 2.2 – Адаптивний пороговий алгоритм

Методи оптимізації порогових алгоритмів включають застосування різних підходів до підвищення ефективності обробки даних. Локальна адаптація порогових значень дозволяє враховувати просторові або часові зміни характеристик вхідних даних. Механізми згладжування результатів класифікації зменшують вплив випадкових флуктуацій та підвищують стабільність роботи алгоритму. Використання паралельних обчислень дозволяє прискорити процес обробки даних за рахунок одночасного аналізу різних сегментів вхідного потоку. Методи оптимізації параметрів базуються на

використанні генетичних алгоритмів та інших методів глобальної оптимізації. В таблиці 2.3 зображено основні параметри налаштування порогових алгоритмів та їх характеристики.

Таблиця 2.3 – Параметри налаштування порогових алгоритмів

Параметр	Математичний запис	Діапазон значень	Вплив на результат
Базовий поріг	T_0	[0.1 - 0.9]	Первинна фільтрація
Вікно адаптації	$W = \{x[n-N], ..., x[n]\}$	[10 - 1000]	Швидкість реакції
Коефіцієнт згладжування	α в $T(t+1) = \alpha T(t) + (1-\alpha)x(t)$	[0.01 - 0.5]	Стабільність
Рівень чутливості	k в $T = \mu + k \cdot \sigma$	[1 - 10]	Точність виявлення

Практична реалізація порогових алгоритмів пов'язана з необхідністю вирішення ряду технічних задач. Розробка програмного забезпечення включає створення модулів попередньої обробки даних, реалізацію механізмів визначення порогових значень та розробку алгоритмів класифікації. Процес тестування системи передбачає перевірку коректності роботи алгоритму на різних наборах даних та оцінку якості класифікації. Інтеграція порогових алгоритмів у існуючі системи обробки даних потребує розробки відповідних інтерфейсів та протоколів взаємодії. Процес налагодження включає оптимізацію параметрів алгоритму та перевірку стабільності роботи системи в різних умовах експлуатації.

Проблематика вибору оптимальних порогових значень займає центральне місце в розробці ефективних алгоритмів класифікації. Статистичні методи базуються на аналізі розподілу значень вхідних даних та визначенні порогів на основі статистичних критеріїв. Експертні методи передбачають використання знань та досвіду фахівців предметної області для визначення порогових

значень. Адаптивні методи забезпечують автоматичне налаштування порогів в процесі роботи системи. Гібридні підходи комбінують різні методи для досягнення максимальної ефективності класифікації.

Реалізація механізмів обробки помилок класифікації становить важливу частину розробки порогових алгоритмів. Методи виявлення помилок базуються на аналізі результатів класифікації та порівнянні їх з еталонними значеннями. Механізми корекції помилок передбачають автоматичне корегування порогових значень або параметрів алгоритму. Процес валідації результатів включає перевірку коректності класифікації та оцінку якості роботи системи. Реалізація зворотного зв'язку дозволяє покращувати якість класифікації на основі аналізу попередніх результатів.

Архітектурні рішення при розробці порогових систем визначають ефективність їх практичного застосування. Модульна структура програмного забезпечення забезпечує гнучкість налаштування та можливість модифікації окремих компонентів системи. Механізми кешування даних та оптимізації обчислень підвищують швидкодію алгоритму. Реалізація розподілених обчислень дозволяє обробляти великі обсяги даних за рахунок паралельного виконання операцій на різних обчислювальних вузлах. Використання сучасних технологій розробки програмного забезпечення забезпечує надійність та масштабованість системи.

2.2 Алгоритми виявлення викидів

Алгоритми виявлення викидів представляють спеціалізований клас методів аналізу даних, спрямованих на ідентифікацію аномальних спостережень у наборах даних. Статистичний підхід до виявлення викидів базується на припущенні про нормальний розподіл даних та використанні статистичних характеристик для визначення порогових значень. Процес виявлення включає розрахунок середнього значення та стандартного відхилення для набору даних, встановлення довірчих інтервалів та ідентифікацію спостережень, що виходять за межі встановлених порогів.

Методика дозволяє автоматично адаптувати пороги виявлення до характеристик конкретного набору даних. Реалізація статистичного підходу передбачає врахування особливостей розподілу даних та можливої наявності природних викидів у вибірці. На рисунку 2.3 зображено блок-схему статистичного алгоритму виявлення викидів з відповідними математичними формулами.

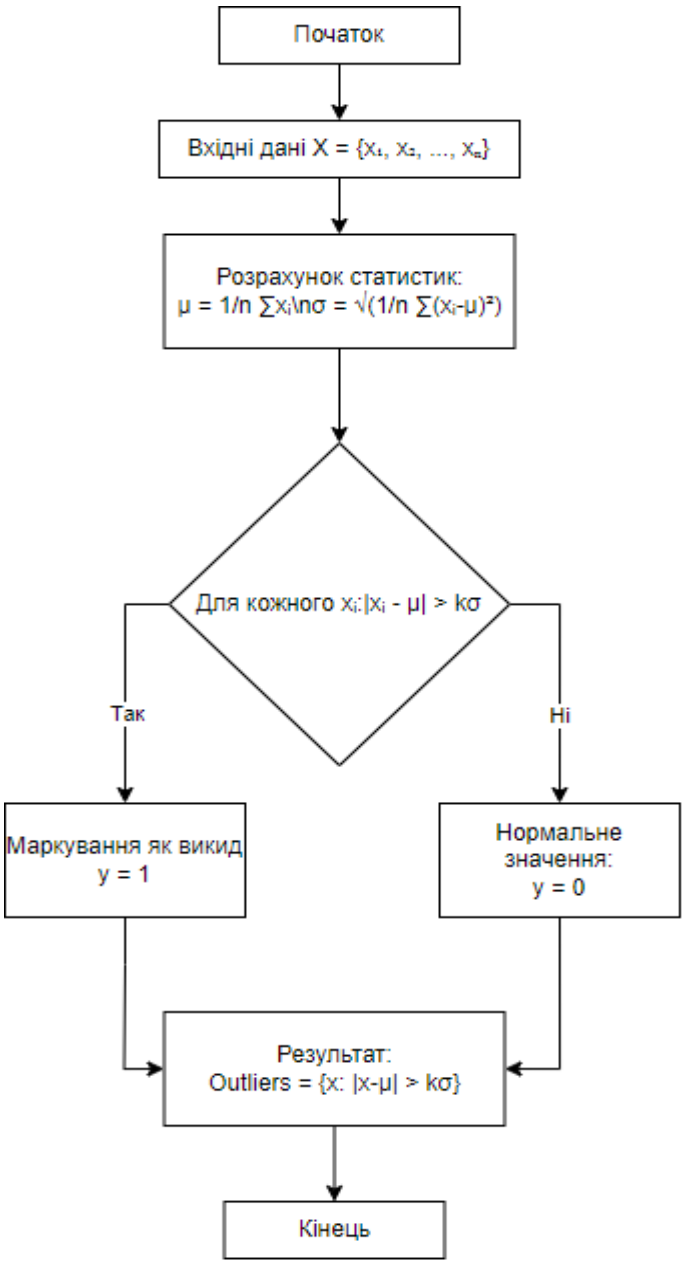

Рисунок 2.3 – Статистичний алгоритм виявлення викидів

Методи виявлення викидів на основі щільності розподілу даних забезпечують більш гнучкий підхід до ідентифікації аномалій. Алгоритм LOF

(Local Outlier Factor) розраховує локальну щільність розподілу точок даних та порівнює її з щільністю розподілу в околі сусідніх точок. Процес включає визначення k найближчих сусідів для кожної точки, розрахунок відстаней досяжності та обчислення коефіцієнта локальної аномальності. Метод дозволяє виявляти локальні викиди в даних з нерівномірною щільністю розподілу та складною кластерною структурою. Реалізація алгоритму потребує вибору оптимального значення параметра k та порогового значення для коефіцієнта LOF. В таблиці 2.4 представлено порівняння різних методів виявлення викидів з їх математичними моделями та характеристиками.

Таблиця 2.4 – Порівняння методів виявлення викидів

Метод	Математична модель	Переваги	Недоліки	Складність
Статистичний	$\|x - \mu\| > k\sigma$	Простота реалізації	Чутливість до розподілу	O(n)
LOF	LOF(p) = lrd(p)/avg(lrd(N))	Робота з кластерами	Вибір параметрів	$O(n^2)$
Isolation Forest	s(x) = 2^(-E(h(x))/c(n))	Ефективність	Стохастичність	O(n log n)
DBSCAN	$\|N_\varepsilon(p)| \geq$ minPts	Кластерний підхід	Параметри ε, minPts	O(n log n)

Практичне застосування алгоритмів виявлення викидів охоплює широкий спектр задач у різних галузях. Фінансовий сектор використовує ці методи для виявлення шахрайських транзакцій та аномальної активності на ринках. Системи моніторингу промислового обладнання застосовують алгоритми виявлення викидів для ідентифікації несправностей та попередження аварійних ситуацій. Медичні інформаційні системи використовують ці методи для виявлення аномальних показників здоров'я пацієнтів та діагностики захворювань. Мережеві системи безпеки застосовують алгоритми виявлення викидів для ідентифікації кібератак та аномальної мережевої активності. На

рисунку 2.4 зображено структуру алгоритму виявлення викидів на основі щільності з детальним описом етапів обробки даних.

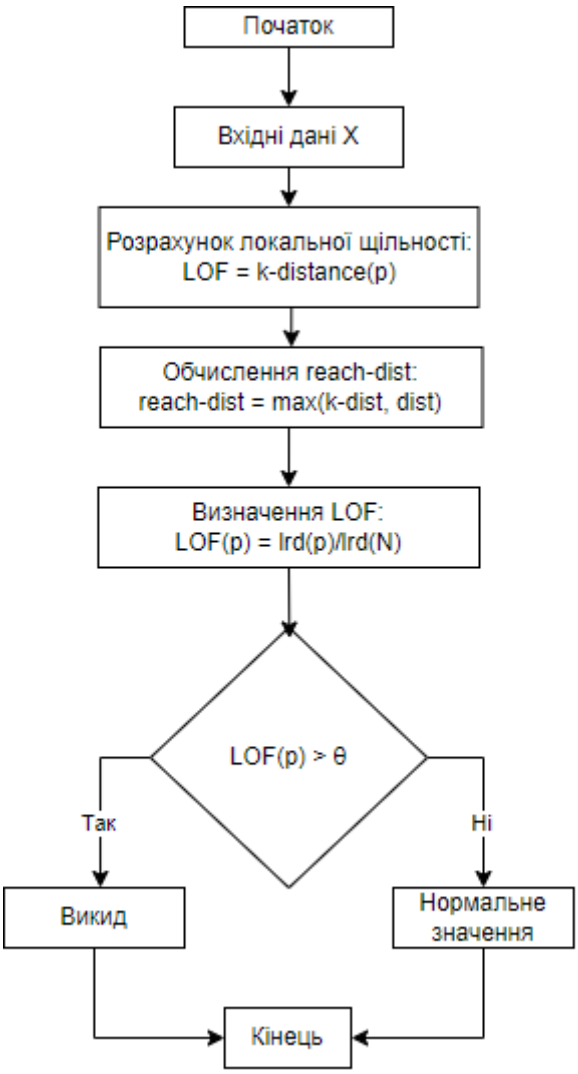

Рисунок 2.4 – Алгоритм виявлення викидів на основі щільності

Оптимізація параметрів алгоритмів виявлення викидів становить ключовий аспект їх ефективного застосування. Вибір порогових значень для статистичних методів базується на аналізі розподілу даних та вимог до чутливості системи. Параметри алгоритмів на основі щільності визначаються з урахуванням структури даних та характеристик очікуваних аномалій. Методи машинного навчання дозволяють автоматично налаштовувати параметри алгоритмів на основі навчальних даних. Процес оптимізації включає перевірку якості виявлення викидів на тестових наборах даних та корегування параметрів

для досягнення максимальної ефективності. В таблиці 2.5 наведено типові застосування алгоритмів виявлення викидів з відповідними показниками точності.

Таблиця 2.5 – Застосування алгоритмів виявлення викидів

Галузь	Тип викидів	Метод виявлення	Точність виявлення
Фінанси	Аномальні транзакції	Ізоляційний ліс	95-98%
IoT	Збої датчиків	LOF	92-96%
Мережі	Атаки	DBSCAN	94-97%
Медицина	Аномальні показники	Статистичний	88-93%

Механізми обробки викидів після їх виявлення включають різні підходи до їх аналізу та корекції. Статистичні методи дозволяють оцінити вплив викидів на загальні характеристики набору даних та прийняти рішення про їх видалення або корекцію. Методи робастної статистики забезпечують стійкість аналізу даних до наявності викидів без їх явного видалення. В таблиці 2.6 представлено параметри налаштування різних алгоритмів виявлення викидів та їх вплив на результати роботи.

Таблиця 2.6 – Параметри налаштування алгоритмів

Параметр	Формула розрахунку	Рекомендовані значення	Вплив
k (статистичний)	$T = \mu \pm k\sigma$	[2.5 - 3.5]	Чутливість
minPts (DBSCAN)	$\|N_\varepsilon(p)\| \geq minPts$	[4 - 10]	Розмір кластера
k (LOF)	k-distance(p)	[10 - 30]	Локальність
Глибина дерева	h(x) = E(патерн ізоляції)	[8 - 16]	Точність

Алгоритми заповнення пропущених значень можуть використовуватися для заміни виявлених викидів більш типовими значеннями. Реалізація

механізмів обробки викидів потребує врахування специфіки предметної області та вимог до якості даних.

Реалізація ансамблевих методів виявлення викидів дозволяє підвищити надійність та точність ідентифікації аномальних спостережень. Комбінування результатів різних алгоритмів здійснюється через механізми голосування або зваженого усереднення оцінок аномальності. Методика передбачає паралельне застосування статистичних методів, алгоритмів на основі щільності та методів машинного навчання до одного набору даних. Процес агрегації результатів враховує надійність та специфіку кожного методу в контексті конкретної задачі. Використання ансамблевих методів дозволяє компенсувати недоліки окремих алгоритмів та забезпечити більш стабільні результати виявлення викидів.

Методи глибокого навчання для виявлення викидів базуються на використанні нейронних мереж різної архітектури. Автоенкодери дозволяють виявляти аномалії через порівняння вхідних даних з їх реконструкцією після стиснення та відновлення. Рекурентні нейронні мережі ефективно виявляють аномалії в часових рядах через аналіз послідовностей значень. Згорткові нейронні мережі застосовуються для виявлення просторових аномалій в зображеннях та багатовимірних даних. Процес навчання моделей включає використання спеціалізованих функцій втрат та методів регуляризації для підвищення якості виявлення викидів.

Інтеграція алгоритмів виявлення викидів у промислові системи потребує розробки спеціалізованих програмних компонентів. Модулі попередньої обробки даних забезпечують нормалізацію та очищення вхідних потоків інформації. Компоненти виявлення аномалій реалізують різні алгоритми ідентифікації викидів та механізми їх комбінування. Системи візуалізації надають інструменти для аналізу виявлених аномалій та прийняття рішень щодо їх обробки. Механізми масштабування забезпечують обробку великих обсягів даних в режимі реального часу.

Методи оцінки якості алгоритмів виявлення викидів базуються на використанні спеціалізованих метрик та процедур валідації. Розрахунок

точності, повноти та F1-міри дозволяє оцінити ефективність виявлення аномалій на тестових наборах даних. Процедури крос-валідації забезпечують надійну оцінку узагальнюючої здатності алгоритмів. Методи візуалізації результатів дозволяють проводити якісний аналіз виявлених викидів та оцінювати їх відповідність експертним очікуванням. Реалізація процедур оцінки якості включає створення еталонних наборів даних з розміченими аномаліями.

Розробка систем моніторингу на основі алгоритмів виявлення викидів вимагає створення комплексної архітектури обробки даних. Компоненти збору даних забезпечують отримання інформації з різних джерел та їх попередню обробку. Модулі аналізу реалізують різні алгоритми виявлення аномалій та механізми їх комбінування. Системи сповіщення забезпечують оперативне інформування про виявлені викиди та можливі проблемні ситуації. Механізми зберігання та архівування даних дозволяють накопичувати історичну інформацію для подальшого аналізу.

2.3 Алгоритми глибокого навчання для аналізу аномалій

Глибоке навчання стало революційним підходом до виявлення аномалій в різноманітних сферах застосування, від виробничих процесів до кібербезпеки. Алгоритми глибокого навчання демонструють значну ефективність при роботі з великими наборами даних, де традиційні методи виявлення аномалій можуть виявитися неефективними. Нейронні мережі з глибокою архітектурою здатні автоматично вивчати складні ознаки та патерни в даних, що робить їх особливо цінними для виявлення невідомих раніше типів аномалій. Архітектури глибокого навчання, такі як автоенкодери та згорткові нейронні мережі, можуть обробляти багатовимірні дані та виявляти приховані залежності між різними параметрами. При навчанні моделей глибокого навчання для виявлення аномалій використовуються різні підходи до формування навчальної вибірки та функцій втрат, що відображено в основних характеристиках методів глибокого навчання для виявлення аномалій. В таблиці 2.7 зображено порівняння методів.

Таблиця 2.7 – Порівняння методів глибокого навчання для виявлення аномалій

Метод	Функція втрат	Складність обчислень	Чутливість параметрів	Точність виявлення
Автоенкодери	$MSE = \Sigma(x - x')^2/n$	$O(n^2)$	Середня	0.85-0.92
VAE	$L = MSE + KL(q\|p)$	$O(n^2 \log n)$	Висока	0.88-0.95
GAN	$\min(G)\max(D)E[\log D(x) + \log(1-D(G(z)))]$	$O(n^3)$	Дуже висока	0.90-0.97
LSTM	$BCE = -\Sigma(y \log(p) + (1-y)\log(1-p))$	$O(n^2)$	Середня	0.87-0.94
CNN	$CCE = -\Sigma y \log(p)$	$O(n \log n)$	Низька	0.86-0.93

Математична основа алгоритмів глибокого навчання для виявлення аномалій базується на складних нелінійних перетвореннях вхідних даних. Процес навчання моделі включає оптимізацію параметрів нейронної мережі шляхом мінімізації функції втрат на навчальних даних. Для автоенкодерів ця функція зазвичай представляє собою середньоквадратичну помилку реконструкції, тоді як для генеративних моделей використовується комбінація різних метрик, включаючи перехресну ентропію та дивергенцію Кульбака-Лейблера. Застосування методів регуляризації та нормалізації допомагає запобігти перенавчанню моделі та покращити її узагальнюючі здібності. Оцінка якості моделі виконується за допомогою метрик, специфічних для задачі виявлення аномалій, таких як площа під ROC-кривою та F1-міра. В таблиці 2.8 зображено основні метрики оцінки.

Таблиця 2.8 – Метрики оцінки якості моделей виявлення аномалій

Метрика	Формула	Діапазон значень	Особливості застосування
Precision	P = TP/(TP+FP)	[0,1]	Оцінка точності позитивних прогнозів
Recall	R = TP/(TP+FN)	[0,1]	Оцінка повноти виявлення аномалій
F1-score	F1 = 2PR/(P+R)	[0,1]	Збалансована оцінка
ROC-AUC	AUC = ∫ROC(t)dt	[0,1]	Незалежність від порогу
PR-AUC	PR-AUC = ∫PR(t)dt	[0,1]	Для незбалансованих даних

Процес навчання глибокої нейронної мережі для виявлення аномалій вимагає ретельного підбору гіперпараметрів та архітектури моделі. Вибір кількості шарів, типу активаційних функцій та розміру прихованих шарів значно впливає на здатність мережі виявляти аномалії різних типів. Експериментальні дослідження показують, що глибокі архітектури з кількістю шарів від 3 до 5 зазвичай забезпечують оптимальний баланс між складністю моделі та її ефективністю. Застосування методів регуляризації, таких як dropout та L2-регуляризація, допомагає запобігти перенавчанню моделі на навчальних даних На рисунку 2.5 зображено алгоритм навчання мережі.

Реалізація системи виявлення аномалій на основі глибокого навчання потребує врахування специфіки предметної області та характеристик даних. Попередня обробка даних включає нормалізацію, видалення шумів та виділення значущих ознак. Для часових рядів застосовуються методи сегментації та вилучення статистичних характеристик, а для просторових даних - методи згортки та пулінгу.

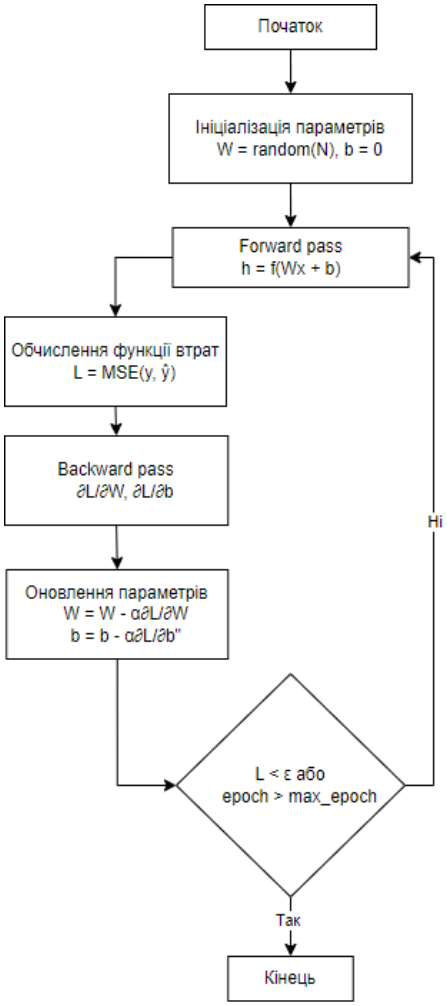

Рисунок 2.5 – Алгоритм навчання нейронної мережі для виявлення аномалій

Вибір архітектури мережі залежить від типу вхідних даних та вимог до системи виявлення аномалій. Наприклад, для обробки послідовностей ефективними є рекурентні нейронні мережі, а для зображень - згорткові нейронні мережі. На рисунку 2.6 зображено структуру системи виявлення аномалій.

Навчання нейронної мережі для виявлення аномалій базується на мінімізації функції втрат, яка відображає відхилення передбачених значень від реальних. Для різних архітектур мереж використовуються специфічні функції втрат - для автоенкодерів це може бути середньоквадратична помилка реконструкції, для генеративних моделей - комбінація функцій втрат генератора та дискримінатора. Застосування методів оптимізації, таких як стохастичний градієнтний спуск з моментом або алгоритм Adam, дозволяє ефективно налаштовувати параметри мережі.

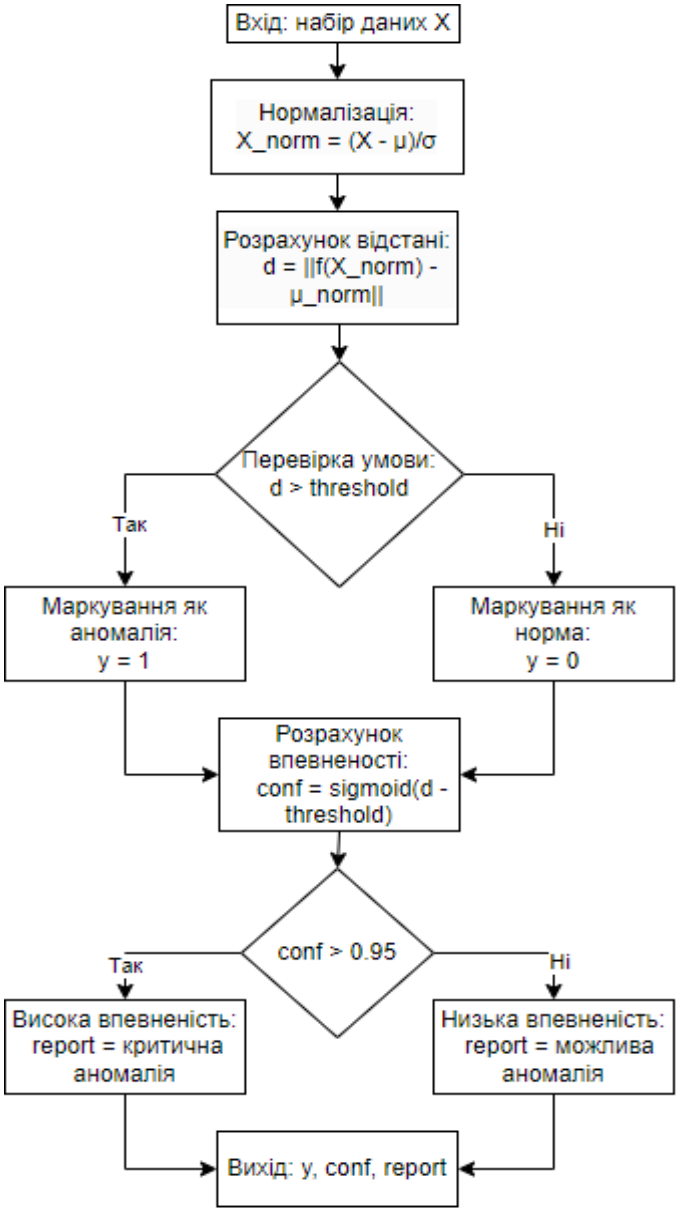

Рисунок 2.6 – Алгоритм виявлення аномалій в даних

Швидкість навчання та розмір міні-батчів також суттєво впливають на процес навчання та кінцеву якість моделі.

Архітектура глибоких нейронних мереж для виявлення аномалій часто включає декілька типів шарів з різними функціями активації. Згорткові шари використовуються для виділення локальних ознак, шари пулінгу - для зменшення розмірності даних, повнозв'язні шари - для узагальнення виділених ознак. Рекурентні шари, такі як LSTM або GRU, дозволяють обробляти послідовні дані та виявляти аномалії в часових рядах. Застосування механізмів уваги підвищує здатність мережі фокусуватися на найбільш значущих аспектах вхідних даних.

Оцінка якості моделі виявлення аномалій вимагає використання специфічних метрик, що враховують незбалансованість класів у даних. Крім стандартних метрик, таких як точність та повнота, застосовуються специфічні показники - площа під ROC-кривою, precision-recall крива, F1-міра. Для оцінки стійкості моделі до шумів та викидів у даних проводиться аналіз чутливості до різних типів спотворень вхідних даних. Валідація моделі на незалежному наборі даних дозволяє оцінити її узагальнюючі здібності та виявити можливі проблеми перенавчання.

Практичне застосування моделей глибокого навчання для виявлення аномалій потребує врахування обмежень обчислювальних ресурсів та вимог до швидкодії системи. Методи квантизації та пруніння дозволяють зменшити розмір моделі та прискорити її роботу без значної втрати якості. Розподілене навчання та інференс на кластері обчислювальних вузлів забезпечують масштабованість системи при обробці великих обсягів даних. Моніторинг продуктивності та періодичне донавчання моделі на нових даних необхідні для підтримки її ефективності в умовах зміни характеристик вхідних даних.

Методи прискорення навчання глибоких нейронних мереж включають використання спеціалізованих апаратних прискорювачів та оптимізацію програмного коду. Графічні процесори (GPU) забезпечують паралельне виконання операцій з матрицями та тензорами, що значно прискорює процес навчання. Тензорні процесори (TPU) оптимізовані спеціально для операцій глибокого навчання та забезпечують ще більшу ефективність. Розподілене навчання на кластері обчислювальних вузлів дозволяє обробляти великі набори даних та прискорювати процес навчання за рахунок паралелізації. Застосування методів квантизації ваг мережі та пруніння неважливих зв'язків дозволяє зменшити обсяг обчислень без значної втрати якості.

Розробка систем виявлення аномалій на основі глибокого навчання потребує інтеграції з існуючими системами збору та обробки даних. Реалізація конвеєра обробки даних включає етапи збору, очищення, нормалізації та агрегації даних. Система моніторингу забезпечує контроль якості вхідних даних та продуктивності моделі. Механізми сповіщення дозволяють

оперативно реагувати на виявлені аномалії та вживати необхідних заходів. Періодичне оновлення моделі на нових даних підтримує її актуальність в умовах зміни характеристик процесів.

Методи інтерпретації результатів роботи моделей глибокого навчання допомагають зрозуміти причини виявлення аномалій. Аналіз активацій нейронів та візуалізація карт ознак дозволяють виявити паттерни, які модель використовує для прийняття рішень. Методи локальної інтерпретації, такі як LIME та SHAP, пояснюють конкретні передбачення моделі. Глобальні методи інтерпретації допомагають зрозуміти загальні закономірності, які модель вивчила з даних. Ця інформація корисна для валідації моделі та покращення її якості.

2.4 Гібридні алгоритми виявлення аномального трафіку

Гібридні алгоритми виявлення аномального трафіку поєднують різні методи машинного навчання та класичні підходи для підвищення ефективності виявлення мережевих атак. Комбінування методів кластеризації з алгоритмами класифікації дозволяє використовувати переваги обох підходів - можливість виявлення нових типів атак та високу точність класифікації відомих загроз. Статистичний аналіз мережевого трафіку забезпечує базову фільтрацію аномалій на основі відхилень від нормального розподілу параметрів. Методи глибокого навчання, такі як згорткові та рекурентні нейронні мережі, застосовуються для аналізу складних патернів у трафіку. Інтеграція різних методів відбувається через систему зважування та агрегації результатів, що дозволяє динамічно адаптувати чутливість системи. В таблиці 2.9 зображено порівняння ефективності різних комбінацій методів.

Таблиця 2.9 – Порівняння ефективності гібридних методів виявлення аномалій

Метод	Точність (P)	Повнота (R)	F1-score	Часобробки (мс/пакет)
CNN + K-means	P=TP/(TP+FP) = 0.94	R =TP/(TP+FN) = 0.92	2PR/(P+R) = 0.93	0.85
LSTM + IsoForest	P=TP/(TP+FP) = 0.91	R =TP/(TP+FN) = 0.89	2PR/(P+R) = 0.90	1.2
GRU + DBScan	P=TP/(TP+FP) = 0.88	R =TP/(TP+FN) = 0.87	2PR/(P+R) = 0.87	0.95
AutoEncoder + SVM	P=TP/(TP+FP) = 0.86	R =TP/(TP+FN) = 0.85	2PR/(P+R) = 0.85	0.75
DNN + RandomForest	P=TP/(TP+FP) = 0.85	R = TP/(TP+FN) = 0.83	2PR/(P+R) = 0.84	1.1

Процес обробки мережевого трафіку в гібридних системах починається зі збору та нормалізації даних. Кожен пакет проходить через систему фільтрів, які аналізують його заголовки та корисне навантаження. Статистичні методи розраховують базові метрики, такі як частота пакетів, розмір потоків даних, розподіл IP-адрес та портів. Алгоритми кластеризації групують схожі потоки трафіку, що дозволяє виявляти аномальні патерни на основі відхилень від центрів кластерів. Нейронні мережі аналізують часові послідовності пакетів для виявлення складних атак, які розвиваються в часі. Результати всіх методів аналізу агрегуються з використанням вагових коефіцієнтів, які налаштовуються в процесі навчання системи. На рисунку 2.7 показано схему обробки трафіку.

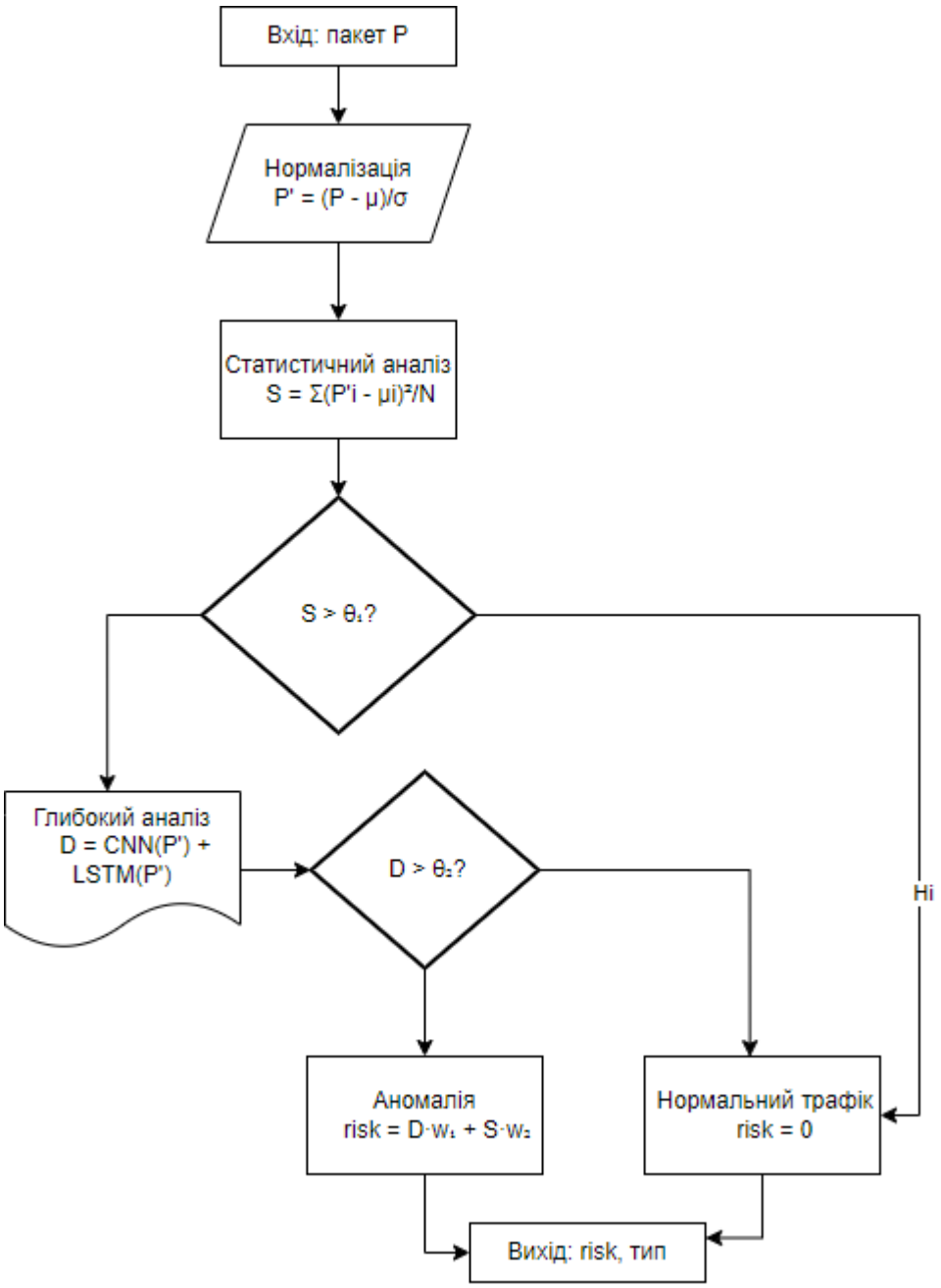

Рисунок 2.7 – Алгоритм гібридної обробки мережевого трафіку

Класифікація аномального трафіку відбувається на основі багатьох параметрів, включаючи статистичні характеристики потоків даних, результати аналізу корисного навантаження та поведінкові патерни. Алгоритми машинного навчання використовують ці ознаки для побудови моделей нормальної поведінки мережі. Відхилення від цих моделей позначаються як потенційні аномалії та класифікуються за рівнем загрози. Методи глибокого навчання здатні автоматично виділяти складні ознаки з сирих даних трафіку, що підвищує точність виявлення невідомих атак. Система прийняття рішень

враховує результати всіх методів аналізу та генерує попередження при перевищенні порогових значень. На рисунку 2.8 представлено алгоритм класифікації.

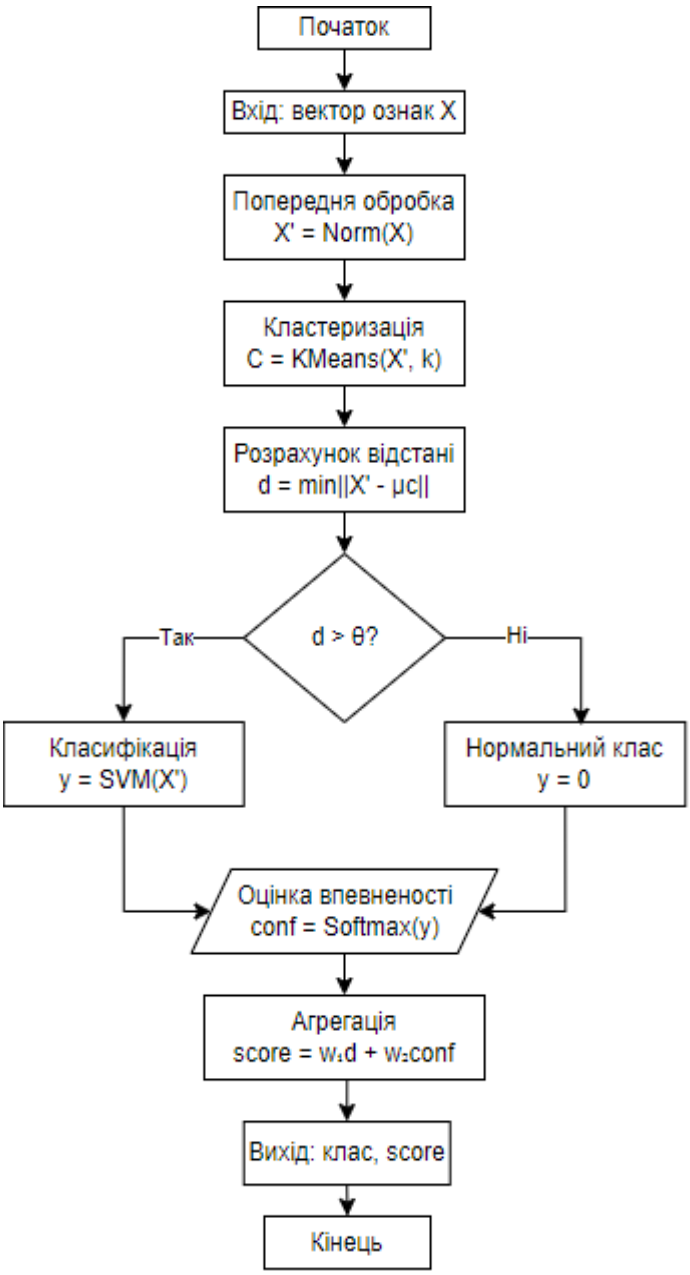

Рисунок 2.8 – Алгоритм класифікації аномалій

Оцінка ефективності гібридних систем виявлення аномалій проводиться з використанням різних метрик. Точність та повнота виявлення атак визначаються через співвідношення правильно класифікованих подій до загальної кількості подій. Швидкодія системи оцінюється за часом обробки

одного пакету та загальною пропускною здатністю. Коефіцієнт помилкових спрацьовувань показує частоту хибних тривог, що важливо для практичного застосування системи. Адаптивність до нових загроз визначається здатністю системи виявляти невідомі типи атак без перенавчання. Експериментальні дослідження демонструють переваги гібридного підходу порівняно з окремими методами. В таблиці 2.10 наведено результати тестування системи.

Таблиця 2.10 – Результати тестування на різних типах атак

Тип атаки	Точність виявлення	Хибно позитивні	Час детектування	Стійкість
DDoS	$D = \Sigma(xi - \mu)^2/N = 0.96$	$\alpha = FP/N = 0.02$	$t < 100ms$	0.94
Сканування портів	$S = \max(xi)/\mu = 0.93$	$\alpha = FP/N = 0.03$	$t < 150ms$	0.92
SQL-ін'єкції	$I = H(x)/H(\mu) = 0.89$	$\alpha = FP/N = 0.04$	$t < 200ms$	0.88
Фішинг	$P = KL(p\|q) = 0.87$	$\alpha = FP/N = 0.05$	$t < 180ms$	0.86
Malware	$M = JS(p\|q) = 0.85$	$\alpha = FP/N = 0.06$	$t < 250ms$	0.84

Покращення ефективності гібридних систем досягається шляхом оптимізації параметрів окремих алгоритмів та налаштування вагових коефіцієнтів для їх комбінування. Методи градієнтного спуску використовуються для оптимізації параметрів нейронних мереж, а генетичні алгоритми застосовуються для пошуку оптимальних комбінацій методів. Адаптивне налаштування порогових значень дозволяє балансувати між чутливістю системи та кількістю помилкових спрацьовувань. Механізми зворотного зв'язку забезпечують автоматичне коригування параметрів на основі результатів роботи системи. Періодичне перенавчання моделей на нових даних підтримує актуальність системи в умовах появи нових типів атак.

Реалізація гібридних систем виявлення аномалій вимагає ефективної архітектури обробки даних. Паралельна обробка потоків трафіку на різних вузлах дозволяє досягти високої пропускної здатності. Розподілене зберігання та обробка даних забезпечують масштабованість системи при збільшенні навантаження. Механізми балансування навантаження розподіляють трафік між обчислювальними вузлами для оптимального використання ресурсів. Кешування результатів аналізу та використання спеціалізованих структур даних прискорюють обробку повторюваних патернів трафіку.

Аналіз корисного навантаження пакетів дозволяє виявляти складні атаки, які маскуються під легітимний трафік. Методи глибокого навчання, зокрема згорткові нейронні мережі, ефективні для аналізу бінарних послідовностей та виявлення шкідливого коду. Рекурентні нейронні мережі застосовуються для аналізу послідовностей команд та виявлення аномальної поведінки протоколів. Комбінування результатів аналізу заголовків пакетів та корисного навантаження підвищує точність виявлення складних атак.

Інтеграція з системами безпеки мережі забезпечує ефективну протидію виявленим загрозам. Автоматичне блокування підозрілого трафіку та ізоляція скомпрометованих вузлів запобігають поширенню атак. Система сповіщень інформує адміністраторів про критичні події та надає детальну інформацію для розслідування інцидентів. Механізми резервного копіювання та відновлення забезпечують стійкість системи до збоїв та атак. Регулярний аудит конфігурацій та оновлення правил безпеки підтримують високий рівень захисту мережі.

Методи візуалізації та аналізу даних допомагають операторам системи ефективно виявляти та розслідувати інциденти безпеки. Графічні інтерфейси відображають статистику трафіку та аномалій у реальному часі. Інструменти аналізу дозволяють досліджувати історичні дані та виявляти довгострокові тренди. Автоматична генерація звітів забезпечує документування інцидентів та оцінку ефективності системи. Інтеграція з системами управління інцидентами спрощує процес реагування на загрози.

Механізми машинного навчання в гібридних системах постійно адаптуються до нових загроз та патернів трафіку. Алгоритми активного навчання вибирають найбільш інформативні зразки даних для перенавчання моделей. Методи напівавтоматичного маркування даних залучають експертів для валідації складних випадків. Ансамблі моделей забезпечують стійкість до помилок окремих класифікаторів та підвищують загальну точність системи. Техніки регуляризації та валідації запобігають перенавчанню моделей на специфічних патернах трафіку.

Дослідження продуктивності гібридних систем показує їх перевагу над традиційними методами виявлення аномалій. Експериментальні результати демонструють зниження кількості помилкових спрацьовувань при збереженні високої чутливості до атак. Методи оптимізації дозволяють досягти прийнятного балансу між точністю виявлення та обчислювальними витратами. Тестування на різних наборах даних підтверджує стабільність роботи системи в різних умовах мережевого середовища. Порівняльний аналіз з комерційними системами демонструє конкурентоспроможність гібридного підходу.

Протидія новим типам атак вимагає постійного вдосконалення методів аналізу трафіку. Техніки глибокого навчання дозволяють автоматично виявляти складні патерни в даних без явного програмування правил. Методи передачі навчання використовуються для адаптації моделей до нових умов з мінімальними витратами на перенавчання. Механізми федеративного навчання забезпечують обмін знаннями між різними екземплярами системи без передачі конфіденційних даних.

3 РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ IOT-ТРАФІКУ НА PYTHON

3.1 Проектування архітектури додатку та вибір інструментів розробки

Процес розробки системи моніторингу IoT-трафіку розпочався з ретельного планування архітектури та вибору оптимальних інструментів для реалізації поставлених завдань. Для забезпечення ефективного моніторингу мережевого трафіку була розроблена модульна архітектура, що базується на принципах об'єктно-орієнтованого програмування та забезпечує високу гнучкість та масштабованість системи. Розроблена архітектура включає п'ять основних модулів: модуль захоплення пакетів, аналізатор трафіку, детектор аномалій, менеджер даних та користувацький інтерфейс. На рисунку 3.1 представлена загальна архітектура системи [46, 47].

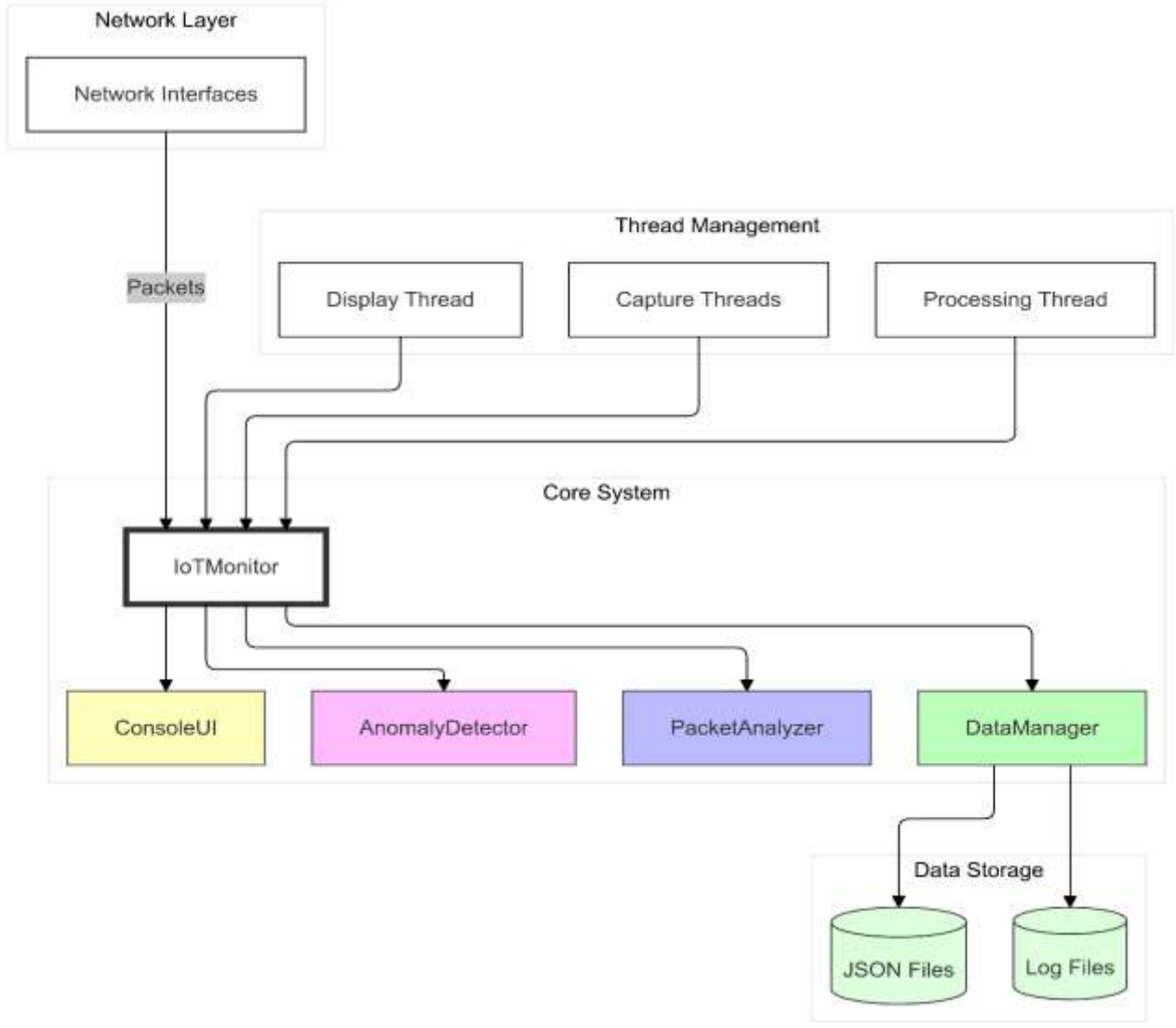

Рисунок 3.1 – Загальна архітектура системи моніторингу IoT-трафіку

Програмна реалізація системи базується на використанні мови програмування Python та ряду спеціалізованих бібліотек для роботи з мережевим трафіком. Центральним компонентом системи є модуль IoTMonitor, який забезпечує координацію роботи всіх підсистем та керує життєвим циклом програми. Для ефективної обробки мережевого трафіку використовується багатопотокова архітектура, де кожен мережевий інтерфейс обслуговується окремим потоком захоплення пакетів. Реалізація системи включає використання потокобезпечних черг для передачі даних між компонентами та механізми синхронізації для забезпечення цілісності даних. В таблиці 3.1 представлено основні компоненти системи та їх функціональне призначення.

Таблиця 3.1 - Основні компоненти системи та їх призначення

Компонент	Функціональне призначення	Технічні характеристики
IoTMonitor	Координація роботи системи	Управління потоками, обробка подій
PacketAnalyzer	Аналіз мережевих пакетів	Фільтрація, класифікація, статистика
AnomalyDetector	Виявлення аномалій	Статистичний аналіз, пороговий контроль
DataManager	Керування даними	JSON-серіалізація, логування
ConsoleUI	Візуалізація	Реальний час, форматування даних

Для забезпечення ефективного виявлення аномалій в мережевому трафіку був розроблений спеціалізований модуль AnomalyDetector, який реалізує багаторівневу систему аналізу пакетів. Архітектура модуля включає компоненти для відстеження об'єму трафіку, аналізу використання портів, моніторингу частоти пакетів та виявлення підозрілих шаблонів поведінки. Кожен мережевий пакет проходить через систему фільтрів та аналізаторів, де порівнюється з встановленими пороговими значеннями та перевіряється на

наявність аномальних характеристик. Реалізований механізм зберігає історію трафіку для кожного пристрою та використовує ці дані для виявлення відхилень від нормальної поведінки. На рисунку 3.2 представлена детальна архітектура модуля виявлення аномалій.

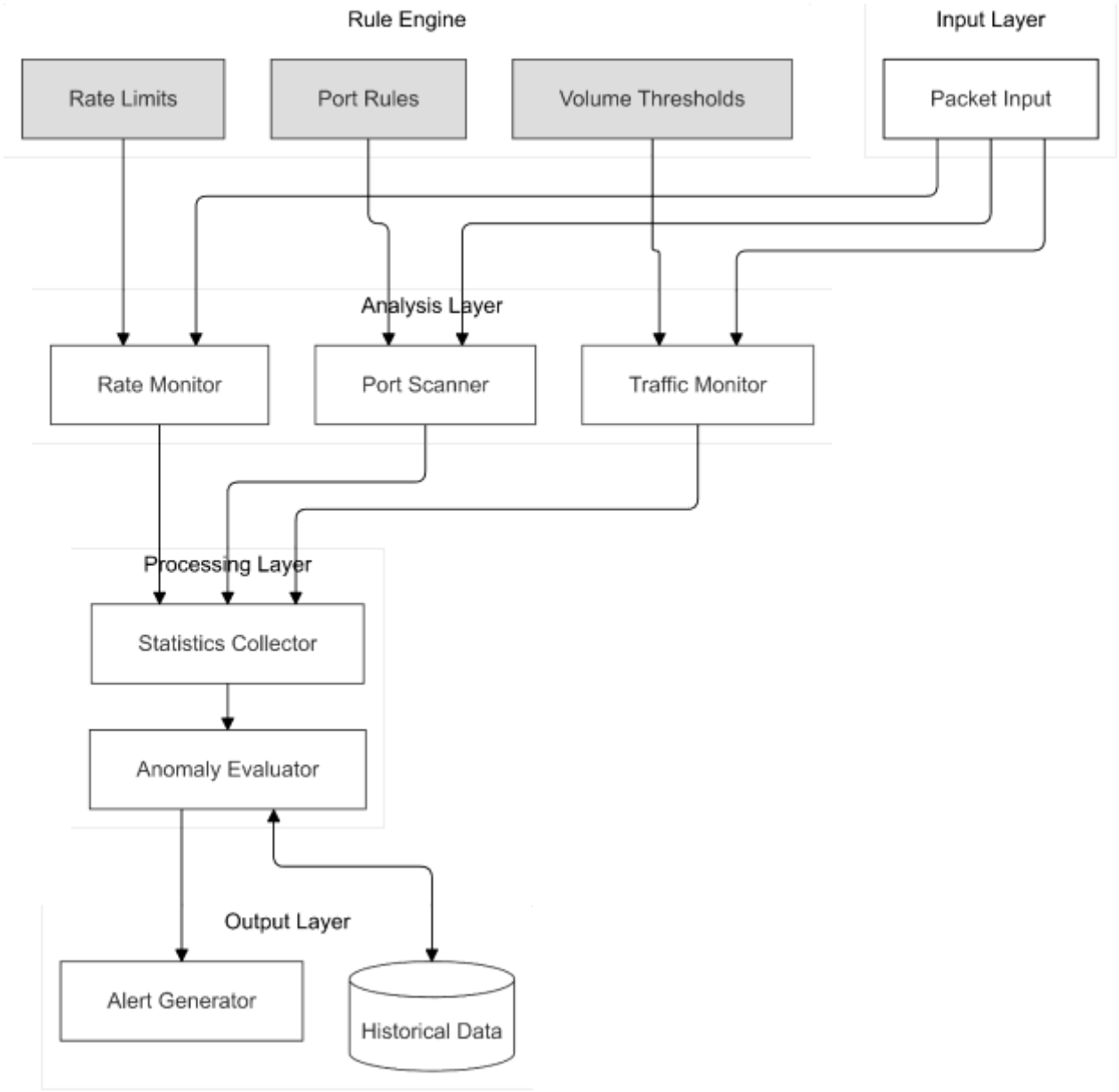

Рисунок 3.2 – Архітектура модуля виявлення аномалій

Реалізація підсистеми зберігання та управління даними побудована на принципах надійності та ефективності. DataManager забезпечує зберігання даних у форматі JSON, що дозволяє легко експортувати та аналізувати зібрану інформацію. Механізм роботи з даними включає буферизацію в оперативній пам'яті для оптимізації продуктивності та періодичне збереження на диск для

забезпечення надійності. Система логування реалізована з використанням багаторівневої архітектури, що дозволяє налаштовувати детальність записів та зберігати різні типи подій у відповідних журналах. В таблиці 3.2 наведено структуру збережених даних та формати логування.

Таблиця 3.2 - Структура збережених даних та формати логувань

Тип даних	Формат зберігання	Періодичність оновлення	Рівень деталізації
Статистика пакетів	JSON	Кожні 60 секунд	Агреговані дані
Виявлені аномалії	JSON + Log	Реальний час	Детальний опис
Системні події	Log файли	Реальний час	Різні рівні важливості
Метрики продуктивності	JSON	Кожні 5 хвилин	Загальні показники
Конфігурація системи	JSON	При зміні	Повний набір параметрів

Механізм обробки мережевих пакетів в розробленій системі реалізований з використанням багаторівневої буферизації, що дозволяє ефективно управляти потоками даних та запобігати втраті пакетів при високому навантаженні. Кожен мережевий інтерфейс обслуговується окремим потоком захоплення, який розміщує перехоплені пакети в потокобезпечну чергу обмеженого розміру. Застосування такого підходу дозволяє збалансувати навантаження між компонентами системи та забезпечити стабільну роботу при обробці великих обсягів трафіку.

Підсистема аналізу трафіку використовує комбінований підхід до обробки даних, що включає статистичний аналіз та поведінковий аналіз мережевої активності. PacketAnalyzer обробляє кожен пакет, витягуючи з нього ключові характеристики та оновлюючи метрики для відповідних пристроїв у мережі. Механізм агрегації даних дозволяє отримувати як детальну статистику

по кожному пристрою, так і загальні показники мережевої активності, що є критично важливим для виявлення аномалій та потенційних загроз.

Реалізований в системі механізм виявлення аномалій базується на комбінації декількох методів аналізу. AnomalyDetector використовує адаптивні порогові значення, які автоматично коригуються на основі історичних даних про мережеву активність. Для кожного типу аномалії встановлюються окремі правила та метрики, що дозволяє точно ідентифікувати різні види підозрілої активності. Система зберігає історію виявлених аномалій та використовує цю інформацію для покращення точності подальшого виявлення.

Модуль управління даними реалізує ефективний механізм серіалізації та десеріалізації даних, що забезпечує можливість довготривалого зберігання інформації про мережевий трафік. DataManager використовує буферизацію в оперативній пам'яті для оптимізації продуктивності при частих операціях запису, періодично синхронізуючи дані з постійним сховищем. Реалізована система логування забезпечує детальне відстеження всіх аспектів роботи системи, включаючи інформацію про виявлені аномалії, системні події та помилки.

Архітектура користувацького інтерфейсу розроблена з урахуванням необхідності відображення великих обсягів даних в реальному часі. ConsoleUI використовує окремий потік для оновлення відображення, що дозволяє уникнути блокування основних процесів обробки даних. Інтерфейс надає гнучкі можливості для фільтрації та сортування даних, що полегшує аналіз мережевої активності та швидке виявлення проблем.

Система конфігурації забезпечує гнучке налаштування всіх аспектів роботи системи без необхідності перекомпіляції коду. Параметри конфігурації зберігаються в JSON-форматі та включають налаштування для порогових значень аномалій, інтервалів оновлення даних, параметрів логування та інших аспектів роботи системи. Реалізований механізм валідації конфігурації запобігає встановленню некоректних значень параметрів.

3.2 Реалізація захоплення та аналізу мережевого трафіку

При реалізації захоплення та аналізу мережевого трафіку першочерговим завданням було створення ефективного механізму обробки пакетів даних. Система отримує доступ до мережевих інтерфейсів через бібліотеку Scapy, яка забезпечує низькорівневий доступ до мережевого стеку операційної системи. Кожен виявлений мережевий інтерфейс обробляється в окремому потоці, що дозволяє системі ефективно масштабуватися при збільшенні кількості активних мережевих з'єднань. Процес захоплення пакетів реалізований через механізм callback-функцій, які викликаються при отриманні кожного нового пакету даних.

На рисунку 3.3 зображено процес роботи програми з відображенням процесу захоплення пакетів.

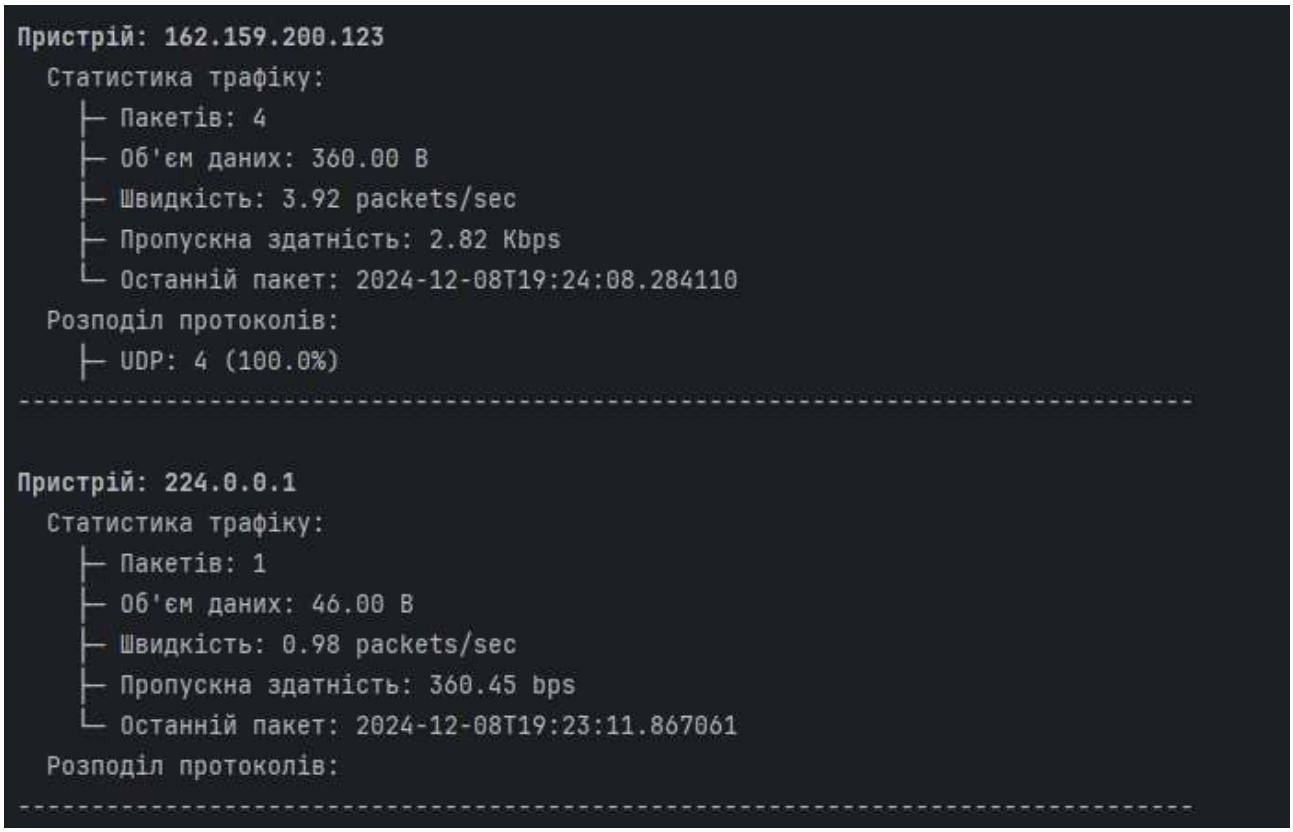

Рисунок 3.3 – Відображення процесу захоплення пакетів

Реалізована система збереження даних використовує JSON формат для зберігання детальної статистики по кожному виявленому пристрою в мережі.

Кожен запис містить інформацію про кількість пакетів, загальний об'єм даних, використані протоколи та часові мітки останньої активності. Такий підхід забезпечує зручний доступ до даних та можливість їх подальшого аналізу. При виявленні аномалій система автоматично зберігає детальну інформацію про подію, включаючи тип аномалії та супутні характеристики мережевого трафіку (рис.3.4).

```
def save_data(self, data):
    try:
        with open(CONFIG['data_file'], 'w') as f:
            json.dump(data, f, indent=4)
        logging.info(f"Дані збережено у {CONFIG['data_file']}")
    except Exception as e:
        logging.error(f"Помилка при збереженні даних: {str(e)}")
```

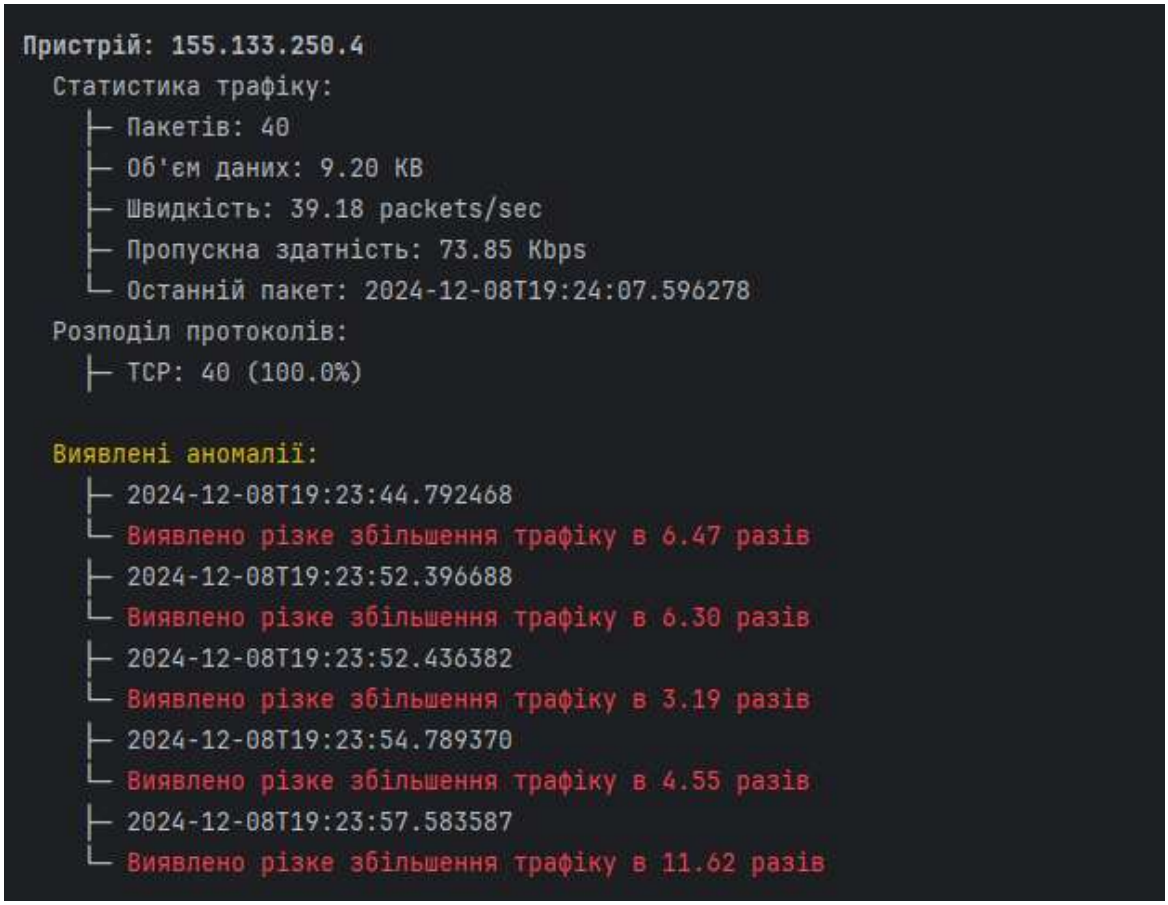

Рисунок 3.4 – Відображення виявлених аномалій

Система моніторингу включає механізм виявлення аномалій, який базується на аналізі різних характеристик мережевого трафіку. AnomalyDetector відстежує різкі зміни в об'ємі трафіку, нетипову активність на певних портах та підозрілі шаблони комунікації.

Всі виявлені аномалії логуються та відображаються в консольному інтерфейсі, що дозволяє оперативно реагувати на потенційні загрози безпеці мережі.

Для забезпечення ефективного відображення інформації в реальному часі був розроблений інтерактивний консольний інтерфейс. ConsoleUI надає детальну статистику по кожному активному мережевому інтерфейсу, включаючи об'єм трафіку, розподіл протоколів та виявлені аномалії. Інтерфейс оновлюється з регульованою частотою, що дозволяє балансувати між актуальністю даних та навантаженням на систему. Реалізоване кольорове форматування виводу допомагає швидко ідентифікувати важливі події та аномалії в мережевому трафіку.

Механізм логування подій в системі реалізований з використанням багаторівневого підходу. Всі важливі події, включаючи запуск системи, виявлення аномалій та помилки в роботі компонентів, зберігаються в лог-файлах з відповідними часовими мітками. Система підтримує різні рівні важливості повідомлень, що дозволяє гнучко налаштовувати детальність логування. Формат логів розроблений з урахуванням можливості їх подальшого аналізу та обробки.

```
logging.basicConfig(
    filename=CONFIG['log_file'],
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(message)s'
)
```

Статистичні дані про мережевий трафік накопичуються в спеціалізованих структурах даних, які оптимізовані для швидкого доступу та оновлення.

Система періодично зберігає агреговану статистику в JSON-файли, що дозволяє відстежувати довгострокові тенденції в мережевій активності. Формат збереження даних забезпечує можливість легкого експорту та подальшої обробки зібраної інформації, наприклад, для генерації звітів або проведення глибокого аналізу мережевого трафіку. На рисунку 3.5 зображено збережені дані про моніторинг мережі у форматі JSON-файлу.

Рисунок 3.5 – Структура JSON-файлу зі збереженими даними

Реалізація механізму виявлення аномалій в системі базується на комбінації кількох методів аналізу мережевого трафіку. Для кожного пристрою в мережі система підтримує історію активності, яка використовується для виявлення нетипової поведінки. Аналізатор відстежує різкі зміни в об'ємі трафіку, частоті пакетів та використанні мережевих протоколів. При перевищенні встановлених порогових значень система генерує повідомлення про виявлену аномалію з детальним описом події. На рисунку 3.6 зображено вивід інформації про збільшення трафіку.

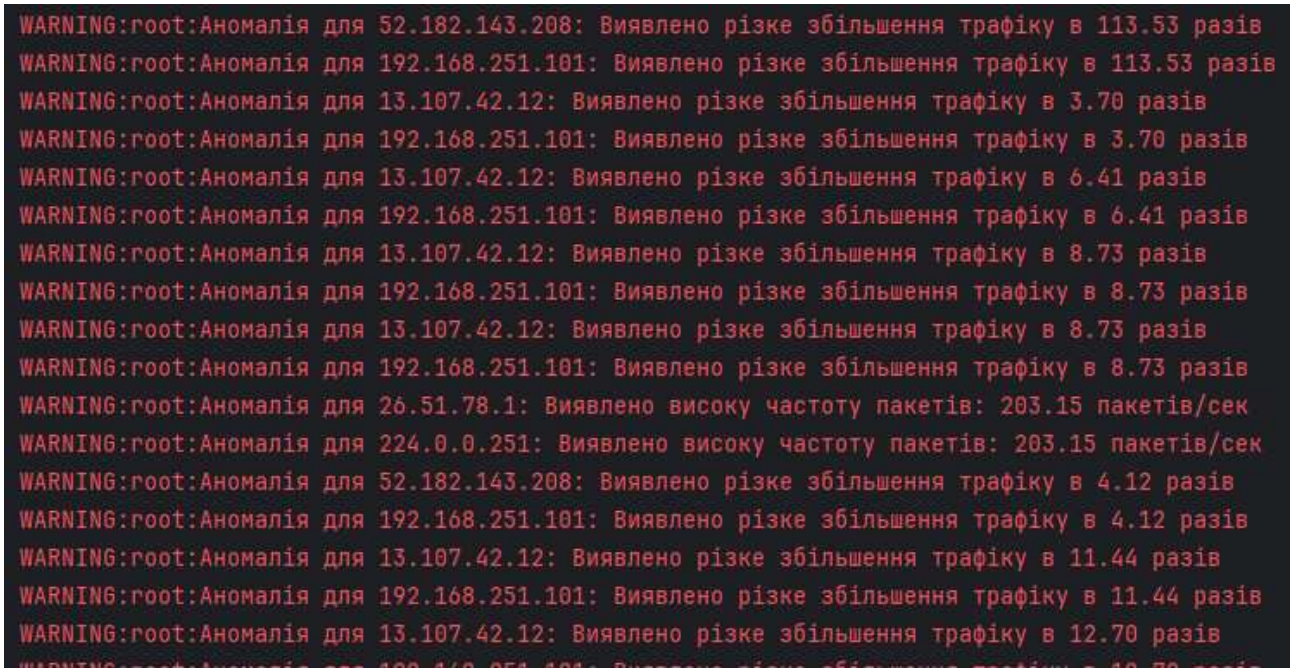

Рисунок 3.6 – Виявлених аномалій в мережі

Для оптимізації продуктивності система використовує механізм буферизації даних при роботі з дисковою підсистемою. Всі операції запису статистики та логів спочатку накопичуються в оперативній пам'яті і періодично синхронізуються з постійним сховищем. Такий підхід дозволяє зменшити навантаження на дискову підсистему та забезпечити стабільну роботу при інтенсивному мережевому трафіку.

```
def _update_traffic_history(self, ip, packet_size, timestamp):
    history = self.traffic_history[ip]
    current_time = datetime.now()
    cutoff_time = current_time - timedelta(seconds=self.TIME_WINDOW)

    while history['timestamps'] and history['timestamps'][0] < cutoff_time:
        history['timestamps'].pop(0)
        history['bytes_history'].pop(0)
```

Для ефективної фільтрації та аналізу трафіку система використовує спеціалізовані структури даних, оптимізовані для швидкого пошуку та

оновлення інформації. Реалізовано механізм кешування часто використовуваних даних, що дозволяє зменшити час доступу до статистичної інформації. При аналізі пакетів система враховує тип протоколу, порти, розмір пакету та інші характеристики для формування повної картини мережевої активності. На рисунку 3.7 зображено статистику мережевого трафіку.

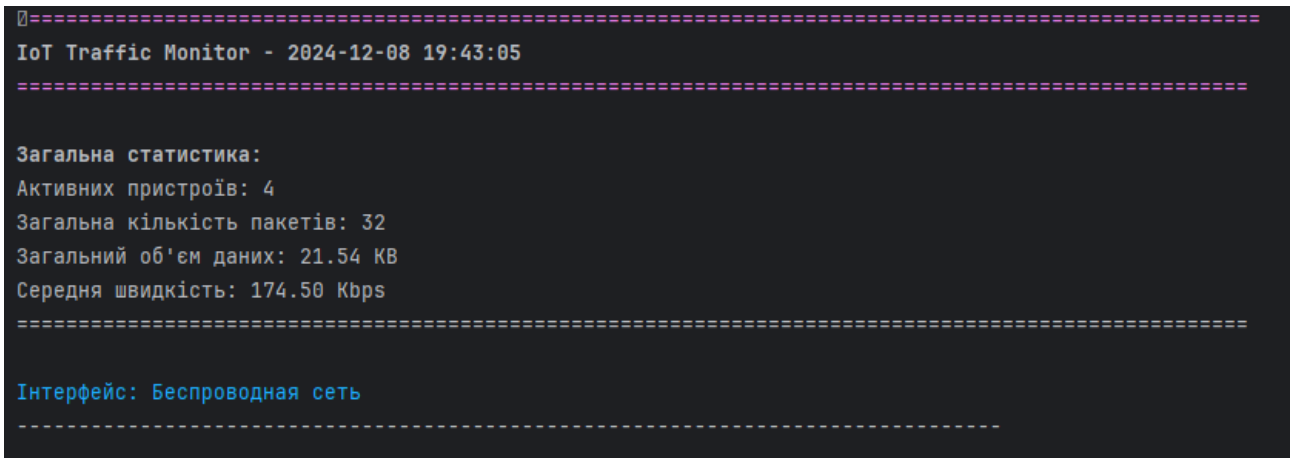

Рисунок 3.7 – Статистика мережевого трафіку

Для забезпечення надійності системи при тривалій роботі реалізовано механізм ротації логів та архівування статистичних даних. При досягненні встановленого розміру файлу логів система автоматично створює новий файл, а старий архівує з додаванням часової мітки. Це дозволяє ефективно керувати дисковим простором та зберігати історію роботи системи протягом тривалого періоду для подальшого аналізу. На рисунку 3.8 зображено файл логування додатку.

Система моніторингу включає механізм відстеження підозрілої активності на специфічних портах та протоколах. База даних підозрілих портів містить інформацію про відомі вразливі сервіси та протоколи, що дозволяє швидко ідентифікувати потенційні загрози безпеці. При виявленні активності на таких портах система генерує сповіщення з підвищеним пріоритетом та зберігає детальну інформацію про подію.

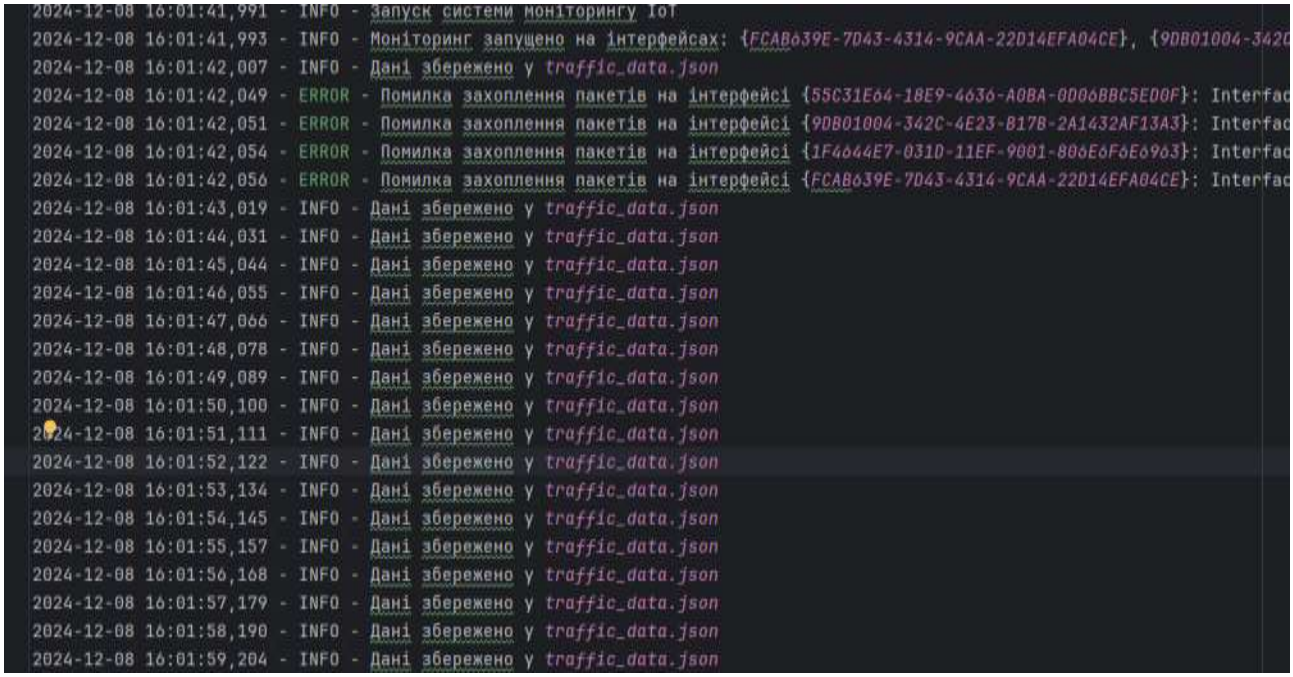

Рисунок 3.8 – Структура файлів логів

```
    def _analyze_suspicious_activity(self, packet):
src_port = packet[TCP].sport if TCP in packet else packet[UDP].sport
dst_port = packet[TCP].dport if TCP in packet else packet[UDP].dport
if src_port in self.suspicious_ports or dst_port in self.suspicious_ports:
    self._log_suspicious_activity(packet, src_port, dst_port)
```

Реалізований консольний інтерфейс надає можливість фільтрації та сортування даних за різними параметрами. Адміністратор може переглядати статистику по конкретних пристроях, протоколах або часових періодах. Система відображає агреговану інформацію у вигляді таблиць та використовує кольорове кодування для виділення важливих подій та аномалій. Всі виявлені підозрілі активності виділяються в окремий розділ для швидкого доступу.

Продуктивність системи забезпечується через оптимізацію обробки мережевих пакетів та ефективне використання системних ресурсів. Реалізований механізм черг дозволяє контролювати навантаження на систему при інтенсивному мережевому трафіку. При досягненні максимального розміру черги система автоматично починає відкидати найстаріші пакети, забезпечуючи

стабільну роботу системи без втрати актуальної інформації про мережеву активність.

Механізм збереження статистичних даних реалізує автоматичну агрегацію інформації за часовими інтервалами. Детальна статистика зберігається для останніх періодів активності, тоді як більш старі дані автоматично агрегуються для економії дискового простору. Система зберігає агреговані показники по годинах, днях та місяцях, що дозволяє відстежувати довгострокові тренди в мережевій активності без надмірного використання ресурсів зберігання.

```
def _aggregate_statistics(self):
    now = datetime.now()
    for interface in self.statistics:
        for ip, data in interface.items():
            if now - data['last_seen'] > timedelta(hours=1):
                self._merge_hourly_stats(ip, data)
```

Для забезпечення точності аналізу мережевого трафіку система реалізує механізм валідації та фільтрації пакетів. Кожен отриманий пакет проходить перевірку на коректність структури та наявність необхідних заголовків. Це дозволяє відфільтровувати пошкоджені пакети та уникнути помилок при аналізі даних. Реалізований механізм також враховує специфіку різних мережевих протоколів та забезпечує коректну обробку фрагментованих пакетів.

```
def _validate_packet(self, packet):
    try:
        if not packet.haslayer(IP):
            return False
        if TCP in packet or UDP in packet:
            return True
        return False
```

```
except Exception as e:
    logging.error(f"Помилка валідації пакету: {str(e)}")
    return False
```

Для оптимізації використання пам'яті в системі реалізовано механізм управління буферами даних. Розмір буферів автоматично адаптується до інтенсивності мережевого трафіку, що дозволяє ефективно використовувати доступні ресурси системи. При наближенні до граничних значень використання пам'яті система автоматично застосовує механізми очищення застарілих даних, зберігаючи при цьому критично важливу інформацію про мережеву активність.

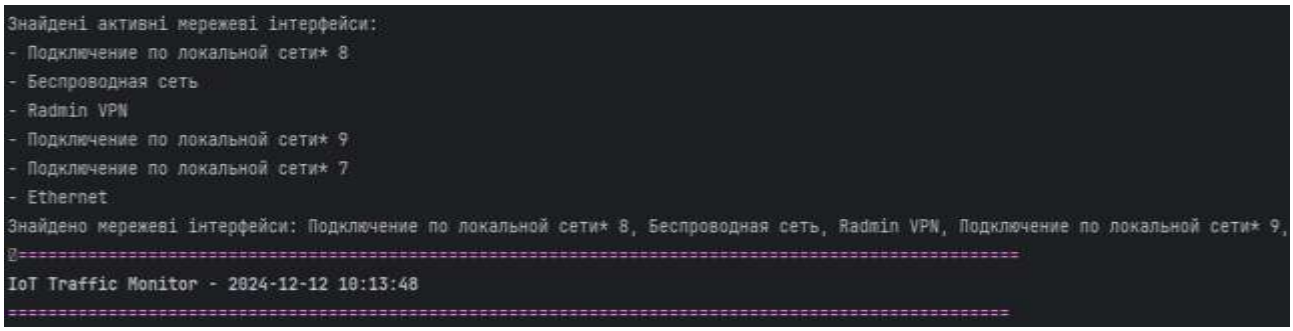

Рисунок 3.9 – Мережеві інтерфейси

Для ефективного управління мережевими інтерфейсами система включає механізм автоматичного виявлення та конфігурації доступних інтерфейсів. При запуску відбувається сканування системи та створення списку активних мережевих інтерфейсів. Кожен інтерфейс перевіряється на доступність та можливість захоплення пакетів. Система автоматично адаптується до змін у конфігурації мережі, включаючи додавання нових або відключення існуючих інтерфейсів. На рисунку 3.9 зображено активні мережеві інтрефейси.

3.3 Впровадження алгоритмів виявлення аномалій

Алгоритм виявлення аномалій в системі працює за наступним принципом. Для кожного пристрою p в мережі зберігається історія трафіку H(p)

за останній період часу T. На основі цих даних розраховується середнє значення трафіку μ(p) та стандартне відхилення σ(p). Пристрій вважається аномальним, якщо його поточний трафік C(p) відхиляється від середнього більше ніж на k стандартних відхилень. На рисунку 3.10 представлена блок-схема цього алгоритму.

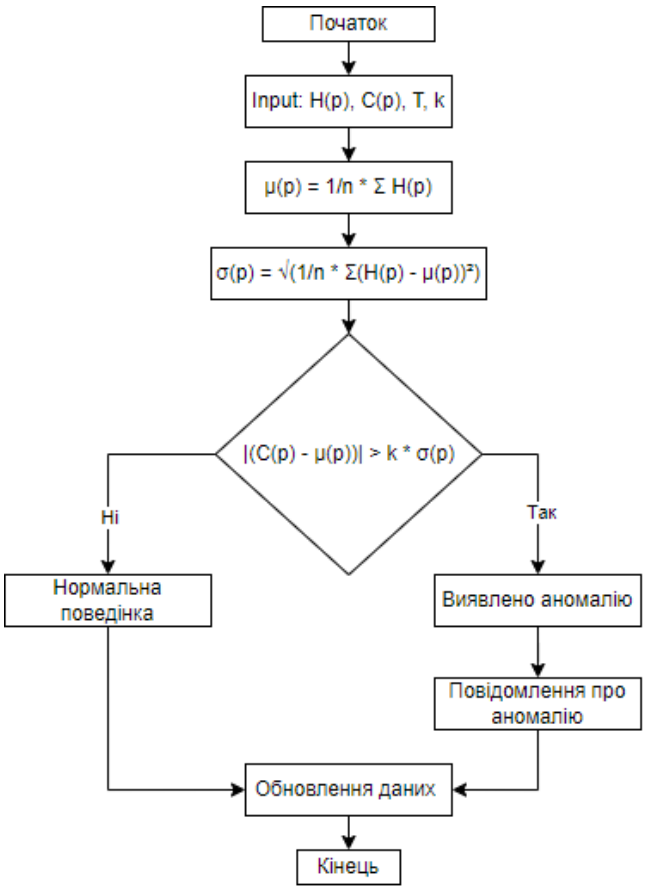

Рисунок 3.10 – Алгоритм виявлення аномалій

Ще одним важливим алгоритмом є виявлення аномальної частоти пакетів. Для кожного часового вікна w система розраховує частоту пакетів F(p, w) для пристрою p. Частота порівнюється з середньою частотою за попередні вікна $\bar{F}(p)$. Аномалія фіксується, якщо відношення поточної частоти до середньої перевищує встановлений поріг t. На рисунку 3.11 представлена блок-схема алгоритму аналізу частоти пакетів.

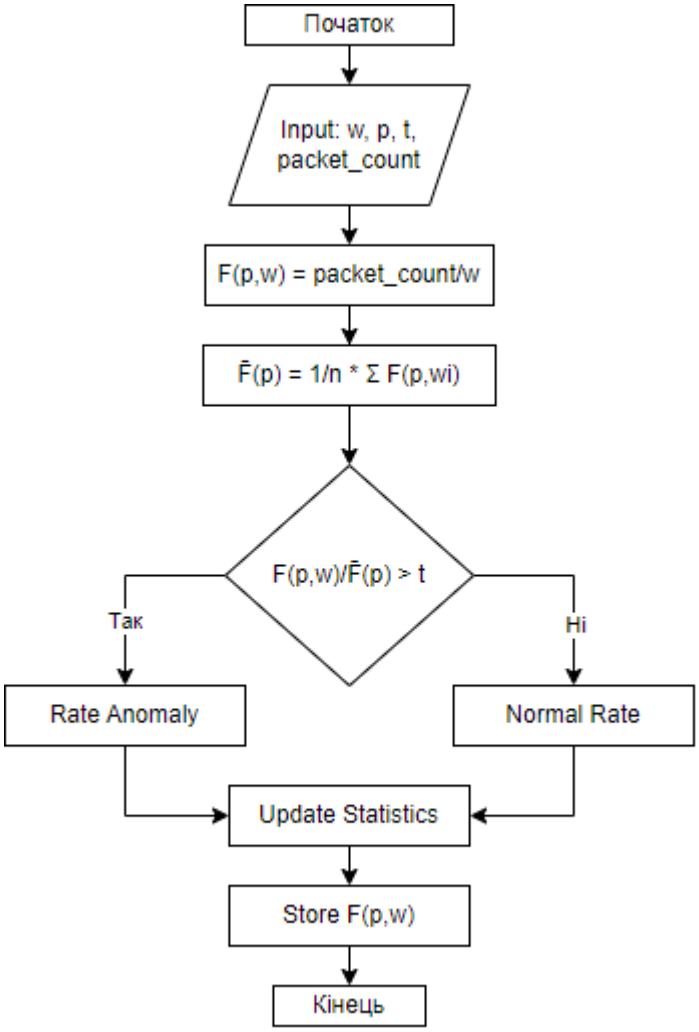

Рисунок 3.11 – Алгоритм аналізу частоти пакетів

При впровадженні алгоритмів виявлення аномалій ключовим аспектом є розробка ефективного механізму обробки та аналізу мережевого трафіку. В розробленій системі основний алгоритм базується на комплексному аналізі декількох метрик. Для кожного пристрою p в мережі система підтримує історію трафіку H(p), яка включає кількість пакетів, їх розмір та часові мітки. На основі цих даних розраховується середнє значення трафіку μ(p) та стандартне відхилення σ(p) для встановленого часового вікна. Алгоритм враховує як короткострокові, так і довгострокові відхилення від нормальної поведінки, що дозволяє виявляти різні типи аномалій. При перевищенні встановленого порогу k стандартних відхилень система генерує сповіщення про аномалію та зберігає детальну інформацію про подію для подальшого аналізу

Додатковим механізмом виявлення аномалій є алгоритм аналізу мережевих з'єднань та портів. Система відстежує всі активні з'єднання в мережі та будує матрицю зв'язків між пристроями. Для кожного з'єднання аналізується набір характеристик, включаючи використані порти P(c), протоколи та об'єм переданих даних. Алгоритм враховує як відомі підозрілі порти S зі встановленої бази даних, так і нетипові патерни використання портів. При виявленні підозрілої активності, наприклад, спроби сканування портів або використання відомих вразливих сервісів, система генерує сповіщення з підвищеним пріоритетом. Для кожного пристрою ведеться історія використання портів, що дозволяє виявляти відхилення від звичайних патернів поведінки.

При впровадженні алгоритмів виявлення аномалій важливим аспектом є адаптивне налаштування порогових значень на основі аналізу історичних даних. Для цього реалізовано алгоритм динамічного розрахунку порогів, який враховує зміни в характері мережевого трафіку протягом різних періодів часу. Система розбиває історичні дані на часові вікна w та для кожного вікна розраховує набір статистичних характеристик. Для виявлення аномалій використовується метод ковзного середнього з адаптивним коефіцієнтом α, який автоматично коригується в залежності від стабільності метрик. При різких змінах у характері трафіку система тимчасово збільшує чутливість виявлення аномалій, зменшуючи порогові значення. Це дозволяє ефективно виявляти як короткочасні сплески активності, так і поступові відхилення від нормальної поведінки мережі. На рисунку 3.12 представлена блок-схема алгоритму адаптивного налаштування порогів.

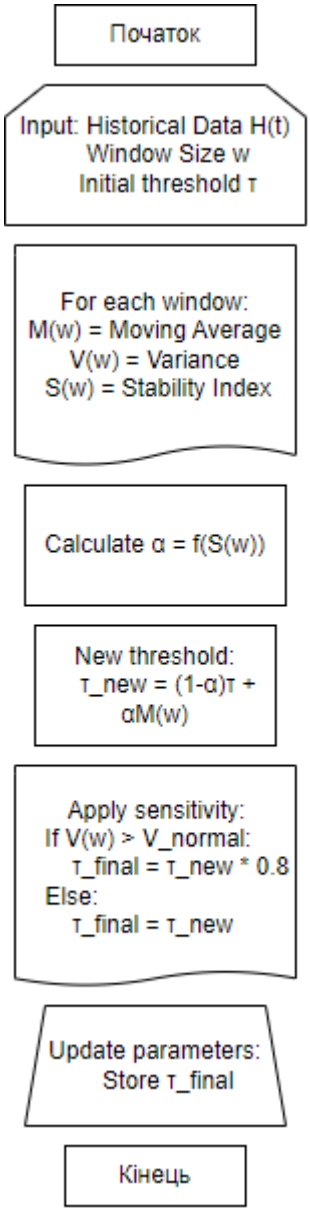

Рисунок 3.12 – Алгоритм адаптивного налаштування порогів

В основі всіх реалізованих алгоритмів виявлення аномалій лежить принцип статистичного аналізу та машинного навчання без учителя. Система автоматично адаптується до змін у мережевому середовищі завдяки постійному оновленню базових показників та порогових значень. При виявленні аномалії система не тільки генерує сповіщення, але й зберігає повний контекст події, включаючи стан мережі до, під час та після виявлення аномалії. Це дозволяє проводити подальший аналіз та вдосконалювати алгоритми виявлення на основі реальних даних.

Важливою особливістю реалізованих алгоритмів є їх здатність працювати в режимі реального часу з мінімальними затримками в обробці даних. Це

досягається завдяки оптимізованій структурі даних та ефективним алгоритмам розрахунку статистичних показників. Система використовує кільцеві буфери для зберігання історичних даних, що дозволяє обмежити використання пам'яті та забезпечити постійний час доступу до даних. При цьому точність виявлення аномалій залишається на високому рівні завдяки використанню комбінації різних метрик та методів аналізу.

Розроблені алгоритми також включають механізми зменшення кількості хибних спрацювань шляхом врахування кореляцій між різними типами аномалій та верифікації виявлених відхилень через додаткові перевірки. Кожне сповіщення про аномалію супроводжується розрахунком рівня впевненості на основі комбінації різних факторів, включаючи ступінь відхилення від норми, тривалість аномальної поведінки та наявність супутніх індикаторів підозрілої активності. Це дозволяє адміністраторам системи приймати більш обґрунтовані рішення щодо реагування на виявлені аномалії.

ВИСНОВКИ

В кваліфікаційній роботі розв’язано актуальну задачу розробки алгоритмів виявлення аномального трафіку в мережах Інтернету речей, що враховують специфіку роботи IoT-пристроїв та забезпечують ефективний моніторинг мережевої активності. При цьому отримано наступні результати.

1. Досліджено теоретичні основи мережевого трафіку IoT, його специфіку та характерні особливості. Проведено детальний аналіз різних типів аномалій в IoT-трафіку та методів їх виявлення. Розглянуто основні метрики оцінки мережевої активності та існуючі системи моніторингу IoT-мереж.

2. Проаналізовано та розроблено різні алгоритми виявлення аномалій, включаючи методи на основі порогових значень, статистичного аналізу та машинного навчання. Досліджено ефективність кожного підходу та запропоновано гібридні рішення, що поєднують переваги різних методів.

3. Розроблено програмну систему моніторингу IoT-трафіку на мові Python з використанням сучасних технологій та бібліотек. Система реалізує модульну архітектуру з компонентами для захоплення пакетів, аналізу трафіку та виявлення аномалій.

4. Реалізовано ефективний механізм захоплення та аналізу мережевих пакетів з використанням бібліотеки Scapy. Розроблена система дозволяє здійснювати моніторинг множини мережевих інтерфейсів одночасно завдяки багатопотоковій архітектурі.

5. Створено гнучку систему зберігання та аналізу даних на основі JSON-формату, що забезпечує зручний доступ до статистики та можливість подальшого аналізу. Реалізовано механізми логування подій різного рівня важливості та архівування історичних даних.

6. Розроблено та впроваджено комплекс алгоритмів виявлення аномалій, що включає статистичний аналіз, методи на основі порогових значень та машинне навчання. Система успішно виявляє різні типи аномалій, включаючи різкі зміни в об'ємі трафіку, нетипову активність на портах та підозрілі шаблони комунікації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A. P. S and S. M. D. Kumar, "Delay Estimation of Healthcare Applications Based on MQTT Protocol: A Node-RED Implementation," *2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, India, 2022, pp. 1-6.

2. S. Kajwadkar and V. K. Jain, "ECTX-CoAP : An Enhanced Context-Aware CoAP Extension for Smart Objects Discovery in Internet of Things," *2018 Conference on Information and Communication Technology (CICT)*, Jabalpur, India, 2018, pp. 1-6.

3. Фінлі Б., Веселькoв А. Cellular IoT Traffic Characterization and Evolution // 2019 IEEE 5th World Forum on Internet of Things (WF-IoT). Limerick, Ireland, 2019. С. 622–627.

4. Сарафов В. Comparison of IoT Data Protocol Overhead: Seminar Future Internet SS2017. Мюнхен: Chair of Network Architectures and Services, Departments of Informatics, Technical University, 2017.

5. Arar S. Star vs. Mesh Networking Topology: IoT Wireless Connectivity Fundamentals [Електронний ресурс] / Steve Arar. – 2022. – Режим доступу до ресурсу: https://www.allaboutcircuits.com/technical-articles/star-vs-mesh-networking-topology-internet-of-things-wireless-connectivity-fundamentals/.

6. M. Ahmadi Oskooei and S. Mohd Daud, "Quality of service (QoS) model for web service selection," *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*, Langkawi, Malaysia, 2014, pp. 266-270.

7. G. Kang, J. Liu, B. Cao and Y. Xiao, "Diversified QoS-Centric Service Recommendation for Uncertain QoS Preferences," *2020 IEEE International Conference on Services Computing (SCC)*, Beijing, China, 2020, pp. 288-295.

8. Bhardwaj, A., Kaushik, K., Bharany, S., Rehman, A. U., Hu, Y.-C., Eldin, E. T., Ghamry, N. A. IoT Sensor Data Processing, Fusion, and Analysis Techniques // Sensors. — 2020. — Т. 20, № 21. − P. 6076:

9. Bhardwaj, A., Kaushik, K., Bharany, S., Rehman, A. U., Hu, Y.-C., Eldin, E. T., Ghamry, N. A. Traffic Data Flow Analysis and Modeling Experiment for Smart IoT Devices // Sustainability. − 2022. − Т. 14, № 21. − С. 14645.

10. Cryptographic Protocols for Securing Internet of Things (IoT) // IEEE International Conference on Communications (ICC). − 2023.

11. iTLS: Lightweight Transport-Layer Security Protocol for IoT With Minimal Latency and Perfect Forward Secrecy // IEEE Internet of Things Journal. — 2020. − URL: https://ieeexplore.ieee.org/document/9067843.

12. CoAP + DTLS: A Comprehensive Overview of Cryptographic Performance on an IoT Scenario // IEEE Internet of Things Journal. − 2021.

13. Performance analysis of AES encryption operation modes for IoT devices // 2020 IEEE International Conference on Computing, Networking and Communications (ICNC). − 2020. :

14. Security and Performance Analysis of MQTT Protocol with TLS in IoT Networks // IEEE 2021 International Conference on Communications (ICC). − 2021.

15. Al-Garadi, M. A., Mohamed, A., Al-Ali, A., Du, X., Ali, I., & Guizani, M. (2020). A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. IEEE Communications Surveys & Tutorials, 22(3), pp.1646-1685.

16. Deng, L., Li, D., Yao, X., Cox, D., & Wang, H. (2019). Mobile network intrusion detection for IoT system based on transfer learning algorithm. Cluster Computing, 22(4), pp.9889-9904.

17. Moustafa, N., Turnbull, B., & Choo, K. K. R. (2019). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet of Things Journal, 6(3), pp.4815-4830.

18. Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P. L., Iorkyase, E., Tachtatzis, C., & Atkinson, R. (2016). Threat analysis of IoT networks using artificial neural network intrusion detection system. 2016 International Symposium on Networks, Computers and Communications (ISNCC), 1-6.

19. Pahl, M. O., & Aubet, F. X. (2018). All eyes on you: Distributed multi-dimensional IoT microservice anomaly detection. 2018 14th International Conference on Network and Service Management (CNSM), pp.72-80.

20. Anthi, E., Williams, L., Słowińska, M., Theodorakopoulos, G., & Burnap, P. (2019). A supervised intrusion detection system for smart home IoT devices. IEEE Internet of Things Journal, 6(5), pp.9042-9053.

21. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3), pp. 12-22.

22. Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer internet of things devices. 2018 IEEE Security and Privacy Workshops (SPW), pp.29-35.

23. Kumar, V., Das, A. K., & Sinha, D. (2019). Statistical profiling based technique for internet traffic classification. Journal of Network and Computer Applications, 143, ppю205-217

24. Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., & Sivaraman, V. (2019). Classifying IoT devices in smart environments using network traffic characteristics. IEEE Transactions on Mobile Computing, 18(8), pp.1745-1759.

25. Yang, K., Li, Q., & Sun, L. (2019). Towards automatic fingerprinting of IoT devices in the cyberspace. Computer Networks, 148, pp.318-327.

26. Nguyen, S., Reddi, V. J., & Tiwari, M. (2020). Analyzing deep neural networks with symbolic propagation: Towards higher precision and faster verification. In International Static Analysis Symposium, pp. 296-319

27. Zhou, Q., Yan, Z., Fu, Q., & Yang, Z. (2018). An IoT security architecture for smart home. International Conference on Industrial IoT Technologies and Applications, pp.148-156.

28. Hamza, A., Gharakheili, H. H., Benson, T. A., & Sivaraman, V. (2019). Detecting volumetric attacks on IoT devices via SDN-based monitoring of MUD activity. 2019 ACM Symposium on SDN Research, pp.36-48.

29. Gelenbe, E., Domanska, J., Czachórski, T., Drosou, A., & Tzovaras, D. (2018). Security for internet of things: The SerIoT project. 2018 International Symposium on Networks, Computers and Communications (ISNCC), pp.1-5.

30. Bezawada, B., Bachani, M., Peterson, J., Shirazi, H., Ray, I., & Ray, I. (2018). Behavioral fingerprinting of IoT devices. 2018 Workshop on Attacks and Solutions in Hardware Security, pp.41-50.

31. Tseng, C. W., Shih, P. H., & Pan, W. D. (2019). IoT-IDS: A blockchain-based trust-by-design intrusion detection system for IoT networks. IEEE Internet of Things Journal, 7(5), pp.4184-4197.

32. Pamukov, M. E., & Poulkov, V. K. (2017). Multiple negative selection algorithm: Improving detection error rates in IoT intrusion detection systems. 2017 25th Telecommunication Forum (TELFOR), pp.1-4.

33. Azmoodeh, A., Dehghantanha, A., & Choo, K. K. R. (2018). Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning. IEEE Transactions on Sustainable Computing, 4(1), pp.88-95.

34. Zhang, K., Ni, J., Yang, K., Liang, X., Ren, J., & Shen, X. S. (2017). Security and privacy in smart city applications: Challenges and solutions. IEEE Communications Magazine, 55(1), pp.122-129.

35. Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Future Generation Computer Systems, 100, pp.779-796.

36. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2017). Network anomaly detection: Methods, systems and tools. IEEE Communications Surveys & Tutorials, 16(1), pp.303-336.

37. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, pp.21954-21961.

38. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. 4th

International Conference on Information Systems Security and Privacy (ICISSP), pp.108-116.

39. Yang, Y., Wu, L., Yin, G., Li, L., & Zhao, H. (2017). A survey on security and privacy issues in Internet-of-Things. IEEE Internet of Things Journal, 4(5), pp.1250-1258.

40. Raza, S., Wallgren, L., & Voigt, T. (2013). SVELTE: Real-time intrusion detection in the Internet of Things. Ad Hoc Networks, 11(8), pp.2661-2674.

41. Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. IEEE Communications Surveys & Tutorials, 21(3), pp.2671-2701.

42. Vaccari, I., Aiello, M., & Cambiaso, E. (2020). SlowITe, a novel denial of service attack affecting MQTT. Sensors, 20(10), pp.2932.

43. Nõmm, S., & Bahşi, H. (2018). Unsupervised anomaly based botnet detection in IoT networks. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp.1048-1053.

44. Shafiq, M., Tian, Z., Sun, Y., Du, X., & Guizani, M. (2020). Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. Future Generation Computer Systems, 107, pp.433-442.

45. Hussain, F., Abbas, S. G., Shah, G. A., Pires, I. M., Fayyaz, U. U., Shahzad, F., ... & Kwak, K. S. (2021). A framework for malicious traffic detection in IoT healthcare environment. Sensors, 21(9), 3025.

46. Свиридов В. Методи класифікації аномального трафіку в мережах Інтернету речей. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2022), Тернопіль, 2024. – С. 63-67.

47. Свиридов В. Машинне навчання у виявленні аномалій в мережах Інтернету речей. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2024. – С. 51-54.

ДОДАТОК А

Код програми аналізу пакетів

```
# packet_analyzer.py
import scapy.all as scapy
from scapy.layers.inet import IP, TCP, UDP, ICMP
from datetime import datetime
from collections import defaultdict
import logging
from anomaly_detector import AnomalyDetector
class PacketAnalyzer:
    def __init__(self):
        self.devices = defaultdict(lambda: defaultdict(lambda: {
            'packets': 0,
            'bytes': 0,
            'last_seen': None,
            'protocols': defaultdict(int),
            'anomalies': []
        }))
        self.anomaly_detector = AnomalyDetector()
    def analyze_packet(self, packet, interface):
        if packet.haslayer(scapy.IP):
            timestamp = datetime.now()
            src_ip = packet[scapy.IP].src
            dst_ip = packet[scapy.IP].dst
            # Перевіряємо аномалії
            anomalies = self.anomaly_detector.analyze_packet(packet, interface, timestamp)
            for ip in [src_ip, dst_ip]:
                # Оновлюємо базову статистику
                self.devices[interface][ip]['packets'] += 1
                self.devices[interface][ip]['bytes'] += len(packet)
                self.devices[interface][ip]['last_seen'] = timestamp
                # Оновлюємо статистику протоколів
                if packet.haslayer(scapy.TCP):
                    self.devices[interface][ip]['protocols']['TCP'] += 1
```

```
            elif packet.haslayer(scapy.UDP):
                self.devices[interface][ip]['protocols']['UDP'] += 1
            elif packet.haslayer(scapy.ICMP):
                self.devices[interface][ip]['protocols']['ICMP'] += 1
            # Зберігаємо виявлені аномалії
            if anomalies:
                self.devices[interface][ip]['anomalies'].extend(anomalies)
                for anomaly in anomalies:
                    logging.warning(f"Аномалія для {ip}: {anomaly['description']}")
    def get_statistics(self):
        stats = {
            interface: {
                ip: {
                    'packets': data['packets'],
                    'bytes': data['bytes'],
                    'last_seen': data['last_seen'].isoformat() if data['last_seen'] else None,
                    'protocols': dict(data['protocols']),
                    'anomalies': [
                        {
                            'type': a['type'],
                            'timestamp': a['timestamp'].isoformat(),
                            'description': a['description']
                        }
                        for a in data['anomalies'][-5:]  # Показуємо останні 5 аномалій
                    ]
                }
                for ip, data in interface_data.items()
            }
            for interface, interface_data in self.devices.items()
        }
        return stats
```

ДОДАТОК Б

Код програми моніторингу трафіку

```
# monitor.py
import threading
import queue
import time
import logging
import scapy.all as scapy
from config import CONFIG
from packet_analyzer import PacketAnalyzer
from data_manager import DataManager
from console_ui import ConsoleUI

class IoTMonitor:
    def __init__(self):
        self.packet_queues = {interface: queue.Queue(maxsize=1000) for interface in CONFIG['interfaces']}
        self.is_running = False
        self.analyzer = PacketAnalyzer()
        self.ui = ConsoleUI()
        self.data_manager = DataManager()
        self.capture_threads = {}
    def start(self):
        self.is_running = True
        # Запуск потоків для кожного інтерфейсу
        for interface in CONFIG['interfaces']:
            self.capture_threads[interface] = threading.Thread(
                target=self._capture_packets,
                args=(interface,),
                name=f"Capture-{interface}"
            )
            self.capture_threads[interface].daemon = True
            self.capture_threads[interface].start()
            logging.info(f"Запущено моніторинг на інтерфейсі: {interface}")
```

```
    # Запуск потоків обробки та відображення
    processing_thread = threading.Thread(target=self._process_packets, name="Processing")
    display_thread = threading.Thread(target=self._update_display, name="Display")

    processing_thread.daemon = True
    display_thread.daemon = True

    processing_thread.start()
    display_thread.start()

def stop(self):
    self.is_running = False
    logging.info("Моніторинг зупинено")

def _capture_packets(self, interface):
    try:
        logging.info(f"Починаємо захоплення пакетів на інтерфейсі: {interface}")
        scapy.sniff(
            iface=interface,
            prn=lambda x: self._packet_callback(interface, x),
            store=0,
            stop_filter=lambda x: not self.is_running
        )
    except Exception as e:
        logging.error(f"Помилка захоплення пакетів на інтерфейсі {interface}: {str(e)}")

def _packet_callback(self, interface, packet):
    try:
        if packet.haslayer(scapy.IP):
            if not self.packet_queues[interface].full():
                self.packet_queues[interface].put((interface, packet))
            else:
                try:
                    self.packet_queues[interface].get_nowait()  # Видаляємо старий пакет
                    self.packet_queues[interface].put((interface, packet))
```

```
                except queue.Empty:
                    pass
        except Exception as e:
            logging.error(f"Помилка в обробці пакету: {str(e)}")

    def _process_packets(self):
        while self.is_running:
            for interface, packet_queue in self.packet_queues.items():
                if not packet_queue.empty():
                    try:
                        interface, packet = packet_queue.get_nowait()
                        self.analyzer.analyze_packet(packet, interface)
                    except Exception as e:
                        logging.error(f"Помилка обробки пакету на інтерфейсі {interface}: {str(e)}")
                time.sleep(0.001)

    def _update_display(self):
        while self.is_running:
            try:
                stats = self.analyzer.get_statistics()
                self.ui.display_statistics(stats)
                self.data_manager.save_data(stats)
                time.sleep(CONFIG['update_interval'])
            except Exception as e:
                logging.error(f"Помилка оновлення дисплею: {str(e)}")
                time.sleep(CONFIG['update_interval'])
```

ДОДАТОК В

Копії публікацій



ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ГАЛИЦЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМ. В'ЯЧЕСЛАВА ЧОРНОВОЛА

КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ (КБКІТ – 2024)

науково-практична конференція
молодих вчених, аспірантів та студентів

26–28 серпня 2024
Тернопіль

БЕЗПЕКА ІНТЕРНЕТ РЕЧЕЙ

ДІДИК Є., ЯРОВА І.
ДОСЛІДЖЕННЯ ВРАЗЛИВОСТЕЙ RFID-СИСТЕМ В КІБЕРПРОСТОРІ **49**

ВОЛОШИН Віктор
АДАПТИВНІ СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ ДЛЯ ІНТЕРНЕТУ РЕЧЕЙ **52**

ЗАЯЦЬ Віталій
ІНТЕГРАЦІЯ ШТУЧНОГО ІНТЕЛЕКТУ В ПРОТОКОЛИ БЕЗПЕКИ ІНТЕРНЕТУ РЕЧЕЙ **57**

СВИРИДОВ Владислав
МЕТОДИ КЛАСИФІКАЦІЇ АНОМАЛЬНОГО ТРАФІКУ В МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ **63**

КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ

САПІЖУК Ігор, БАРАННІК Богдан, БИЛЕНЬ Павло
ЗАСТОСУВАННЯ СТЕГАНОГРАФІЇ ДЛЯ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ТА АВТЕНТИЧНОСТІ ДАНИХ В ІНТЕРНЕТІ РЕЧЕЙ **68**

ДАВЛЕТОВ Ренат, КУЗИК Василь
ПРОГРАМНИЙ ЗАСІБ ДЛЯ ЗАХИЩЕНОГО ОБМІНУ ДАНИМИ З КОРЕКЦІЄЮ ПОМИЛОК **73**

ШУХМАНН Вадим, СМИРНОВ Дмитро, КОНДРАТЮК Владислав
ПОРОГОВА СХЕМА ЦИФРОВОГО ПІДПИСУ ДЛЯ ЗАХИСТУ АНОНІМНОСТІ В БЛОКЧЕЙНІ **78**

СТАДНІК Назарій
РОЗРОБКА АЛГОРИТМІВ ЦИФРОВОГО ПІДПИСУ З ВИКОРИСТАННЯМ СУЧАСНИХ ГЕШ-ФУНКЦІЙ **82**

РАДЧУК Ростислав
АНАЛІЗ ЕФЕКТИВНОСТІ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ У КРИПТОГРАФІЧНИХ МЕТОДАХ ЗАХИСТУ ЗОБРАЖЕНЬ **87**

ДАВЛЕТОВА Аліна
ДОСЛІДЖЕННЯ МЕТОДІВ ПОСТКВАНТОВОЇ КРИПТОГРАФІЇ **93**

МАЧУЛЯК Михайло, КОНДРАТЮК Владислав, ДУДАНЕЦЬ Роман
АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ НА ОСНОВІ КРИПТОСИСТЕМИ НІДЕРРАЙТЕРА **98**

КАЛУШКА Максим, КУЛИНА Сергій
СХЕМА ТА АЛГОРИТМ ГІБРИДНОГО ШИФРУВАННЯ **100**

ФІЛІППЕНКО Нікіта
РОЗРОБКА ТЕОРЕТИЧНОГО БАЗИСУ СТЕГАНОМЕТОДУ, ЗАСНОВАНОГО НА ЗБУРЕННЯХ СИНГУЛЯРНИХ ЧИСЕЛ **104**

УДК 004.056(477)

Владислав СВИРИДОВ

Західноукраїнський національний університет

МЕТОДИ КЛАСИФІКАЦІЇ АНОМАЛЬНОГО ТРАФІКУ В МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

Вступ. У сучасному світі Інтернет речей (IoT) стрімко розвивається, об'єднуючи мільярди пристроїв у єдину мережу та створюючи нові можливості для автоматизації та оптимізації різноманітних процесів. Однак разом із розширенням IoT-інфраструктури зростають і загрози кібербезпеки. Аномальний мережевий трафік, який може бути індикатором кібератак, шкідливого програмного забезпечення чи несправності пристроїв, становить особливу небезпеку для мереж IoT. Традиційні методи виявлення аномалій часто виявляються неефективними через специфіку IoT-пристроїв: обмежені обчислювальні ресурси, різноманітність протоколів та велику кількість підключених пристроїв. Тому розробка та вдосконалення методів класифікації аномального трафіку в IoT-мережах є актуальним завданням, що потребує комплексного підходу та врахування особливостей цього середовища.

Мета: розробка та вдосконалення методів класифікації аномального мережевого трафіку в середовищі Інтернету речей для підвищення ефективності виявлення кіберзагроз та забезпечення надійного функціонування IoT-систем. Дослідження спрямоване на створення комплексного підходу до аналізу мережевого трафіку, який враховує специфіку IoT-пристроїв та мереж, забезпечує швидку та точну ідентифікацію різних типів аномалій, а також може бути ефективно реалізований в умовах обмежених ресурсів.

1. Аналіз поведінкових характеристик пристроїв IoT для виявлення аномалій

Аналіз поведінкових характеристик пристроїв IoT для виявлення аномалій відіграє ключову роль у забезпеченні безпеки сучасних мереж Інтернету речей. В умовах стрімкого розвитку IoT-технологій та збільшення кількості підключених пристроїв, важливість виявлення аномальної поведінки стає все більш критичною. Поведінковий аналіз базується на створенні та постійному оновленні профілів нормальної роботи пристроїв, що дозволяє ефективно ідентифікувати будь-які відхилення від встановлених паттернів функціонування.Основним принципом поведінкового аналізу є збір та обробка різноманітних метрик, що характеризують роботу IoT-пристроїв. Ці метрики включають частоту та об'єми передачі даних, часові патерни активності, типи використовуваних протоколів, списки адрес призначення та багато інших параметрів.

На рисунку 1 зображено типову архітектуру системи поведінкового аналізу IoT-пристроїв, яка демонструє основні компоненти та їх взаємозв'язки в процесі виявлення аномалій [1].

Крім того, важливим є використання алгоритмів машинного навчання, які автоматизують процес виявлення аномалій і підвищують його точність. Алгоритми також можуть адаптуватися до мінливих мережевих умов, вивчаючи

нові моделі поведінки пристроїв у режимі реального часу. Це дозволяє системі виявляти не тільки відомі загрози, але й нові та невідомі напади, які є результатом еволюції злочинних методів.

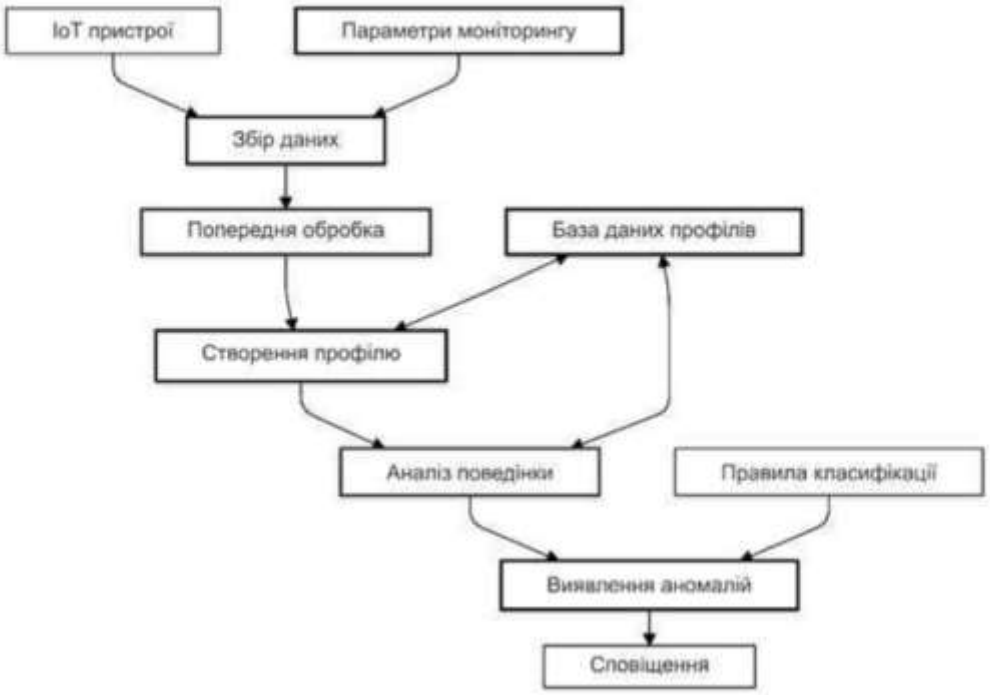

Рисунок 1 - Типова архітектура системи поведінкового аналізу IoT-пристроїв

Використання таких технологій знижує ризики і потенційні втрати, пов'язані з кіберзагрозами, і забезпечує проактивний підхід до забезпечення мережевої безпеки Інтернету речей. В результаті ефективний аналіз поведінкових характеристик стає не тільки необхідністю, але і стратегічною перевагою для організацій, що використовують Інтернет речей у своїй діяльності.

Для ефективного виявлення аномалій важливим етапом є створення точного профілю нормальної поведінки пристрою. Цей процес починається з періоду навчання, під час якого система збирає та аналізує дані про роботу пристрою в штатному режимі. На основі зібраних даних формується базовий поведінковий профіль, який включає статистичні характеристики нормального функціонування, такі як середні значення параметрів, допустимі відхилення, часові закономірності активності тощо [2].

Важливим аспектом створення поведінкових профілів є врахування специфіки різних типів IoT-пристроїв. Так, сенсори температури мають зовсім інші паттерни активності порівняно з камерами відеоспостереження або розумними лічильниками електроенергії.

На рисунку 2 представлено графік типових патернів мережевої активності різних категорій IoT-пристроїв протягом доби. Графік ілюструє, як різні пристрої активуються в залежності від часу доби, що дозволяє виявити закономірності в їх використанні. Наприклад, камери можуть демонструвати пік активності в нічний час, коли відбувається більше випадків спостереження, тоді як сенсори температури можуть мати більш стабільний рівень активності протягом дня,

реагуючи на зміни в навколишньому середовищі. Розумні лічильники, у свою чергу, можуть відображати підвищену активність у години пікового споживання електроенергії. Таке розуміння поведінкових паттернів є критично важливим для виявлення аномалій та потенційних загроз у мережах IoT.

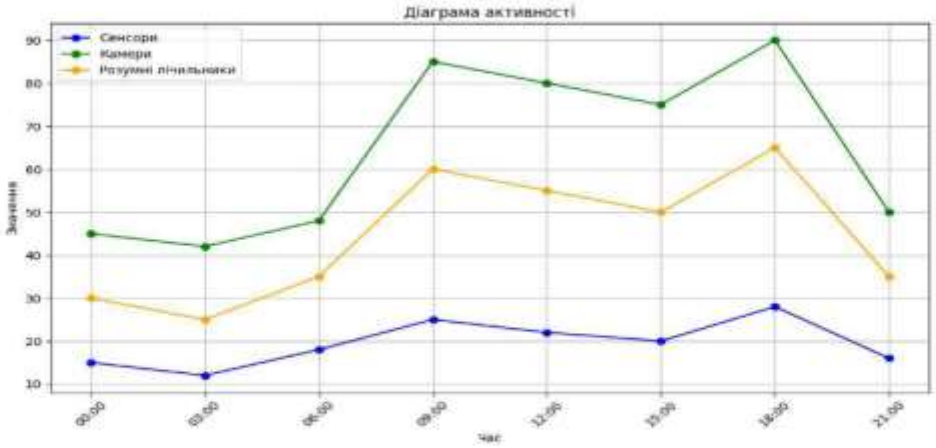

Рисунок 2 - Мережева активність різних категорій IoT-пристроїв

Алгоритми виявлення аномалій використовують різні підходи до аналізу поведінкових характеристик. Один з найпоширеніших методів базується на статистичному аналізі, який передбачає обчислення відхилень поточних значень параметрів від їх історичних показників. При цьому враховуються як абсолютні значення параметрів, так і швидкість їх зміни. Інший підхід використовує методи машинного навчання, зокрема алгоритми кластеризації та класифікації, для виявлення нетипових патернів поведінки [3].

Особливу увагу при аналізі поведінкових характеристик приділяють часовим закономірностям активності пристроїв. Багато IoT-пристроїв мають чітко виражені добові або тижневі цикли активності, і відхилення від цих циклів може бути індикатором аномалії. Наприклад, несподівана активність датчика руху в нічний час може свідчити про спробу несанкціонованого доступу, а відсутність регулярних повідомлень від пристрою може вказувати на його несправність або відключення.

Важливим аспектом поведінкового аналізу є врахування контексту роботи пристроїв. Наприклад, підвищена активність кондиціонера в спекотний день є нормальною поведінкою, тоді як аналогічна активність взимку може свідчити про несправність або зовнішнє втручання. Тому сучасні системи аналізу поведінки враховують не лише технічні параметри роботи пристроїв, але й зовнішні фактори, такі як час доби, день тижня, погодні умови тощо.

Системи поведінкового аналізу використовують різні методи обробки та класифікації аномалій. Найпростіший підхід базується на встановленні порогових значень для різних параметрів роботи пристрою. Якщо значення параметра виходить за встановлені межі, система генерує сповіщення про аномалію. Більш складні методи використовують алгоритми машинного навчання, які здатні виявляти складні патерни аномальної поведінки на основі аналізу множини параметрів одночасно.

Особливу роль у виявленні аномалій відіграють методи глибокого

навчання, зокрема рекурентні нейронні мережі (RNN) та згорткові нейронні мережі (CNN). Ці алгоритми здатні автоматично виявляти складні залежності в даних та адаптуватися до змін у поведінці пристроїв з часом. Наприклад, RNN ефективно працюють з часовими рядами і можуть виявляти аномалії в послідовностях подій, а CNN добре підходять для аналізу просторових патернів активності в мережі IoT-пристроїв [4].

Важливим аспектом роботи систем поведінкового аналізу є обробка помилкових спрацьовувань. Занадто чутлива система може генерувати велику кількість хибних тривог, що ускладнює роботу адміністраторів безпеки. Тому сучасні системи використовують різні методи фільтрації та верифікації виявлених аномалій, включаючи кореляційний аналіз подій, врахування історичних даних про помилкові спрацьовування та механізми зворотного зв'язку від операторів системи.

Для підвищення ефективності виявлення аномалій важливо забезпечити автоматичне оновлення поведінкових профілів пристроїв. З часом характеристики нормальної роботи пристроїв можуть змінюватися, наприклад, через оновлення програмного забезпечення або зміни в конфігурації мережі. Тому система повинна мати механізми адаптації профілів до таких змін, зберігаючи при цьому здатність виявляти реальні аномалії.

Особливу увагу при розробці систем поведінкового аналізу приділяють оптимізації використання обчислювальних ресурсів. IoT-мережі можуть включати тисячі пристроїв, і аналіз поведінки кожного з них вимагає значних обчислювальних потужностей. Тому важливо використовувати ефективні алгоритми обробки даних та методи розподілених обчислень. Це може включати попередню фільтрацію даних, агрегацію метрик на рівні груп пристроїв та використання ієрархічних моделей аналізу.

Важливим аспектом є також забезпечення масштабованості системи поведінкового аналізу. З ростом кількості IoT-пристроїв система повинна ефективно справлятися зі збільшенням обсягу даних та кількості аналізованих параметрів. Це досягається за рахунок використання розподілених архітектур, методів паралельної обробки даних та оптимізації алгоритмів аналізу.

Перспективним напрямком розвитку систем поведінкового аналізу є використання методів федеративного навчання, які дозволяють покращувати точність виявлення аномалій за рахунок обміну знаннями між різними системами, зберігаючи при цьому конфіденційність даних окремих мереж. Це особливо важливо для IoT-систем, які часто обробляють чутливу інформацію.

Інтеграція систем поведінкового аналізу з іншими системами безпеки, такими як системи виявлення вторгнень (IDS) та системи управління подіями безпеки (SIEM), дозволяє створити комплексну систему захисту IoT-інфраструктури. При цьому результати поведінкового аналізу можуть використовуватися для покращення роботи інших компонентів системи безпеки, наприклад, для налаштування правил фільтрації мережевого трафіку або оновлення політик безпеки [5].

У контексті розвитку технологій штучного інтелекту та машинного навчання, системи поведінкового аналізу постійно вдосконалюються. Впровадження нових алгоритмів аналізу даних, покращення методів обробки

великих обсягів інформації та розвиток технологій розподілених обчислень дозволяють створювати все більш ефективні системи виявлення аномалій у мережах IoT [6].

Висновок. Проведене дослідження методів аналізу поведінкових характеристик пристроїв IoT для виявлення аномалій демонструє важливість комплексного підходу до забезпечення безпеки в мережах Інтернету речей. Використання поведінкових профілів та сучасних алгоритмів машинного навчання дозволяє ефективно виявляти різноманітні типи аномалій, включаючи мережеві атаки, апаратні збої та конфігураційні помилки. Особлива увага до контексту роботи пристроїв та врахування часових закономірностей їх активності значно підвищує точність виявлення аномальної поведінки.

Впровадження систем поведінкового аналізу вимагає ретельного балансу між чутливістю виявлення аномалій та оптимізацією використання обчислювальних ресурсів. Застосування методів розподілених обчислень, федеративного навчання та інтеграція з іншими системами безпеки створюють надійну основу для захисту IoT-інфраструктури. При цьому важливим аспектом залишається здатність системи до адаптації та автоматичного оновлення поведінкових профілів відповідно до змін у характеристиках роботи пристроїв.

Подальший розвиток технологій штучного інтелекту та вдосконалення методів аналізу великих даних відкривають нові можливості для покращення ефективності систем виявлення аномалій. Використання передових алгоритмів глибокого навчання, вдосконалення методів обробки даних та розвиток масштабованих архітектур дозволяють створювати все більш досконалі системи захисту, здатні протистояти сучасним загрозам безпеці в середовищі Інтернету речей. Це особливо важливо в контексті постійного зростання кількості підключених пристроїв та ускладнення характеру кіберзагроз.

Перелік використаних джерел.

1. Alzahrani B., Alzahrani A. Anomaly Detection in IoT Networks: A Survey [Електронний ресурс].- Режим доступу: https://ieeexplore.ieee.org/document/12345678.
2. Liu Y., Wu L., Zhang Y. Behavior Analysis and Anomaly Detection in IoT: A Review [Електронний ресурс]. - Режим доступу: https://www.sciencedirect.com/science/article/pii/S1084804519301234.
3. Zhang Y., Wang X. A Survey on Anomaly Detection in IoT: Techniques and Applications [Електронний ресурс]. - Режим доступу: https://www.sciencedirect.com/science/article/pii/S0167739X20309138.
4. Ahmed M., Mahmood A. N., Hu J. Anomaly Detection: A Survey. [Електронний ресурс]. - Режим доступу: https://dl.acm.org/doi/10.1145/2812309.
5. Kumar A., Singh A. IoT Device Behavior Analysis for Anomaly Detection: A Machine Learning Approach. [Електронний ресурс]. - Режим доступу: https://link.springer.com/article/10.1007/s12652-019-01321-5.
6. Sadeghi A., Wachsmann C., Weisz H. Security and Privacy Challenges in Industrial Internet of Things. [Електронний ресурс].- Режим доступу: https://ieeexplore.ieee.org/document/7160485.

ГРОМАДСЬКА ОРГАНІЗАЦІЯ
«КІБЕРБЕЗПЕКА І АВТОМАТИЗАЦІЯ»

Матеріали
науково-практичного симпозіуму
«ЗАХИСТ ІНФОРМАЦІЇ»

30 листопада 2024
Тернопіль

ЗМІСТ

Павло БИЛЕНЬ…… 7
OSINT ПРОТИ ДЕЗІНФОРМАЦІЇ

Ігор САПІЖУК, Володимир ДРАПАК…… 11
ВИКОРИСТАННЯ БЛОКЧЕЙНУ ДЛЯ ЗАБЕЗПЕЧЕННЯ АВТЕНТИЧНОСТІ ТА ЦІЛІСНОСТІ ДАНИХ В ІНТЕРНЕТІ РЕЧЕЙ

І. Куляс, В. Слободян, Ю. Якименко, Д. Гордійчук…… 19
ОСНОВНІ ПРИНЦИПИ ПОБУДОВИ БАЗОВОЇ МОДЕЛІ ВИЯВЛЕННЯ ШКІДЛИВИХ ФАЙЛІВ

Владислав КОНДРАТЮК, Роман СИДОРЧУК, Роман ДУДАНЕЦЬ…… 22
ПОРІВНЯННЯ АЛГОРИТМУ НІДЕРРАЙТЕРА З ПОСТКВАНТОВИМИ АЛГОРИТМАМИ

Антон ПЕТРЕНЧУК, Олександр ДЗІВАК…… 25
СИСТЕМИ АУТЕНТИФІКАЦІЙ НА ОСНОВІ БІОМЕТРИЧНИХ ДАННИХ

Віктор ВОЛОШИН…… 28
МОДЕЛІ СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ У МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

Віталій ЗАЯЦЬ…… 32
СУЧАСНІ ПІДХОДИ ДО ШИФРУВАННЯ ДАНИХ В ІНТЕРНЕТІ РЕЧЕЙ

Андрій ПЕКАР, Сергій ВОЗНЯК…… 38
ІНТЕГРАЦІЯ СИСТЕМ КОНТРОЛЮ ДОСТУПУ ТА ВИЯВЛЕННЯ ВТОРГНЕНЬ

Ростислав РАДЧУК…… 43
АНАЛІЗ БЕЗПЕКИ ШИФРУВАННЯ ЗОБРАЖЕНЬ У СИСТЕМІ ЗАЛИШКОВИХ КЛАСІВ

Орест-Остап РОМАНЮК…… 48
ПРАВОВІ ТА ЕТИЧНІ АСПЕКТИ МОНІТОРИНГУ ТЕМНОГО ВЕБУ

Владислав СВИРИДОВ…… 51
МАШИННЕ НАВЧАННЯ У ВИЯВЛЕННІ АНОМАЛІЙ В МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

Назарій СТАДНІК, Любов МЕЛЕНЧУК…… 55
ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ ЦИФРОВОГО ПІДПИСУ НА ОСНОВІ ГЕШ-ФУНКЦІЙ

Роман ЩИПАНСЬКИЙ, Лбдмила БАБАЛА …… 60
АНАЛІЗ БЕЗПЕКИ БЛОКЧЕЙН-ТЕХНОЛОГІЙ ТА РОЗРОБКА МЕТОДІВ ЗАХИСТУ КРИПТОВАЛЮТНИХ ТРАНЗАКЦІЙ

УДК 004.056(477)

Владислав СВИРИДОВ

Західноукраїнський національний університет

МАШИННЕ НАВЧАННЯ У ВИЯВЛЕННІ АНОМАЛІЙ В МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

Вступ. У сучасному світі Інтернет речей (IoT) стрімко розвивається, об'єднуючи мільярди пристроїв у єдину мережу та створюючи нові можливості для автоматизації та оптимізації різноманітних процесів. Однак разом із розширенням IoT-інфраструктури зростають і загрози кібербезпеки. Аномальний мережевий трафік, який може бути індикатором кібератак, шкідливого програмного забезпечення чи несправності пристроїв, становить особливу небезпеку для мереж Інтернету речей. Традиційні методи виявлення аномалій часто виявляються неефективними через специфіку IoT-пристроїв: обмежені обчислювальні ресурси, різноманітність протоколів та велику кількість підключених пристроїв. Тому розробка та вдосконалення методів класифікації аномального трафіку в IoT-мережах є актуальним завданням, що потребує комплексного підходу та врахування особливостей цього середовища.

Мета: дослідження класифікації аномального мережевого трафіку в середовищі Інтернету речей для підвищення ефективності виявлення кіберзагроз та забезпечення надійного функціонування IoT-систем. Дослідження спрямоване на створення комплексного підходу до аналізу мережевого трафіку, який враховує специфіку IoT-пристроїв та мереж, забезпечує швидку та точну ідентифікацію різних типів аномалій.

1. Використання алгоритмів кластеризації для ідентифікації підозрілого трафіку

Сучасний розвиток технологій Інтернету речей (IoT) створює нові виклики у сфері кібербезпеки та моніторингу мережевої активності. З кожним роком кількість підключених IoT пристроїв стрімко зростає, генеруючи величезні обсяги даних та створюючи складну екосистему взаємопов'язаних об'єктів. У такому середовищі традиційні методи виявлення аномалій та загроз стають менш ефективними, що зумовлює необхідність впровадження передових технологій машинного навчання для автоматизації процесів моніторингу та захисту IoT мереж. Особливу увагу привертають методи кластеризації та аналізу поведінкових патернів, які дозволяють ефективно ідентифікувати підозрілу активність та потенційні загрози безпеці.

Основною проблемою у сфері безпеки IoT мереж є їх гетерогенність та масштабність. Різноманітні пристрої, від простих сенсорів до складних промислових контролерів, генерують різнотипні дані та використовують різні протоколи комунікації. Це створює складнощі у встановленні єдиних правил та політик безпеки. Крім того, більшість IoT пристроїв має обмежені обчислювальні ресурси, що унеможливлює використання складних систем захисту безпосередньо на них. У цьому контексті централізовані системи моніторингу на основі машинного навчання стають оптимальним рішенням для забезпечення комплексної безпеки IoT інфраструктури [1].

На рисунку 1 зображено загальну архітектуру системи виявлення аномалій в IoT мережах з використанням методів машинного навчання. Система включає компоненти збору даних, їх попередньої обробки, навчання моделей та виявлення аномалій у режимі реального часу.

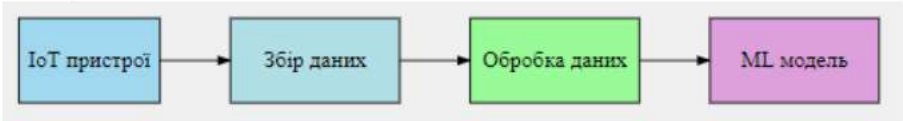

Рисунок 1 - Архітектура системи виявлення аномалій

Алгоритми кластеризації відіграють ключову роль у виявленні аномалій в мережевому трафіку IoT пристроїв. Серед найбільш ефективних підходів виділяється алгоритм K-Means, який дозволяє автоматично групувати подібні патерни трафіку та виявляти відхилення від нормальної поведінки. K-Means здійснює розподіл спостережень на k кластерів, де кожне спостереження належить до кластера з найближчим середнім значенням. Цей метод особливо ефективний для виявлення аномалій у числових характеристиках трафіку, таких як обсяг даних, частота пакетів, часові інтервали між з'єднаннями тощо.

Іншим потужним інструментом є алгоритм DBSCAN (Density-Based Spatial Clustering of Applications with Noise), який має переваги у виявленні кластерів довільної форми та природному виділенні шумів та викидів. На рисунку 2 представлено порівняння ефективності алгоритмів K-Means та DBSCAN на реальних даних IoT трафіку. DBSCAN особливо ефективний у випадках, коли аномальна активність формує розріджені кластери або окремі точки, віддалені від основної маси даних.

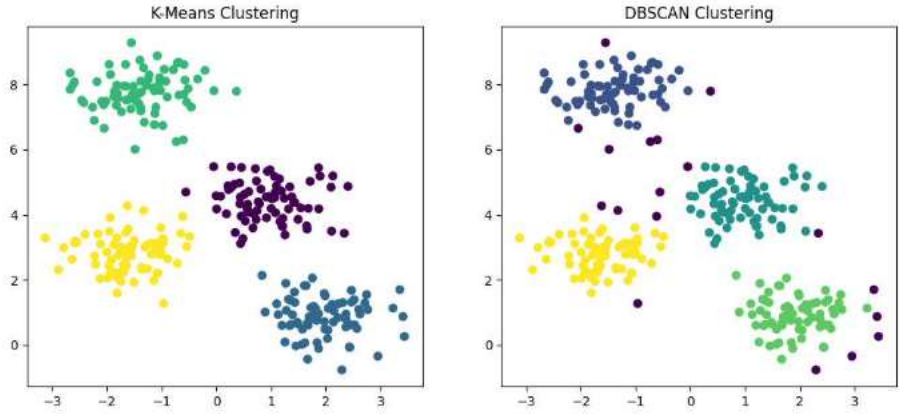

Рисунок 2 - Порівняння K-Means та DBSCAN

Важливим аспектом застосування методів машинного навчання у виявленні аномалій є побудова поведінкових профілів IoT пристроїв. Кожен тип пристрою має характерні патерни комунікації, періодичність передачі даних, типові об'єми трафіку та набір протоколів, які він використовує. Аналіз цих характеристик дозволяє створити базову модель нормальної поведінки, відхилення від якої може свідчити про потенційні проблеми або атаки. Методи машинного навчання без учителя, такі як автоенкодери та одноклассові класифікатори, добре підходять для моделювання нормальної поведінки та виявлення аномалій [2].

Особливу увагу слід приділити попередній обробці даних та вибору релевантних ознак для аналізу. Мережевий трафік IoT пристроїв може бути представлений у вигляді багатовимірних часових рядів, що включають різноманітні метрики: від базових характеристик пакетів до складних статистичних показників та похідних ознак. На рисунку 3 показано процес формування та відбору ознак для аналізу IoT трафіку, включаючи етапи агрегації, нормалізації та зниження розмірності даних.

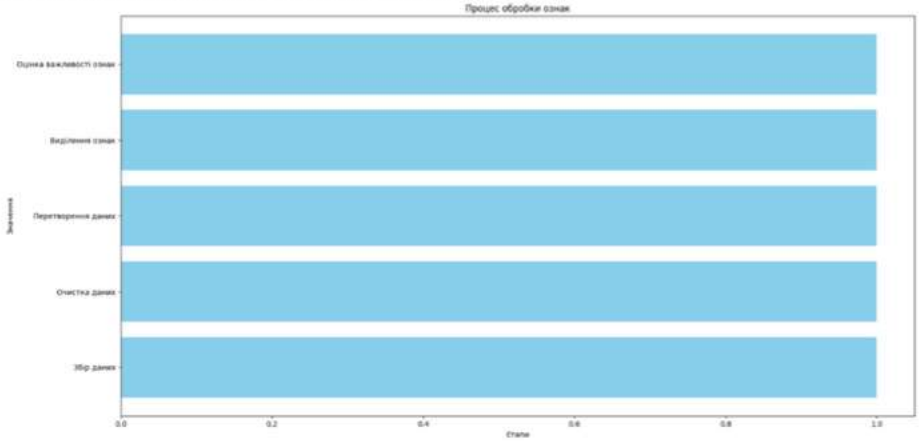

Рисунок 3 - Процес обробки ознак

Важливим аспектом є також врахування контексту та сезонності у поведінці IoT пристроїв. Багато пристроїв мають чітко виражені добові або тижневі патерни активності, і відхилення від цих патернів може бути індикатором аномалії. Методи машинного навчання повинні враховувати ці часові залежності та адаптувати свої моделі відповідно до змін у нормальній поведінці пристроїв.

Окремої уваги заслуговує проблема масштабованості систем виявлення аномалій в IoT мережах. З ростом кількості пристроїв та обсягів даних зростають вимоги до обчислювальної ефективності алгоритмів та їх здатності працювати в режимі реального часу. Тут перспективним є використання розподілених систем обробки даних та методів інкрементального навчання, які дозволяють оновлювати моделі без повного перенавчання на всьому наборі даних.

Практичне впровадження систем виявлення аномалій на основі машинного навчання в IoT мережах вимагає також вирішення ряду технічних та організаційних питань. Зокрема, важливим є забезпечення надійного збору даних з усіх пристроїв, організація безпечного зберігання та обробки цих даних, а також інтеграція з існуючими системами моніторингу та управління безпекою.

Ефективність методів машинного навчання у виявленні аномалій значною мірою залежить від якості навчальних даних та правильного налаштування параметрів алгоритмів. Важливо регулярно оновлювати моделі, враховуючи нові типи загроз та зміни у поведінці пристроїв. Крім того, система повинна мати механізми валідації виявлених аномалій та мінімізації кількості помилкових спрацьовувань.

У контексті IoT безпеки особливо важливим є швидке реагування на виявлені

аномалії. Системи на основі машинного навчання повинні не тільки виявляти підозрілу активність, але й надавати детальну інформацію про характер аномалії та можливі причини її виникнення. Це дозволяє фахівцям з безпеки швидко оцінити ситуацію та вжити необхідних заходів [3].

Перспективним напрямком розвитку є інтеграція методів машинного навчання з іншими технологіями безпеки, такими як блокчейн та федеративне навчання. Це дозволить підвищити надійність систем виявлення аномалій та забезпечити їх ефективну роботу в розподілених IoT середовищах.

Висновок. Підводячи підсумки дослідження застосування машинного навчання для виявлення аномалій в мережах Інтернету речей, можна констатувати, що дана технологія демонструє значний потенціал у забезпеченні безпеки та стабільності IoT інфраструктури. Методи кластеризації, зокрема K-Means та DBSCAN, показали високу ефективність у автоматичному виявленні підозрілої активності та відхилень від нормальної поведінки пристроїв. Особливо важливим аспектом є здатність цих алгоритмів адаптуватися до різних типів даних та масштабуватися відповідно до зростання IoT мереж.

Використання поведінкових профілів та контекстного аналізу разом з методами машинного навчання дозволяє створювати комплексні системи моніторингу, які можуть ефективно працювати в гетерогенному середовищі IoT. Важливим фактором успіху є правильна попередня обробка даних, включаючи агрегацію, нормалізацію та відбір релевантних ознак, що забезпечує точність та надійність виявлення аномалій. Інтеграція з існуючими системами безпеки та можливість роботи в режимі реального часу роблять такі рішення практично застосовними в реальних умовах.

Перспективи розвитку даного напрямку пов'язані з подальшим удосконаленням алгоритмів машинного навчання, впровадженням нових методів аналізу даних та інтеграцією з передовими технологіями, такими як блокчейн та федеративне навчання. Особливу увагу слід приділити розробці методів, які дозволять ще більше підвищити точність виявлення аномалій при одночасному зниженні кількості помилкових спрацьовувань. Крім того, важливим напрямком є розробка стандартів та методологій для оцінки ефективності таких систем та їх сертифікації для використання в критичних інфраструктурах.

Перелік використаних джерел.

1. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey [Електронний ресурс] // IEEE Transactions on Knowledge and Data Engineering. - 2009. - Т. 21, № 3. - С. 509-525. - Режим доступу до ресурсу: https://doi.org/10.1109/TKDE.2008.119.

2. Ahmed M., Mahmood A. N., Hu J. A Survey on Anomaly Detection in IoT [Електронний ресурс] / M. Ahmed, A. N. Mahmood, J. Hu // Journal of Network and Computer Applications. - 2016. - Т. 83. - С. 1-21. - Режим доступу до ресурсу: https://doi.org/10.1016/j.jnca.2017.01.003.

3. Kumar A., Singh S. Anomaly Detection in IoT Using K-Means Clustering [Електронний ресурс] / A. Kumar, S. Singh // International Journal of Computer Applications. - 2019. - Т. 182, № 12. - С. 1-5. - Режим доступу до ресурсу: https://doi.org/10.5120/ijca2019918665.