ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«**Методи виявлення аномалій у трафіку IoT-мереж із використанням машинного навчання**»

на здобуття освітнього ступеня магістр
за спеціальності 126 Інформаційні системи та технології
(код,найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Владислав Білоус
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав:
здобувач вищої освіти
група ІСДМ-61 Владислав Білоус
(ім'я, ПРІЗВИЩЕ)

Керівник Ірина Срібна
д.т.н.
доцент *(ім'я, ПРІЗВИЩЕ)*

Рецензент: _____
(ім'я, ПРІЗВИЩЕ)

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти магістр
Спеціальність 126 Інформаційні системи та технології
Освітньо-професійна прорама Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ
_____ Каміла СТОРЧАК

“____” _____2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Білоусу Владиславу Вячеславовичу
(прізвище, ім’я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Методи виявлення аномалій у трафіку IoT-мереж із використанням машинного навчання
керівник кваліфікаційної роботи: Ірина СРІБНА д.т. наук, доцент
(ім’я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від “___ ” жовтня 2025 р. № ____

2. Строк подання кваліфікаційної роботи «26» грудня 2025 р.

3. Вихідні дані кваліфікаційної роботи:
1. Наукові публікації та оглядові матеріали з виявлення аномалій у мережевому трафіку IoT та суміжних задач мережевої безпеки.
2. Матеріали з аналізу IoT-трафіку за метаданими в умовах шифрування прикладного рівня.
3. Технічна документація та специфікації протоколів і стеків IoT (MQTT, CoAP, AMQP) та підходи до сегментації й захисту IoT-інфраструктури.
4. Експериментальні дані корпоративного IoT-сегмента: мережеві потоки та події прикладного рівня
5. Програмні засоби та документація інструментів машинного навчання і MLOps для побудови, оцінювання та супроводу моделей детекції

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно

розробити):

1. Аналіз сучасних методів виявлення аномалій у трафіку IoT-мереж, визначення їх переваг і недоліків.
2. Формування класифікації типових атак і вразливостей, характерних для IoT-інфраструктур.
3. Обґрунтування вибору алгоритмів машинного навчання для аналізу та класифікації аномальної поведінки мережевого трафіку.

5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «30» жовтня 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз науково-технічної літератури та предметної області (IoT-мережі, аномалії, ML)	30.10 – 06.11.25	
2.	Аналіз IoT-інфраструктур, типових аномалій та атак та джерел мережевих даних	07.11 – 12.11.25	
3.	Формування вимог до методики детекції та критеріїв оцінювання ефективності	13.11 – 15.11.25	
4.	Проєктування конвеєра даних, простору ознак та архітектури рішення	16.11 – 25.11.25	
5.	Розробка прототипу: підготовка даних, побудова ознак, навчання та налаштування моделей	26.11 – 08.12.25	
6.	Експериментальна перевірка (підбір порогів, аналіз результатів, тестування)	09.12 – 13.12.25	
7.	Оформлення пояснювальної записки, висновків	14.12 – 18.12.25	
8.	Підготовка презентації та матеріалів до захисту	19.12 – 23.12.25	

Здобувач вищої освіти _____ Владислав БІЛОУС
(підпис) *(ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Ірина СРІБНА
(підпис) *(ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступня магістр: 91 стор., 24 рис., 2 табл., 30 джерел.

Мета роботи: підвищити ефективність виявлення нетипової поведінки в IoT-сегменті корпоративної мережі шляхом застосування методів машинного навчання та інформативних ознак, сформованих на основі мережевих метаданих трафіку.

Об'єкт дослідження: процеси аналізу та моніторингу мережевого трафіку IoT-інфраструктури.

Предмет дослідження: методи виявлення аномалій у трафіку IoT-мереж на основі статистичних, часових і протокольних характеристик та алгоритмів машинного навчання.

Короткий зміст роботи: роботу присвячено розробленню та апробації підходу до виявлення аномалій у трафіку IoT-мереж у корпоративному середовищі. Актуальність теми зумовлена зростанням кількості IoT-пристроїв, гетерогенністю протоколів і режимів роботи, а також ризиками інцидентів безпеки й деградації сервісів. Враховано обмежені ресурси вузлів і поширене шифрування прикладного рівня, тому запропоноване рішення спирається на аналіз метаданих трафіку без глибокої інспекції вмісту пакетів. Розроблено конвеєр підготовки даних (очищення, узгодження, синхронізація часових міток, віконування, нормалізація, псевдонімізація ідентифікаторів) і сформовано матрицю ознак. Для умов дефіциту розмітки обґрунтовано та реалізовано безнаглядний підхід із розрахунком скору аномальності й порогуванням, а також післяобробку для агрегування спрацювань у інциденти з метою зменшення хибних тривог. Показано, що поєднання інженерії ознак на метаданих із керованим контуром експлуатації (MLOps: моніторинг якості даних і дрейфу, відтворюваність експериментів, контроль оновлень) підвищує практичну придатність рішення для інтеграції в системи мережевого моніторингу та SOC/SIEM-процеси.

Ключові слова: ІОТ, МЕРЕЖЕВИЙ ТРАФІК, АНОМАЛІЇ, ВИЯВЛЕННЯ АНОМАЛІЙ, МАШИННЕ НАВЧАННЯ, МЕТАДАНІ ТРАФІКУ, БЕЗНАГЛЯДНІ МЕТОДИ, ISOLATION FOREST, MLOPS, ДРЕЙФ ДАНИХ.

ABSTRACT

The text part of the qualifying work for obtaining a bachelor's degree: 91 pp., 24 fig., 2 tables, 30 sources.

The purpose of the work is to improve the effectiveness of abnormal behavior detection in a corporate IoT network segment by applying machine learning methods and informative features derived from network traffic metadata.

Object of research: the process of analyzing and monitoring IoT infrastructure network traffic.

Subject of research: anomaly detection methods for IoT network traffic based on statistical, temporal, and protocol-level characteristics combined with machine learning algorithms.

Summary of the work: the thesis develops and validates an anomaly detection approach for IoT traffic in a corporate environment. Given device heterogeneity, limited computational resources, and widespread application-layer encryption, the solution relies on traffic metadata rather than payload inspection. A data preparation pipeline is designed (cleaning, alignment of timestamps, window-based aggregation, normalization, and identifier pseudonymization) and a feature matrix is constructed from metadata. Under limited labeled incidents, an unsupervised strategy with anomaly scoring and thresholding is implemented and complemented with post-processing to aggregate low-level alerts into incidents and reduce false positives. An operational MLOps perspective is included, covering data quality and drift monitoring, controlled model updates, and reproducible experimentation, enabling integration into network monitoring and SOC/SIEM workflows.

Keywords: IOT, NETWORK TRAFFIC, ANOMALIES, ANOMALY DETECTION, MACHINE LEARNING, TRAFFIC METADATA, UNSUPERVISED LEARNING, ISOLATION FOREST, MLOPS, DATA DRIFT.

ЗМІСТ

ВСТУП

Актуальність теми. У наш час цифрові світові технології Інтернету речей (Internet of Things, IoT) дуже швидко вливаються в усі сфери діяльності людини , від промислової індустрії, транспорту, медицини та інших до розумних (інтелектуальних) міст та побутових приладів. Велика кількість взаємозв'язаних сенсорів та контролерів згенеровує потоки вихідних даних, які забезпечують автоматизацію різних процесів, їх аналітику в реальному часі та оптимізацію витрат. Одночасно з цим зростання масштабів IoT-середовища створює нові виклики, що пов'язані з питаннями безпеки в інформаційному просторі та конфіденційності даних.

Більша частина пристроїв IoT має обмеження обчислювальних ресурсів, що не дає можливості впроваджувати складні алгоритми шифрування або автентифікації. Окрім цього, відсутність єдиних стандартів безпеки та різноманіття протоколів призводять до ускладнення виявлення підозрілої активності в мережі. Зловмисники активно користуються цим, організовуючи атаки DDoS, підміну даних, викрадання інформації та несанкціоноване керування пристроями. За даними Європейського агентства з кібербезпеки (ENISA) у 2023 році більш ніж 40% атак у сфері IoT були зв'язані з несанкціонованим доступом до мережевого трафіку та компрометацією кінцевих пристроїв [1].

Сучасні традиційні підходи до моніторингу трафіку, що засновані на статичних правилах чи сигнатурних методах, не здатні ефективно виявляти нові типи загроз, які швидко змінюються. За цих обставин методи машинного навчання дедалі стають перспективним напрямом, що дозволяє автоматично виявляти аномалії на підставі аналізу поведінкових характеристик трафіку. Науковці підкреслюють, що поєднання алгоритмів класифікації, кластеризації та глибинного навчання забезпечує підвищення точності детекції атак у порівнянні з класичними методами [2].

Ця проблема є особливо актуальною для України в умовах цифрової трансформації економіки та активного запровадження технологій “розумного міста”, промислових IoT-систем та критичної інфраструктури. Створення національних підходів до виявлення аномалій у трафіку цих систем допоможе підвищенню кіберстійкості державних і приватних об'єктів, а також забезпеченню інформаційної безпеки суспільства.

Метою магістерської роботи є розроблення та дослідження методів виявлення аномалій у трафіку IoT-мереж із використанням алгоритмів машинного навчання, які забезпечують підвищення точності та швидкості дії процесів ідентифікації загроз.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Провести аналіз сучасних методів виявлення аномалій у трафіку IoT-мереж , виявити їх недоліки та переваги.
2. Сформувати класифікацію типових атак та вразливостей, які характерні для IoT-інфраструктур.
3. Обґрунтування вибору алгоритмів машинного навчання для аналізу та класифікації аномальної поведінки мережевого трафіку.
4. Розробка моделі виявлення аномалій відповідно до відібраних методів машинного навчання.
5. Провести експериментальну перевірку ефективності запропонованої моделі та провести оцінку її точності та продуктивності.

Об'єктом дослідження є процес аналізу та моніторингу мережевого трафіку IoT-систем із метою виявлення аномальних відхилень, що можуть свідчити про загрози безпеці або порушення нормального функціонування пристроїв.

Предмет дослідження - методи виявлення аномалій у трафіку IoT-мереж з використанням алгоритмів машинного навчання, які спрямовані на підвищення ефективності систем кіберзахисту.

Методи дослідження

У роботі використано комплекс теоретичних та практичних методів. Теоретичні дослідження - аналіз наукових джерел, систематизація існуючих підходів, порівняльний аналіз алгоритмів машинного навчання для задач класифікації та кластеризації.

Практичні методи - це моделювання та експериментальні дослідження. Для побудови та тестування моделі виявлення аномалій використовуються інструменти програмного середовища Python з використанням бібліотек Scikit-learn, TensorFlow та PyTorch. В процесі дослідження також використовуються методи статичного аналізу, нормалізації даних та побудови метрик точності та візуалізації результатів.

Наукова новизна роботи полягає вдосконаленні підходів до виявлення аномалій в трафіку IoT-мереж шляхом поєднання класичних методів аналізу поведінки пристроїв із сучасними алгоритмами машинного навчання.

Запропонована модель враховує обмежені ресурси IoT-пристроїв, нестабільність мережевих каналів та неоднорідність форматів даних, що дає змогу підвищити точність детекції аномалій без суттєвого збільшення обчислювального навантаження [2].

Практична значимість дослідження полягає у можливості використання отриманих результатів для підвищення рівня кіберзахисту державних та корпоративних мереж, що використовують IoT-інфоструктури. Запропонована модель може бути інтегрована у системи моніторингу мережевого трафіку та системи виявлення вторгнень (IDS/IPS) або рішення для управління безпекою промислових систем (ISD/SCADA). Це в подальшому відкриває перспективи створення інтелектуальних платформ кіберзахисту для критично важливих об'єктів в Україні, зокрема в транспортній та енергетичній галузях , сфері розумних міст та інш.[3].

Таким чином результати магістерської роботи мають як теоретичну, так і практичну (прикладну) цінність - вони забезпечують в подальшому розвиток технологій машинного навчання для забезпечення (Захисту) безпеки IoT-мереж, підвищуючи рівень кіберстійкості сучасних інформаційних систем.

1 ТЕОРЕТИКО-МЕТОДИЧНІ ОСНОВИ ВИЯВЛЕННЯ АНОМАЛІЙ У ТРАФІКУ IoT-МЕРЕЖ

1.1 Розвиток, архітектура та класифікація систем Інтернету речей

Концепція взаємодії фізичних об’єктів із цифровими системами виникла ще на початку другої половини XX століття. Якраз в цей час відбувся стрімкий розвиток обчислювальних та телекомунікаційних технологій. В 60-ті роки XX століття почали активно вивчати концепції кіберфізичних систем, але в той час можливості апаратного забезпечення були дуже обмежені, що і стало причиною затримки практичної реалізації цих ідей. І лише в 1980-х роках завдяки розробці нових мікроконтролерів та сенсорних технологій виникла можливість фізичного підключення пристроїв до мережі для збору даних та подальшої передачі їх у віддалені обчислювальні системи.

Одним з перших експерементів, що продемонстрував потенціал мережевої взаємодії пристроїв, стала автоматизована система моніторингу з запасів напоїв у торгових автоматах Coca-Cola в Університеті Карнегі-Меллона (США). Система працювала через мережу ARPANET і надавала інформацію про кількість товару та температуру напоїв. Цей приклад вперше продемонстрував , що можливо передавати данні від фізичного пристрою через мережу без залучення участі людини[4].

Поступовий розвиток мережевих технологій, у тому числі поява протоколу TCP/IP та стантартизація обміну даними між пристроями, зумовили появу концепції повсюдного підключення пристроїв до мережі. У 1990-х роках з розвитком технології RFID і зниженням вартості мікропроцесорних систем виникли сприятливі умови для впровадження ідентифікації об'єктів у логістичних ланцюгах, роздрібній торгівлі та на виробництві [5].

Саме на даному етапі з'явився термін “Internet of Things”, йог у 1999 році запропонував Кевін Аштон . Він був дослідником у сфері автоматизації і працював над технологіями відстежування за товарами Procter &

Gamble.Поняття цього терміну відображало ідею про те, що фізичні речі будуть учасниками інформаційного обміну так, як і комп'ютера, автоматично передаючи данні без залучання людини.

На початку 2000-х років розвиток IoT відбувався на фоні поширення бездротових мереж, таких як Bluetooth та Wi-Fi, що дало змогу підключати побутові пристрої до мережі Інтернету. Того ж десятиріччя з'являються перші комерційні прототипи розумних пристроїв, таких як системи ”розумний дім»”.

У XXI сторіччі розвиток хмарних технологій, зростання пропускної здатності мереж та здешевшення вартості модулів для обробки даних призвели до експоненційного зростання IoT-пристроїв, які на сьогоднішній час обчислюються мільярдами - від сенсорів у мережах промисловості до персональних фітнес-браслетів.Значно розширилися також і архітектурні підходи: замість централізованої обробки даних впроваджуються концепції edge-computing та fog-computing, що передбачають обробку інформації на рівні пристроїв або локальних вузлів.

Таким чином, еволюція IoT- це результат багаторічного поєднання технологій зв'язку, обчислень, програмного забезпечення та сенсорики. На даний момент вона трансформує як промисловість так і побут та формує глобальну екосистему взаємодії фізичного та цифрового світів.

Розвиток технології Інтернету речей у XXI сторіччі визначається стрімким зростанням масштабів застосування та різноманітністю сфер використання. На сьогоднішній час IoT охоплює як промислові системи (IIoT) так і транспортні мережі, аграрні системи моніторингу, медичні діагностичні комплекси, розумні оселі, інфраструктури безпеки міста та інше.

За даними аналітичного звіту компанії N-iX, у 2023 році кількість IoT пристроїв у світі становила приблизно 16 мільярдів, а до 2030 року очікується перевищення позначки у 29 мільярдів підключень [6]. Така динаміка стає можлива завдяки здешевшенню сенсорів, розвитку комунікаційних технологій 5G та зростанню вимог до автоматизації бізнес-процесів.

Одна з провідних тенденцій – це інтеграція IoT із системами штучного інтелекту. Взаємодія цих технологій забезпечує побудову автономних систем управління, предиктивної діагностики, адаптивних бездротових мереж і сервісів, які діють в реальному часі. Машинне навчання та методи обробки великої кількості даних (Big Data) дедалі все частіше застосовуються для аналізу телеметрії з пристроїв, виявлення аномалій, прогнозування поломок та для оптимізації процесів взаємодії.

Ще однією ключовою тенденцією є зрушення від централізованої хмарної обробки до edge-computing - архітектури, в якій обробка даних відбувається на рівні пристроїв або шлюзів мережі, що дозволяє зменшити затримки, знизити навантаження на мережу та забезпечити більшу надійність роботи у реальному часі [7].

Завдяки розвитку “розумних міст” (smart cities) IoT інтегрується у транспортні системи, енергомережі, адміністративні сервіси та громадську безпеку. В цих випадках IoT виконують критичні функції моніторингу, звітності та реагування, що підвищує ефективність управління міськими процесами та сприяє підвищенню рівня життя громадян.

Попри широкі можливості, розвиток IoT також супроводжується серйозними викликами, зокрема - це такі питання як кібербезпека, конфіденційність даних, сумісність пристроїв різних виробників і нестача стандартів на рівні протоколів. Особлива увага має приділятися виявленню аномалій у мережевому трафіку IoT, оскільки такі системи стають об'єктами атак, що може призвести до збоїв у роботі критичної інфроструктури.

Завантаженість мереж, велика кількість взаємодіючих вузлів і залежність від бездротового середовища створюють ризики для конфіденційності та доступності і цілісності даних. Сьогодні зростає і потреба в автоматизованих методах аналізу трафіку - зокрема, це методи машинного навчання, що дозволяють оперативно виявляти викривлення та вторгнення до мережевої інфраструктури.

Архітектура систем Інтернету речей (IoT) виступає фундаментальною базою функціювання “розумних” екосистем, де фізичні пристрої, мережі передачі даних та хмарні сервіси взаємодіють між собою для забезпечення збору, передачі, обробки та використання інформації. Візуалізація та стандартизація архітектурних шарів дозволяють науково обґрунтовувати методи моніторингу, аналізу та виявлення аномалій у такій мережі.

Одним з типових підходів до малювання архітектури IoT є багаторівнева модель, що складається з трьох або п'яти шарів. У трирівненому варіанті виділяють : рівень пристроїв (Sensors/Things), рівень шлюзу чи проміжної обробки (Gateway/Edge) і рівень хмарної платформи (Cloud). Водночас п'ятишарова модель розширює цей опис, включаючи наступні компаненти : рівень сприйняття (Perception), рівень передачі (Network), рівень обробки даних (Processing/Transport), рівень сервісів та платформи (Service/Cloud) і рівень застосунків (Application) [8].

Рівень сприйняття включає фізичні сенсори та виконавчі механізми, що взаємодіють із реальним світом. Дані, що генеруються цими пристроями , передаються на рівень мережі, де здійснюється їх агрегація, передача через протоколи зв'язку (наприклад, Wi-Fi, ZigBee, LoRaWAN) і забезпечується зв'язок з шлюзами. На рівні обробки/транспорту можливо використовувати проміжні вузли (Edge/Fog) для попереднього навантаження на центральну систему.Рівень сервісів/платформи включає хмарну інфроструктуру для зберегання, аналізу та майнінгу великих масивів даних, а також надання API та аналітичних сервісів. Нарешті, рівень додатків охоплює сценарії користувачів – “розумний дім”, промисловий моніторинг, транспортні системи та інше.

Найбільш сучасні архітектури активно використовують підхід edge-fog – обчислень, що дозволяє виконувати аналіз безпосередньо на пристроях чи шлюзах перед їх передачею у хмару. Такий підхід значно зменшує ризики, підвищує масштабованість та зменшує ризики передачі чутливих даних централізованим центрам. У той же час він ставить нові виклики : необхідність

управління гетерогенними пристроями, забезпечення безпеки даних на проміжних вузлах, управління оновленнями та протоколами.

Оскільки архітектура IoT є фундаментом для побудови систем виявлення аномалій, вона має важливе значення для подальших досліджень. Знання різних рівнів, точок обробки та каналів зв'язку дає можливість вимикати аномалії - чи то на рівні сенсора, чи на шлюзі, чи в хмарі. В майбутньому це дозволить створювати адаптивні моделі моніторингу, які зорієнтовані на специфіку кожного рівня.

Існує широкий спектр систем Інтернету речей (Internet of Things ,IoT) , які відрізняються за сферою використання , масштабом, технічними характеристиками , енергоспоживанням, типами переданих даних та рівнем інтеграції з іншими системами. Така різноманітність спричиняє потребу в класифікації IoT-систем для правильного їх аналізу, проєктування та забезпечення кібербезпеки.

Одним з базових підходів до класифікації IoT-систем є поділ їх за сферою використання. Виділяють основні категорії - це побутові (Consumer IoT), промислові (Industrial IoT), медичні (Healthcare IoT), транспортні (Transportation IoT) та міські (Smart City IoT).

Побутові (Consumer IoT) це - пристрої “розумного дому” (розумні лампи, охоронні камери, фітнес -трекери, термостати та інше), а також персональні гаджети, що забезпечують комфорт і безпеку користувачів у повсякденному житті.

У виробництві, енергетиці, логістиці, сільському господарстві використовуються промислові (Industrial IoT, IIoT) системи. Такі системи забезпечують моніторинг процесів, автоматизацію операцій та контроль стану обладнання, прогнозування можливих аварій.

Медичні (Healthcare IoT системи забезпечують збирання даних із біосенсорів, відстеження стану здоров'я пацієнтів, телемедицину та дфагностику стану в реальному часі.

В автоматизованих системах управління рухом, на розумних парковках, безпілотному транспорті та логістичних мережах застосовують транспортні (Transportation IoT).

Міські (Smart Cits IoT) охоплюють освітлення , відеоспостереження, “розумні” лічильники ресурсів, системи безпеки, моніторинг якості повітря та управління інфраструктурою міста.

Інший підхід передбачає класифікацію за масштабом розгортання - це локальні IoT-системи, регіональні IoT-мережі, глобальні IoT-платформи.

Локальні IoT-системи - це системи , що функціанують у межах окремої будівлі або підприємства.

Мережі, що об'єднують інфраструктуру в межах міста або галузі - це регіональні IoT-мережі.

Мережі , які охоплюють міжнародні об'єкти, розподілені мережі та хмарні сервіси - це глобальні IoT-платформи.

Також системи IoT можливо класифікувати за режимом обміну даними - періодичний, безперервний, реактивний.

Системи IoT також класифікують за типом переданих сигналів - цифрові або аналогові; за джерелами живлення - автономні, з дротовим живленням, енергоефективні; топологіями мережі - зірка, сітка, каскадна структура.

Окремо можливо класифікувати IoT-системи з розподілом за критеріями безпеки та захищенності на системи із вбудованими механізмами (шифрування, автентифікація, виявлення вторгнень) та системи з низьким рівнем захисту, що працюють у відкритих або неконтрольованих середовищах.

Інформація про класифікацію IoT-систем має практичну цінність при аналізі специфіки мережевого трафіку, побудові систем моніторингу та створенні алгоритмів виявлення аномалій. Розуміння типу, масштабу і призначення IoT-системи дає змогу коректно визначити вимоги до обробки даних, обирати відповідні протоколи та забезпечувати належний рівень кібербезпеки.

Мережевий трафік систем Інтернету речей (IoT) суттєво відрізняється від трафіку в традиційних IT-мережах і це пов'язано з гетерогенністю пристроїв, енергетичними обмеженнями, різноманіттям протоколів і режимів роботи, що на пряму має вплив на моделі моніторингу та виявлення аномалій [9].

Для більшості вузлів характерні три базові патерни (так звані типи шаблонів трафіку) - це періодична телеметрія з малими пакетами; подієвий (event-driven)трафік з сплесками передавання при настанні події та керувальні команди/конфігурації з жорсткими вимогами до затримки та цілісності. Комбінація цих режимів формує нерівномірне ”вибухове” навантаження (bursty), що призводить до ускладнень в класичному трафік-інженірингу.

У багатьох IoT-мережах канали мають низький бітрейт, тобто низьку пропускну здатність(LPWAN , ZigBee , BLE), а корисний payload (малі корисні навантаження) обмежуються одиницями чи десятками байтів, що робить накладні службові заголовки відносно значними та стимулює до використання компактних форматів кодування.

Передавання трафіку є часто асиметричним (uplink / downlink), значна кількість пристроїв працює з глибоким сном і короткими вікнами активності; це призводить до кластеризації пакетів у часі та довгих безтрафікових пауз, що важливо враховувати у часових ознаках (режим “сну”).

На різних рівнях застосовують легковажні протоколи (MQTT, CoAP, AMQP, UDP), канальні технології (ZigBee, LoRaWAN, BLE), а також специфічні механізми підтверджень на прикладному рівні(різновидність протоколів і стеків), унаслідок чого розподіли портів/прапорців і типові сеансові патерни відрізняються від корпоративних мереж.

Орієнтація на IPv6-стек (6LoWPAN) у поєднанні з малими MTU та ненадійними каналами збільшує частку фрагментації, повторних передавань і варіативність затримок/втрат, що впливає на спостережувані статистики.

Критичні сценарії (аварійні сповіщення, промислове керування) вимагають низької затримки й гарантованої доставки, тоді як побутова телеметрія є більш толерантною до затримок і втрат; отже, “аномальність” має

контекстну природу, зумовлену класом застосунку (чутливістю до затримки та надійності).

Захисні механізми (TLS/DTLS, OSCORE) збільшують накладні витрати(профілі енергоспоживання та безпеки), тому шифрування інколи термінується на шлюзах; у таких випадках аналіз переносять на edge-рівень або оперують метаданими замість payload, що впливає на доступні ознаки для детекції.

Події в IoT часто корельовані у просторі/часі (кластерні сплески повідомлень), розподіли міжпакетних інтервалів мають «довгі хвости», спостерігаються сезонність і дискретність; ці властивості визначають вимоги до інженерії ознак (вікна, лог-масштабування, ентропійні метрики).

Зазначені особливості визначають вибір моделей для виявлення аномалій: методи, стійкі до розріджених нерівномірних часових рядів (автоенкодери реконструкції, LSTM/TCN, Isolation Forest, DTW-подібні метрики), а також підходи, придатні для розміщення на edge-вузлах у режимі обмежених ресурсів.

1.2 Кіберзагрози й аномалії трафіку в IoT-мережах

Ефективність і надійність систем Інтернету рече рівнях й значною мірою визначається правильною добіркою протоколів на всіх рівнях мережевого стеку -від фізичного з'єднання до прикладного інтерфейсу. Специфіка IoT (обмежені енергоресурси, малі playoad, інколи нестабільний бездротовий канал, вимоги до масштабування) призвели до появи окремого класу “легковажних” протоколів та адаптацій класичних мережевих рішень.

На найнижчому рівні використовують технології з різними компромісами між дальністю, швидкістю та енергоспоживанням. Для персональних та побутових сценаріїв поширені DLE та класичні PAN-технології на підставі 802.15.4 (на приклад ZigBee), які забезпечують малу потужність і середню дальність. У будівельній автоматизації застосовуються сітчасті мережі (mesh) на базі ZigBee/Z-Wave, де вузли ретранслюють пакети, підвищуючи надійність

покриття. Для міських та промислових розгортань, де важлива велика дальність та автономна робота на батареї роками, застосовують LPWAN-рішення (приклад – LoRaWAN), які забезпечують низький бітрейт, але кілометрові дистанції. У сценаріях, які потребують стільникового покриття та керованого QoS, користуються вузькосмуговими стільниковими технологіями для IoT (як приклад- NB-IoT, LTE-M), а також енергоефективні варіанти WI-FI для пристроїв з періодичною передачею даних. Вибір технології визначається профілем трафіку (періодичний або подієвий), мобільністю вузлів, вимогами до затримки та доступним енергобюджетом.

Щоб уніфікувати взаємодію різнорідних радіомереж із IP-світами, широко застосовують IPv6 з механізмами стиснення заголовків та фрагментації для малих MTU. У багатоступеневих бездротових топологіях використовують протоколи маршрутизації, що враховують енерговитрати, якість каналу та стабільність посилань у сітчастих мережах. У практиці IoT часто присутні шлюзи, які виконують роль прикордонних вузлів. Вони транслюють кадри з канального рівня в IP-пакети, з'єднують (агрегують) трафік, термінують захист і виконують попередню аналітику (edge).

На транспорті домінує UDP, як безз'єднаний протокол з мінімальними накладними витратами та кращею поведінкою в мережах із втратами. TCP застосовують там, де потрібні надійна доставка та контроль потоку, проте його механізми можуть бути надмірними для батарейних вузлів і каналів з частими перешкодами. Для сценаріїв з потребою в квазіреальному часі та контролі чергування іноді використовують протоколи з налаштованими таймерами повторної передачі і додатковими механізмами підтверджень на прикладному рівні. Важливою практикою є перенесення частини вимог на рівень застосунку, саме там задаються політики підтвердження доставки, визначаються пріоритетність та повтори.

На прикладному рівні домінують два стилі взаємодії . Перше – це Rublish/Subskcrsbe (публікація/підписка)-типовий для телеметрії й подієвого трафіку. Брокер отримує повідомлення від пристроїв-видавців і розповсюджує їх

підписникам за тематичними «топіками». Цей підхід добре маштабується, ізолює пристрої від прямого з'єднання з усіма споживачами даних, знижує вимоги до адресації та спрощує контроль доступу. У мережах моделі pub/sub застосовують легковажні протоколи з різними рівнями гарантій доставки (QoS), підтримкою офлайн-буферизації та “утриманих” повідомлень (retained). Друге- це Reguest/Response (запит/відповідь)- природний для керування , конфігурацій та інтеграції з REST-сервісами. Клієнт формує запит на ресурс (читання параметра, зміна стану), отримує відповідь і може опрацьовувати коди помилок, тайм аути й повтори.

Обидві моделі можуть співіснувати в одному рішенні: наприклад, телеметрія й події публікуються через брокер, а критичні команди та управління пристроями відбуваються через запити до конкретних вузлів чи до шлюзів.

Для публікаційно-підписної моделі використовують легкі брокерні протоколи з QoS-рівнями, фільтрами тем і механізмами збереження офлайн-повідомлень.

Для моделі запит/відповідь застосовують компактні протоколи поверх UDP із можливістю підтвердження і повторів, або ж класичні REST-підходи поверх HTTP(S) у випадках, коли пристрої є відносно “потужними” чи працюють за шлюзом. Обмін даними здійснюють у форматах JSON для читабельності, CBOR/MessagePack - коли важливі розмір і швидкість серіалізації, протокол-буфери - для чітко типізованих контрактів.

IoT-сценарії різняться вимогами: від ”краще приблизно й зараз” (некритична телеметрія) до “рівно один раз і негайно” (керування приводами). Тому прикладні протоколи пропонують кілька рівнів QoS — від “без підтвердження” до “гарантовано один раз”, із різними механізмами підтверджень і дедуплікації. Де потрібні суворі гарантії, застосовують транзакційну логіку на брокері/шлюзі, черги, повторні спроби з експоненційною затримкою та маркери ідемпотентності.

Окрім передачі даних, IoT потребує протоколів для керування життєвим циклом вузлів: первинне «заводське підключення й довіра (bootstrapping),

надання облікових даних, дистанційні оновлення прошивки (OTA), читання/зміна конфігурацій, діагностика. Всі ці функції реалізуються через спеціальні профілі поверх прикладних протоколів або через окремі сервіси керування на шлюзах/ у хмарі. Важливий принцип- відокремлення «каналу даних» і «каналу керування», що спрощує політики доступу та аудит.

Захист каналу реалізують за допомогою шифрування та взаємною автентифікацією на транспорті (наприклад, із сертифікатами або попередньо спільними ключами) чи на прикладному рівні (об'єктне шифрування повідомлень). Легковажні протоколи безпосередньо підтримують «полегшені» механізми захисту, для того щоб зменьшити криптографічні накладні витрати. У розподіленій архітектурі часто практикують завершення захисту на шлюзах із подальшим внутрішнім захищеним каналом до хмари. На шлюз також покладають контроль доступу, нормалізацію протоколів, виявлення базових аномалій і кешування для офлайн-режиму.

Через різноманіття фізичних і прикладних протоколів у реальних впровадженнях використовують шлюзи-транслятори. Вони приймають трафік від «периферії» (наприклад, з радіомодулів або польових шин), перетворюють його в уніфіковані повідомлення, додають метадані (час, геотеги, якість сигналу), застосовують локальні правила фільтрації/агрегації й лише потім пересилають у центр обробки. Це знижує навантаження на магістраль і дозволяє реалізувати політики “дані-біля-джерела”[26].

В умовах гетерогенності виробників критичними є узгоджені схеми даних, словники властивостей пристроїв та профілі сумісності. Стандартизовані моделі (телеметрія, події, команди) полегшують інтеграцію різних вертикалей, спрощують аналітику та прискорюють розробку. На практиці це означає використання однакових структур повідомлень, ідентифікацію пристроїв за єдиними правилами, нормалізацію одиниць вимірювання та часових відміток.

Під час проєктування необхідно балансувати між :

1. Енергоспоживанням (батарейні вузли проти живлення від мережі).

2. Дальністю сигналу (підвали, промислові зони, міська забудова).

3. Надійністю (аварійні події проти не критичної телеметрії).

4.Складністю стеку та вартістю (вбудоване ПЗ, ліцензії, обслуговування).

5. Вимогами безпеки (критична інфраструктура, персональні дані).

6. Потребою в інтеграції з існуючими платформами і мовами даних.

Підсумовуючи, стек протоколів для IoT - це багатоваріантний конструктор, де кожен рівень має "легкі" та "важкі"» опції. Грамотний вибір і поєднання технологій дозволяє досягати цільових показників енергоспоживання, масштабованості й безпеки, водночас створюючи сприятливі умови для подальшого аналітичного моніторингу та виявлення аномалій у трафіку[10].

У середовищі IoT аномалією вважають відхилення характеристик трафіку або поведінки вузлів від очікуваної норми, яке не пояснюється випадковими відхиленнями та потенційно впливає на надійність або безпеку. За структурою відхилення вирізняють точкові аномалії, коли одинична подія різко відрізняється від фонового процесу (наприклад, раптовий пік частоти запитів чи поява нетипового порту для конкретного сенсора), контекстуальні аномалії, що стають нетиповими лише з урахуванням умов (скажімо, збільшення трафіку вночі для лічильника, який у цей період зазвичай "мовчить"), а також колективні аномалії, коли послідовність на перший погляд нормальних подій утворює підозрілий шаблон, такі як синхронні звернення багатьох вузлів до рідкісної теми MQTT або серія пакетів у нехарактерному порядку. За часовою динамікою розрізняють імпульсні сплески, ступінчасті зміни режиму, повільні дрейфи параметрів та порушення сезонності, коли зникає або зміщується звичний добовий чи тижневий цикл. Аномалії можуть проявлятися на різних рівнях стеку: від фізичного та канального (нестандартні значення RSSI, збільшення втрат, зміна топології mesh) до мережевого і транспортного (аномальна кількість унікальних IP-адрес, хвилі SYN/RESET) та прикладного (нетипові шляхи запитів або теми публікацій, порушення схем даних, зміна частоти керувальних команд). За масштабом впливу вони бувають локальними для одного пристрою,

сегментними для групи або підмережі та глобальними, коли охоплюють значну частину інфраструктури.

Причини відхилень теж різноманітні: природні експлуатаційні фактори (збої сенсорів, деградація батареї, перешкоди радіоканалу), помилки конфігурації, зловмисні дії (сканування, C2-активність ботнетів, ін’єкція даних) або адміністративні події на зразок масових OTA-оновлень, які створюють короткочасні “штучні” аномалії. Водночас ступінь спостережності аномалій залежить від доступності даних: у разі наскрізного шифрування аналіз доводиться виконувати за метаданими й часовими ознаками на мережевих або edge-рівнях, тоді як payload-орієнтовані порушення краще виявляються на шлюзах або безпосередньо на пристрої. Від правильної ідентифікації типу аномалії залежить вибір методу детекції -порогові та ізоляційні моделі є доречними для точкових відхилень, контекстуальні та колективні зручніше виявляти за допомогою моделей часових рядів чи рекурентних мереж, а топологічні зміни - методами графового аналізу еволюції з’єднань.

Кіберризики в IoT формуються на перетині обмежених ресурсів пристроїв, різнотипність стеків протоколів, широкої площини атаки та складного ланцюга постачання, що охоплює виробництво, логістику, розгортання і супровід. На рівні пристроїв основні загрози пов’язані з використанням типової або заздалегідь відомої автентифікації, відсутністю безпечного завантаження та підписаних оновлень, зберіганням секретів у відкритому вигляді, а також можливістю фізичного доступу до пам’яті й інтерфейсів налагодження. На рівні комунікацій ризики випливають із застосування легковажних протоколів, що економлять енергію ціною слабших гарантій, з використанням безз’єднаних транспортів та обмежень пропускної здатності, які стимулюють зменшення захисних накладних витрат; у підсумку підвищується чутливість до атак на доступність (глушіння, перевантаження каналів), підміну або повторне відтворення повідомлень. Площина керування пристроями містить окремий клас ризиків: компрометація брокерів і платформ керування, клонування ідентичностей, помилки у політиках доступу, а також компрометація OTA-

каналу, що призводить до масштабованого зловживання через розповсюдження шкідливих прошивок. До цього додаються питання приватності, адже телеметрія нерідко містить непрямі персональні або комерційно чутливі дані; витоки метаданих дозволяють профілювати поведінку об’єктів або користувачів. Нарешті, системні ризики пов’язані з відсутністю повного інвентарю активів і версійного контролю прошивок, браком процедур реагування на інциденти, недосконалістю сегментації між IT і OT-середовищами та складністю оновлення обладнання, що працює роками в польових умовах.

Узгоджене управління цими ризиками передбачає інвентаризацію та класифікацію активів, моделювання загроз для типових сценаріїв, впровадження базових засобів ідентифікації і криптографічного захисту, сегментацію мереж і мінімізацію привілеїв, а також постійний моніторинг поведінки та телеметрії з механізмами виявлення аномалій у трафіку [11].

Практика функціонування великих IoT-екосистем демонструє, що найбільш руйнівними стають кампанії, які поєднують масове зламування однотипних пристроїв із подальшим використанням їх як платформи для координації розподілених атак або прихованих операцій. Хрестоматійним прикладом є ботнет Mirai, який у 2016 році автоматизував перебір типових облікових даних на інтернет-камери й домашні маршрутизатори, після чого зібрана бот-мережа застосовувалася для масштабних DDoS-ударів проти мережевих сервісів, зокрема провайдера DNS, що спричиняло каскадні збої у доступі до популярних веб-платформ; у подальші роки з’явилися десятки похідних варіантів Mirai із додатковими експлойтами для нових векторів проникнення, що підтвердило довготривалу життєздатність цього класу загроз [12].

Другий характерний сюжет — peer-to-peer ботнети родини Mozi, які уникають централізованих центрів керування завдяки розподіленій схемі на основі DHT: заражені вузли одночасно виступають ретрансляторами команд і оновлень, а первинне проникнення здійснюється через слабкі служби віддаленого доступу та відомі вразливості інтерфейсів керування мережевими

пристроями. Такий дизайн ускладнює ідентифікацію “голови” інфраструктури та подовжує життєвий цикл бот-мережі навіть за умов вибіркового блокування трафіку чи вилучення окремих вузлів.

Окремої уваги заслуговують операції, що пов’язані з компрометацією мережевої інфраструктури малого офісу/домашніх користувачів (SOHO). Кампанія VPNFilter демонструвала багатоступеневу архітектуру з модульними компонентами для перехоплення трафіку, прихованого проксування та руйнівного стирання, що безпосередньо зачіпає IoT-ландшафт, адже маршрутизатори й мережеві накопичувачі часто працюють як шлюзи та концентратори для ”розумних” пристроїв; уразливість цих вузлів означає доступ зловмисника до комунікаційних каналів, де циркулює телеметрія, команди й оновлення, а отже - можливість підміни або блокування обміну даними у промислових і побутових сценаріях.

Нарешті, небезпека хмарно-керованих камер відеоспостереження та інших керованих сервісів проявляється там, де поєднуються помилки керування доступом, повторне використання облікових записів і публічно доступні панелі адміністрування. Такі інциденти призводять до масового несанкціонованого перегляду відеопотоків, втручання в налаштування та розгортання ”тихих” проксі-ланцюгів на периферії мереж; у поєднанні з географічним розподілом пристроїв це створює зручний інструмент для обхідних каналів і підготовки до більш масштабних операцій у критичних сегментах інфраструктури.

Зазначені приклади відображають повторювані закономірності: слабкі або типові облікові дані, несвоєчасне встановлення оновлень, публічні інтерфейси керування без належної автентифікації, відсутність сегментації та моніторингу поведінки. Для цілей цієї роботи вони важливі тим, що формують еталонні профілі аномалій у трафіку - від різких приростів кількості сесій і зміни топології з’єднань до появи нетипових послідовностей прикладних запитів і корельованих сплесків телеметрії в групах пристроїв, які слугують маркерами для побудови й навчання моделей виявлення.

1.3 Аномалії в IoT-трафіку: класифікація, ознаки, приклади

У середовищі мереж Інтернету речей під аномалією розуміють спостереження або цілу сукупність спостережень, що сильно відхиляються від очікуваної моделі поведінки пристрою, мережевого сегмента чи сервісу з плином часу, навантаження чи навіть ролі вузла. Під поняттям “очікувана модель” в цьому випадку можна вважати не лише про загальне уявлення про нормальну поведінку, а еталону, який відповідає за певні встановлені норми: статистичний розподіл ознак трафіку, жорстко визначені правила протоколів, стандартні профілі конкретних пристроїв або їхніх груп. Аномальність завжди залежить від контексту того, що є нормальним для промислового датчика у денну зміну і що навпаки може бути нетиповим для цього ж датчика в ночі. Головне це підкреслити різницю між значущею аномалією та випадковими викидами. Важливо, що значущі аномалії відображають стабільний зсув поведінки, тоді як другі пов’язані з шумом вимірювання або ж з відмовами каналу та не мають чіткої структури. Це призводить до порушення семантики протоколу, що буде впливати на надійність системи та її безпеку.

У контексті мереж Інтернету речей аномалією вважають спостереження або сукупність спостережень, що суттєво відхиляються від очікуваної моделі поведінки пристрою, сервісу чи мережевого сегмента за певного контексту часу, навантаження та ролі вузла. Під ”очікуваною моделлю” розуміють не лише інтуїтивне уявлення про “норму”, а формалізований еталон: статистичний розподіл ознак трафіку, детерміновані правила протоколів, поведіневі профілі конкретних пристроїв або їхніх груп. Аномальність при цьому має контекстну природу: те, що є нормальним для промислового датчика у денну зміну, може бути нетиповим для тієї ж лінії вночі; так само припустимий обсяг телеметрії для мультисенсорного шлюзу не є прийнятним для батарейного лічильника з жорстким duty-cycle. Важливо підкреслити різницю між випадковими викидами та істотною аномалією: перші пов’язані з шумом вимірювання або відмовами каналу й не мають стійкої структури, тоді як другі відображають стабільний зсув

поведінки, порушення семантики протоколу або координацію подій між вузлами, що потенційно впливає на надійність і безпеку.

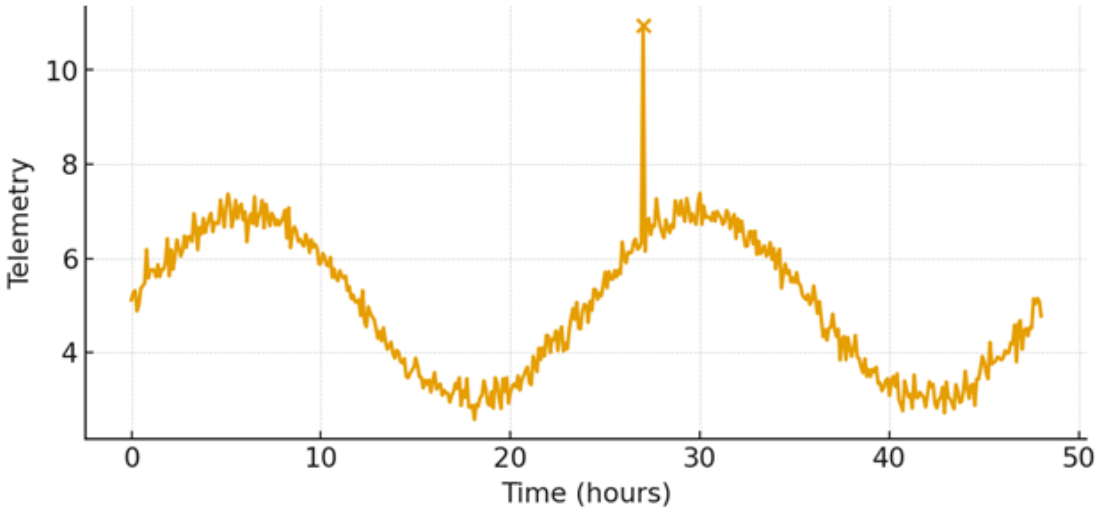


Рис. 1.1 Контекстуальна аномалія в телеметрії IoT

Формально аномалію зручно трактувати як спостереження з малою правдоподібністю відносно базової моделі, тобто таке, що має низьку ймовірність під заданим розподілом “нормальної” поведінки або розташоване далеко від навченої багатовимірної структури ознак. На практиці це реалізують через функцію оцінювання аномальності, яка відображає кожне спостереження у числовий показник і дозволяє порівнювати його з порогом; поріг може бути фіксованим або адаптивним, із урахуванням сезонності, зміни режимів та повільного дрейфу параметрів. У мережах IoT така оцінка рідко спирається на один сигнал: корисним виявляється поєднання часових характеристик (міжпакетні інтервали, стабільність періодики), транспортно-сеансових показників (повторні передавання, співвідношення протоколів), прикладних особливостей (шаблони тем або шляхів запитів, схеми даних) і топологічних індикаторів (динаміка кількості унікальних партнерів зв’язку, зміна маршрутизаційних шляхів).

Поняття аномалії відрізняють також від “новизни”. Новизною називають раніше не спостережуваний, але потенційно легітимний режим, який з’являється внаслідок оновлення прошивки, змін конфігурації чи впровадження нової бізнес-

функції; такий режим спершу реєструється як аномальний, однак після верифікації експлуатаційною командою має бути інкорпорований у еталонну модель, інакше система генеруватиме хибні спрацювання. Для IoT це критично через високу мінливість середовищ і практику масових OTA-оновлень: коректна робота механізмів “навчання на місці” та керованого перенавчення визначає баланс між чутливістю до атаки і стійкістю до експлуатаційних змін.

Аномальність у IoT багаторівнева: вона може фіксуватися на рівні окремого пакета (порушення прапорців, некоректна фрагментація), на рівні потоку або сесії (нестандартна тривалість, збій у послідовності запитів), на рівні пристрою (відхилення від власного історичного профілю), кластера вузлів (синхронні сплески телеметрії) і навіть всієї мережевої топології (поява надцентралізованих “хабів” з’єднань). Від вибраної гранулярності залежить і методологія виявлення: точкові відхилення зручно фіксувати локальними порогами та ізоляційними моделями, а для послідовнісних і колективних проявів потрібні підходи, що відображають часову залежність і кореляцію між вузлами.

Усі ці аспекти підкреслюють, що поняття аномалії не зводиться до одного числа чи правила; це організований підхід до відрізнення значущих відхилень від шуму, який інтегрує статистику, семантику протоколів та доменні знання про роль і обмеження конкретних IoT-компонентів.

Класифікація аномалій у трафіку IoT-мереж доцільна як багатовимірна таксономія, яка описує відхилення одночасно з кількох поглядів: часової динаміки, мережевого рівня, масштабу впливу, причин виникнення, спостережності та семантики протоколів. Такий підхід дозволяє узгодити методи аналізу з природою відхилення: одна й та сама подія може бути “точковою” за часом, “топологічною” за впливом на граф з’єднань і “контекстуальною” з огляду на режим роботи вузла.

З погляду часової структури вирізняють миттєві імпульсні порушення, короткочасні транзиції між режимами, тривалі послідовнісні відхилення та повільні дрейфи параметрів. Імпульсні ефекти характерні для одиничних збоїв або помилок передачі; транзиції виявляють зміну робочого профілю пристрою

(наприклад, подвоєння частоти телеметрії); послідовнісні аномалії проявляються як стійке відхилення протягом вікна часу (типово для автоматизованих атак або деградації мережі); дрейфи відображають повільне “сповзання” показників через старіння елементів або зміну середовища. Кожен із цих класів потребує іншої логіки детекції: від локальних порогів і ізоляційних моделей до аналізу змін розподілів та моделювання послідовностей.

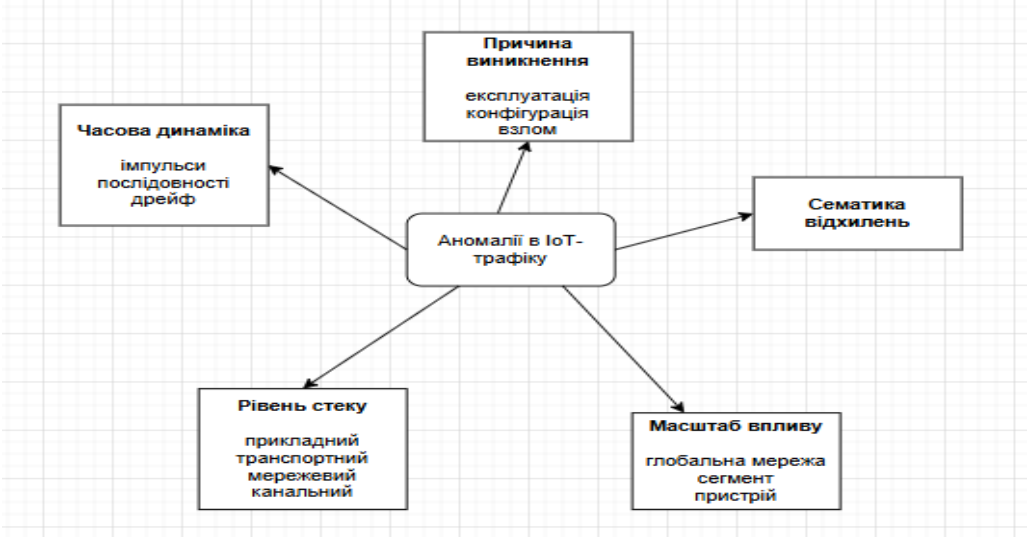


Рис. 1.2 Узагальнена класифікація аномалій у трафіку IoT

У вертикалі мережевого стеку аномалії описують на фізичному та канальному рівнях (нестандартна інтенсивність передач, зростання помилок, порушення сіткової топології), на мережевому й транспортному (нетипова динаміка сесій, фрагментації, прапорці TCP, співвідношення UDP/TCP, лавини повторних передавань) та на прикладному рівні (порушення схем даних, неочікувані шляхи запитів і теми публікацій, зміни QoS і частоти керувальних команд). Додатково виділяють площину керування життєвим циклом пристроїв: підозрілі OTA-оновлення, повторне “бутстрапування”, масові перереєстрації. Стратифікація за рівнями визначає і доступні ознаки: у шифрованих каналах корисним стають метадані та часові профілі, тоді як семантичні порушення легше фіксувати на шлюзах чи в системах керування.

За масштабом впливу корисно розрізняти локальні відхилення для окремого пристрою, сегментні події в межах підмережі або домену

радіопокриття та глобальні явища, які охоплюють значну частину інфраструктури. Локальні випадки добре виявляються через персофіковані еталони “нормальної” поведінки, тоді як сегментні та глобальні потребують кореляційного аналізу між вузлами та відстеження еволюції графа з’єднань у часі.

Класифікація за причиною виникнення допомагає розрізняти експлуатаційні збої, конфігураційні помилки, зловмисні дії та адміністративні події. Експлуатаційні аномалії пов’язані з деградацією елементів, зашумленими каналами чи зміною довкілля; конфігураційні випливають із некоректних інтервалів, адресації, ключів або політик доступу; зловмисні події включають сканування, брутфорс, створення бот-мереж і ін’єкцію даних; адміністративні - це масові законні операції (наприклад, оновлення прошивок), що короткочасно імітують відхилення. Така грань важлива на етапі тріажу: однакові за формою сплески можуть вимагати кардинально різної реакції.

З огляду на спостережність доцільно відрізняти метадані-орієнтовані аномалії, які фіксуються без доступу до payload (ритміка, розміри, темпи, топологія), пейлоад-орієнтовані порушення, що потребують інспекції змісту (схеми, семантика, послідовності викликів), і гібридні випадки, де мережеві метрики узгоджуються з логами брокерів, шлюзів і систем керування. У практиці IoT це визначає місце розміщення сенсорів безпеки: або в мережі на магістралі, або на краю - ближче до джерела даних.

Окремою віссю є семантика відхилення: точкові аномалії - одиничні спостереження з низькою правдоподібністю; контекстуальні - події, що стають нетиповими лише в певному режимі (час доби, роль пристрою, очікувана періодика); колективні - послідовності номінально припустимих дій, які в сумі порушують закономірність (наприклад, тривалий відхід від базового рівня або синхронізація багатьох вузлів). Саме ця вісь найсильніше впливає на вибір алгоритмічної парадигми: від одномоментних статистичних тестів до моделей послідовностей і графових підходів.

Нарешті, корисно відрізняти детерміновані та стохастичні аномалії. Перші порушують однозначні правила або політики (звернення на заборонений домен, некоректний сертифікат), другі виявляються лише як відхилення статистики або геометрії багатовимірного простору ознак. На цій межі лежить і поняття “новизни”: поява раніше не спостережуваного, але легітимного режиму, який необхідно інкорпорувати у базову модель, аби уникнути хибних спрацювань у майбутньому.

Запропонована класифікація не є взаємовиключною - навпаки, її сила в композиції. Описуючи аномалію одночасно за кількома вимірами, ми отримуємо придатний до дії профіль: де саме в стеку вона виникає, який має часовий малюнок, на що впливає і як її найкраще спостерігати. Це, своєю чергою, формує вимоги до інженерії ознак і підказує, чи обрати локальне порогування, ізоляційні моделі, аналіз змін, рекурентні нейромережі або графові методи для подальшої детекції у середовищах IoT.

У середовищі IoT витоки аномальної поведінки рідко мають одну причину; зазвичай це накладання технічних, організаційних і середовищних чинників, які взаємно підсилюють одне одного. На рівні пристрою першоджерелом стають обмеження апаратури та енергобюджету: деградація батареї змінює частоту пробуджень, приводячи до нерівномірних часових інтервалів; старіння сенсорного елемента породжує дрейф показників; перегрів або конденсат дають короткі серії “збоєвих” значень, що у статистиці виглядають як імпульсні відхилення. До цього додаються помилки прошивки: переповнення буферів, некоректні таймери, витоки пам’яті під навантаженням - усе це відбивається на тривалості сесій, кількості повторних передавань і структурі прикладних повідомлень.

Мережеве оточення створює власний пласт причин. У бездротових сегментах коливання рівня сигналу, перешкоди, зміна щільності вузлів у mesh-топології або локальні “пляшкові горлечка” на шлюзах провокують стрибки затримок, втрати та фрагментацію кадрів; у транспорті ці ефекти проявляються хвилями повторних передач і зрушенням частки UDP/TCP. До них приєднуються

часові фактори - розсинхронізація годинників, збій NTP чи некоректне обчислення часових вікон - які порушують сезонні шаблони телеметрії і зміщують “норму”, роблячи легітимні оновлення даних схожими на колективні аномалії.

Окремою групою є джерела на рівні керування життєвим циклом: масові OTA-оновлення, ротація ключів і сертифікатів. Якщо ці операції не узгоджені з моніторингом, системи спостереження реєструватимуть лавиноподібні перереєстрації, стрибки обсягу службового трафіку та нетипові звернення до керівних API. Навіть штатні події - наприклад, зміна частоти телеметрії після встановлення нового профілю - без відповідного маркування у логах сприйматимуться як порушення періодики. Зі свого боку, хмарні бекенди та брокери повідомлень стають точками концентрації ризику: тимчасова деградація черг, зміна політик QoS, перегляд шардінгу чи плану масштабування породжують глобальні патерни відхилень, які помилково можна приписати периферії.

Не менш значущими є організаційні фактори. Недостатня сегментація між IT- і OT-мережами, відсутність повного інвентарю активів, типові облікові записи, слабкі практики керування секретами - усе це збільшує площину атаки і водночас ускладнює інтерпретацію подій: одна й та сама картина в трафіку може відповідати як помилці конфігурації, так і спробі зловмисника закріпитися у системі. Додайте до цього людський чинник - ручне “підкручування” параметрів, тимчасові обхідні схеми під час ремонтів, тестові стенди у продуктивній мережі - і ви отримаєте ще один клас “штучних” аномалій, які потребують знання контексту експлуатації.

Зловмисні джерела зазвичай експлуатують перелічені слабкості. Автоматизовані перебори облікових даних, сканування відкритих інтерфейсів керування, використання відомих вразливостей у веб-панелях та службах віддаленого доступу створюють характерні “гребінці” з’єднань і зміну топології контактів; після компрометації пристрої вступають у координацію через C2-канали, що проявляється у корельованих сплесках сеансів і уніфікації шаблонів

прикладних запитів. Натомість більш “тихі” операції - підміна прошивки, приховане проксування трафіку на шлюзах, маніпуляції сертифікатами - залишають тонкі сліди: зміну розмірів пакетів, метаданих шифрування, статистики рукопотискань і часових “ритмів” повторних спроб.

Нарешті, на стику ланцюга постачання виникають системні джерела аномалій: партії пристроїв із однаковою помилкою прошивки, несумісність версій бібліотек, дефекти партій радіомодулів, а також помилки в конвеєрі збірки, які випадково вмикають режим налагодження чи залишають тестові облікові записи. Такі ситуації створюють “сегментні” картини відхилень, що точно окреслюють конкретну ревізію, модель або часовий інтервал розгортання.

З методичного погляду ключове не лише перелічити потенційні джерела, а й прив’язати кожне до спостережних індикаторів на відповідному рівні стеку і до операційного календаря подій. Тільки поєднання телеметрії мережі, логів керування, даних про зміни конфігурації та відомостей про супровід дозволяє відрізнити справжню загрозу від безпечної “новизни”, а інколи й виявити, що аномалія - лише симптом, за яким стоїть глибинна причина в іншому компоненті екосистеми.

1.4 Методи виявлення аномалій: класичні підходи та сучасні алгоритми

У практиці моніторингу трафіку IoT-мереж виділяють три базові парадигми виявлення відхилень: сигнатурну, статистичну та поведінкову. Кожна з них розв’язує свою частину задачі, спирається на різні уявлення про “норму” і “загрозу” та по-різному поводиться в умовах шифрування, гетерогенності протоколів і обмежених ресурсів пристроїв. Правильне їх поєднання визначає здатність системи одночасно відсікати рутинні події, швидко фіксувати відомі атаки і чутливо реагувати на нові сценарії, що раніше не траплялися.

Сигнатурний підхід ґрунтується на порівнянні поточного трафіку або подій із наперед визначеними шаблонами - послідовностями байтів, комбінаціями полів протоколів, регулярними виразами для прикладних запитів або умовами станового автомата. Його сила - у точності й передбачуваності: якщо подія відповідає сигнатурі, оператор отримує зрозуміле пояснення “яка саме загроза спрацювала” і може відтворити її причину. Для IoT це зручно на межі шлюзів та хмарних компонентів, де проходять повторювані службові транзакції і керувальні команди, а також у сценаріях, де необхідні суворі політики відповідності. Водночас сигнатурний підхід чутливий до варіацій і легко обходиться невеликими модифікаціями трафіку; він майже не бачить “нульових днів”, потребує постійного поповнення бази правил і суттєво втрачає огляд при наскрізному шифруванні, коли доступними лишаються тільки метадані. У розріджених та періодичних потоках, типовим для IoT, сигнатури часто доводиться “виносити” з payload на рівень сеансових характеристик або типових послідовностей подій.

Статистичний підхід моделює ймовірнісну картину нормальної роботи за обраним набором ознак і виявляє події з малою правдоподібністю. Це можуть бути параметричні моделі розподілів, непараметричні щільності, ковзні оцінки з адаптивними порогами, методи детекції змін та сезонно-трендові моделі часових рядів. Для IoT-мереж він природний, оскільки працює по метаданих, не потребує доступу до вмісту пакетів і дозволяє врахувати періодичність телеметрії, duty-cycle, асиметрію “вгору/вниз” та бурстовість подієвих повідомлень. Слабким місцем лишається калібрування: сезонність і дрейф параметрів легко продукують хибні спрацювання, якщо модель не адаптується до змін режиму або не отримує інформацію про планові операції (OTA-оновлення, перепровізіонування). Додатковий виклик - дисбаланс класів: аномалії рідкісні за визначенням, тому критерії мають бути чутливими до “довгих хвостів” і нестабільних хвилинних вікон.

Поведінковий підхід описує “звичку” мережі - профілі активності пристроїв, пар зв’язку, сервісів і топологічних структур - і шукає відхилення від

цих профілів. У сучасних реалізаціях це переважно методи машинного навчання: від простих кластеризацій та ізоляційних алгоритмів до автоенкодерів реконструкції, марковських і рекурентних моделей для послідовностей, а також графових методів, що відстежують еволюцію взаємозв'язків між вузлами. Для IoT поведінкова парадигма особливо приваблива, бо дозволяє будувати "пер-девайс" базові лінії, враховувати роль пристрою і його контекст (час доби, місце розгортання, очікувану періодику) та працювати навіть тоді, коли доступні лише мережеві метадані. Натомість вона вимоглива до якості даних і супровідних процесів: потрібно вирішувати, де саме розміщувати моделі (на краю, на шлюзі, у хмарі), як їх оновлювати без "забування" рідкісних інцидентів, як пояснювати спрацювання операторам і як відділяти справжні загрози від легітимної новизни після змін конфігурації.

У реальних рішеннях ці три підходи не конкурують, а доповнюють один одного. Сигнатури швидко відсікають відомі зловмисні шаблони та помилки конфігурації; статистика надає легку "сітку безпеки", що виявляє грубі порушення ритмів, інтенсивностей і співвідношень протоколів; поведінкові моделі дають глибину, потрібну для фіксації складних, тривалих або координованих відхилень, зокрема там, де шифрування приховує вміст.

Ефективна архітектура виявлення для IoT зазвичай побудована каскадом: швидкі порогові тести на периферії, агрегація та збагачення на шлюзах, складні моделі у хмарі; між рівнями працює зворотний зв'язок, що дозволяє уточнювати пороги, підсилювати сигнатури й ретренувати поведінкові профілі. Такий підхід мінімізує хибні тривоги, зберігає чутливість до невідомого та поважає суворі обмеження енергії й обчислювальних ресурсів, властиві екосистемам Інтернету речей.

Статистичні методи виявлення аномалій у трафіку IoT спираються на формалізоване уявлення про "норму" як про розподіл ознак і часові закономірності сигналу. Для телеметрії з розрідженими подіями, нерівномірними інтервалами та сезонністю (добовою чи тижневою) особливо корисно комбінувати прості описові моделі розподілів із часовими моделями,

здатними врахувати автокореляцію. У практичному конвеєрі це зазвичай означає побудову базових гістограм, обчислення відхилень від центру розподілу у вигляді σ-метрик та моделювання нормальної динаміки за допомогою ARIMA з наступним аналізом залишків.

Гістограма виконує роль первинної “картки здоров’я” для обраної метрики: розміру пакета, інтенсивності вікна, міжпакетних інтервалів або ентропії заголовків. Ключ - коректний вибір бінінгу та часової стабільності. Для мереж із duty-cycle і “вибуховістю” подій типові асиметричні чи багатомодальні розподіли; у такій ситуації класичні припущення нормальності не працюють, тому доцільно зберігати окремі профілі для різних режимів (наприклад, день/ніч або робочі/вихідні) і будувати гістограми на ковзних вікнах, щоб не втрачати чутливість до повільного дрейфу. Аномалією вважається поява масиву спостережень у “рідкісних” хвостах, зміщення моди або раптове виникнення другої моди; ці ефекти відбивають структурні зміни в поведінці пристрою чи сегмента мережі і сигналізують про необхідність глибшої перевірки. Гістограми придатні також для валідації будь-якої наступної моделі: якщо після базової фільтрації розподіл залишків стає ближчим до симетричного та стаціонарного, ми рухаємось у правильному напрямі.

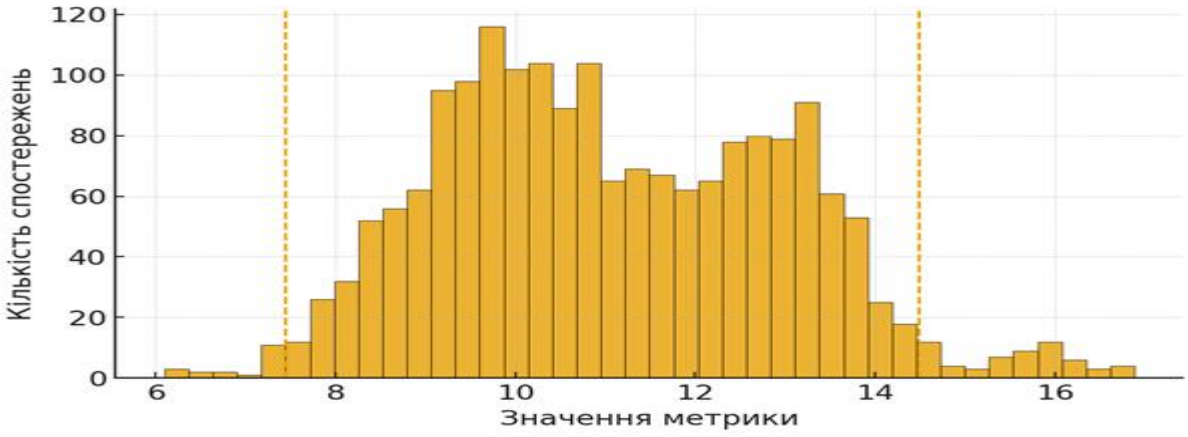


Рис. 1.3 Гістограма метрики трафіку IoT

Методи σ-відхилення переносять інтуїцію контрольних карт на IoT-телеметрію: ми обчислюємо центр (середнє або, у розріджених даних, медіану) та розкид (стандартне відхилення або, для стійкості до викидів, MAD), після чого

позначаємо як аномальні спостереження, що виходять за адаптивні межі. У потоках із сезонністю межі варто робити умовними: зберігати окремі пороги для типових “режимів доби” або застосовувати ковзні оцінки з експоненційним згладжуванням, які швидко підлаштовуються до зміни рівня, але не “проковтують” короткі імпульсні порушення. Перевага σ-підходу - прозорість і пояснюваність для експлуатаційної команди; його обмеження - чутливість до зміни режиму і багатомодальності: без стратифікації контексту він або “дзвенить” хибними тривогами, або, навпаки, стає занадто інертним. Тому на практиці σ-правила застосовують не до сирих пакетів, а до агрегованих ознак (наприклад, “кількість унікальних IP за 5 хвилин”, “частка повторних передавань у вікні”, “стабільність міжпакетних інтервалів”), для яких легше забезпечити стаціонарність.

ARIMA доповнює попередні інструменти там, де вирішальною є часово-послідовна структура. Модель, сфокусована на автокореляціях і сезонних компонентах, дає прогнози “нормальної” траєкторії метрики та дозволяє перевести задачу детекції в площину аналізу залишків: якщо різниця між фактом і прогнозом систематично перевищує довірчі межі, ми маємо справу з аномальним режимом. Це зручно для інтенсивності трафіку, довжини сесій, частоти керувальних команд і навіть для ентропії полів заголовків, якщо вона демонструє регулярні коливання. У застосуванні до IoT важливо підготувати ряд: зняти тренд і сезонність (або явно їх змоделювати), перевірити стаціонарність, підібрати порядки авторегресії та ковзної середньої, а після навчання - обов’язково провести діагностику залишків (відсутність автокореляції, приблизна нормальність), оскільки саме залишки стають “сигналом аномальності” у подальшому моніторингу [13]. Практично модель або оновлюють на ковзному вікні, або ретренують при виявленні стабільних змін режиму, що дає баланс між чутливістю до нових шаблонів і стабільністю прогнозів.

.

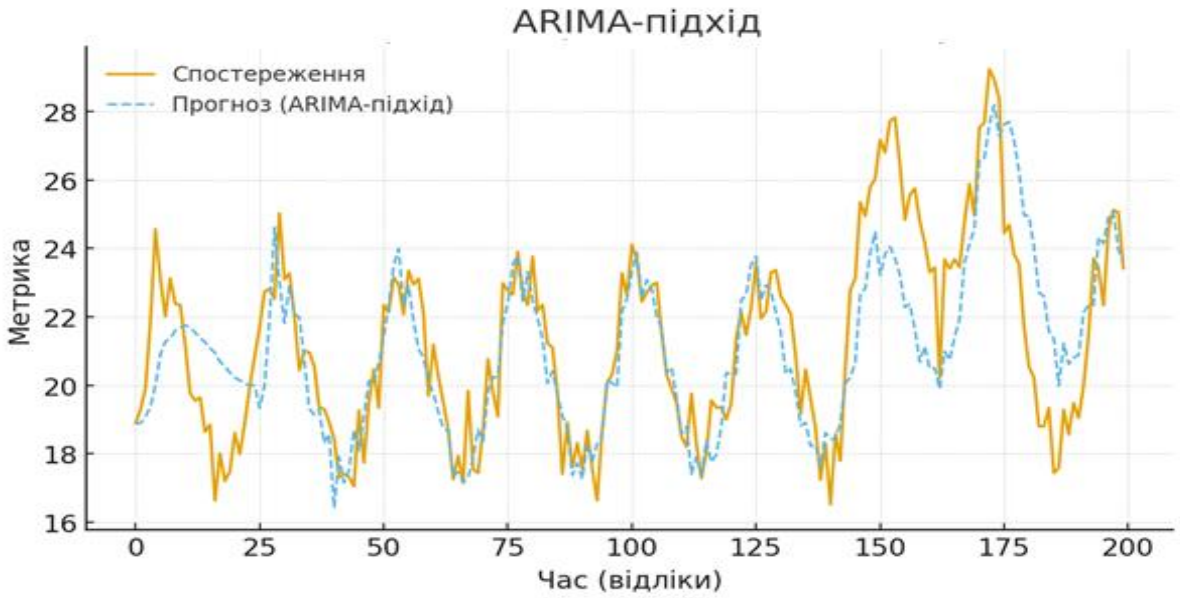


Рис. 1.4 ARIMA-підхід: порівняння факту та прогнозу

Композиція цих трьох інструментів утворює послідовний конвеєр: гістограми забезпечують базову візуальну та кількісну санітарну перевірку розподілів і підказують, де необхідна стратифікація; σ-пороги дають швидку й пояснювану сигналізацію для агрегованих метрик; ARIMA моделює часову структуру, генерує прогнози і перетворює складні патерни на діагностично корисні залишки. У сукупності це підвищує стійкість детекції до шифрування (працюємо з метаданими), до дрейфу (адаптивні оцінки) і до сезонності (явне моделювання), що особливо важливо для гетерогенних і динамічних IoT-мереж [13].

1.5. Застосування машинного навчання у виявленні аномалій в IoT-мережах

Машинне навчання у широкому розумінні - це клас методів штучного інтелекту, які будують моделі на основі даних і покращують якість рішень із досвідом, не будучи жорстко запрограмованими під кожне правило. У контексті аналізу мережевого трафіку IoT це означає побудову алгоритмів, що з даних телеметрії, метаданих з’єднань та журналів подій вивчають закономірності

“нормальної” поведінки та відхилень, аби автоматично сигналізувати про потенційні інциденти або деградації режимів [14].

Так Supervised learning (з учителем) передбачає, що кожний приклад у навчальній вибірці має цільову мітку - клас (наприклад, “норма” чи “атака”) або числове значення, яке потрібно передбачити. Модель навчається мінімізувати помилку між прогнозом і відомою “правдою”, а отже здатна надійно розрізняти шаблони, які вже були представлені під час навчання. Для IoT-детекції це корисно, коли існує накопичена історія інцидентів із точним маркуванням (наприклад, типові сигнатурні атаки на протоколи керування, відомі аномальні послідовності запитів). Обмеження очевидні: отримати повні та збалансовані мітки важко, клас “аномалія” зазвичай рідкісний, а нові сценарії (zero-day) без попередніх прикладів лишаються поза “зоною комфорту” моделі.

A Unsupervised learning (без учителя) відмовляється від міток і зосереджується на геометрії та статистиці даних: кластери, щільності, “віддаленість” спостережень від більшості, стабільність часових профілів. Для аномалій у трафіку це природний вибір, адже системі часто невідомо, що саме слід вважати “загрозою”; натомість можна оцінювати правдоподібність спостережень відносно вивченої “норми” та підсвічувати рідкісні або нові режими. Такі підходи краще масштабуються й не потребують розмічених вибірок, але вимагають обережної роботи з сезонністю, дрейфом режимів і контекстом (день/ніч, робочі/вихідні), щоби не змішувати легітимну новизну з фактичною загрозою.

Як послідовне прийняття Reinforcement learning (підкріплення) формулює задачу рішень агентом, який взаємодіє із середовищем, отримує винагороди й навчається політики дій, що максимізує очікувану корисність у довгостроковій перспективі. На відміну від класифікації або кластеризації, тут результатом є стратегія реагування: наприклад, адаптивне керування порогами та пріоритетами обробки подій на шлюзах, вибір рівня агрегації або маршрутизації для балансування між затримкою і точністю, контекстне вмикання “глибших” моделей лише за появи ознак ризику. У мережах IoT це привабливо через змінні

умови каналів, енергетичні обмеження й необхідність компромісів між хибними тривогами та пропусками; водночас RL потребує ретельно спроєктованої функції винагороди й безпечних механізмів експериментування, аби уникнути ризикових дій у продуктивному середовищі.

На практиці ці парадигми не конкурують, а взаємодоповнюють одна одну: наприклад, безучительські моделі формують базові профілі “норми” і підсвічують кандидати на аномалії, далі частина спрацювань перевіряється експертами і перетворюється на розмічені приклади для навчання супервізованих класифікаторів, а підкріплення використовується для динамічного налаштування політик, що мінімізують сукупні втрати від хибних тривог і пропусків у змінних умовах трафіку та навантаження.

Вибір алгоритму машинного навчання для виявлення аномалій у трафіку IoT визначається доступністю міток, типом ознак (часові, транспортні, топологічні, прикладні), режимністю даних (періодичність, сезонність, “сплесковість”) та обмеженнями середовища виконання (край/шлюз/хмара). У типових сценаріях одна й та сама система поєднує кілька підходів: швидку “лінійку” класичних моделей для інтерпретованих рішень, кластеризацію для профілювання “норми” без міток і глибинні мережі для складних часових залежностей.

Таблиця 1.1

Класичні алгоритми (класифікація/кластеризація)

Алгоритм	**Призначення**	**Сильні сторони**	**Обмеження**	**Ключові параметри**	**Де краще в IoT-трафіку**
SVM	Класифікація/маркування аномалій	Ефективний у високовимірних просторах; робастний із ядрами	Погано масштабується на великі вибірки; потребує підбору ядра	kernel, C, gamma	Класифікація сесій/пристроїв за протокольними й часовими ознаками
Random Forest	Класифікація, оцінка важливості ознак	Нечутливий до масштабів ознак; стійкий до переобучення	Великі моделі; гірша інтерпретованість за окреме дерево	n_estimators, max_depth, max_features	Базова детекція відхилень у метаданих; швидкий бенчмарк

Продовження таблиці 1.1

Алгоритм	**Призначення**	**Сильні сторони**	**Обмеження**	**Ключові параметри**	**Де краще в IoT-трафіку**
Decision Tree	Прості правила/пороги	Інтерпретованість; мінімальні ресурси; зручно для edge	Схильність до переобучення без обмежень глибини	max_depth, min_samples_leaf	Локальні правила на шлюзах/AGV/датчиках
K-means	Кластеризація	Швидкий; добре для сферичних кластерів	Потрібно задавати k; чутливий до масштабування ознак	n_clusters, init, n_init	Сегментація пристроїв/поведінки для базових профілів
DBSCAN	Кластеризація щільності + виявлення шуму	Не потребує k; кластери довільної форми; позначає «шум»	Чутливий до eps/min_samples; гірше у високій вимірності	eps, min_samples	Локальні аномалії у сегментах, де k-means дає змішані кластери

Таблиця 1.2

Глибинне навчання (послідовності/аномалії)

Алгоритм	**Призначення**	**Сильні сторони**	**Обмеження**	**Ключові параметри**	**Де краще в IoT-трафіку**
Autoencoder	Безнаглядне виявлення через помилку реконструкції	Виділяє латентні представлення; гнучкий	Потребує ретельної валідації; ризик запам’ятовування шуму	архітектура, bottleneck, регуляризація, threshold	Point-аномалії у флоу/телеметрії, коли міток немає
LSTM	Моделювання часової залежності	Ловить контекст і довгі залежності	Довге навчання; вимогливий до ресурсів	hidden_units, layers, seq_len, dropout	Контекстуальні/послідовні аномалії (послідовності запитів)
CNN (1D)	Локальні шаблони у рядах	Швидка інференція; паралелізується на GPU	Потребує стаціонарного подання; чутливий до вікна	filters, kernel_size, stride, dilation	Короткі патерни в інтенсивності/порт-частотах; попередній фільтр на edge

Незалежно від вибору алгоритму, для IoT ключовими залишаються питання інженерії ознак і операційної придатності: нормування та стабілізація розподілів, формування ковзних вікон без витоку у час, робота з дисбалансом і вибір метрик оцінювання (PR-AUC), керування дрейфом і “легітимною новизною” після OTA-оновлень. На рівні розгортання важливо визначити, де

саме виконувати інференс (край/шлюз/хмара), які моделі можна стискати (прінінг, квантизація, знання-дистиляція) і як організувати цикл донавчання без деградації чутливості. Саме інтеграція цих технічних рішень дозволяє кожному з перелічених алгоритмів розкрити свої сильні сторони в реальних, гетерогенних та динамічних IoT-мережах.

2 ДОСЛІДЖЕННЯ ТА МЕТОДИКА АНАЛІЗУ ТРАФІКУ IoT-МЕРЕЖІ

2.1. Об’єкт та дані дослідження

Об’єктом дослідження в даній кваліфікаційній роботі є корпоративна мережна інфраструктура з виділеним сегментом Інтернету речей (IoT), який забезпечує функціонування систем моніторингу параметрів середовища, контролю та управління доступом, відеоспостереження, обліку енергоресурсів, а також низки сервісів автоматизації інженерних систем будівель. Така інфраструктура є типовою для сучасних організацій, де IoT-пристрої безпосередньо впливають на фізичну безпеку, енергетичну ефективність, безперервність бізнес-процесів та захищеність інформаційних ресурсів. Будь-які порушення у роботі IoT-сегмента, спричинені зловмисною активністю або технічними збоями, можуть мати критичні наслідки, що зумовлює необхідність системного дослідження трафіку та впровадження засобів виявлення аномалій.

Розглянутий IoT-сегмент реалізовано як логічно ізольовану частину корпоративної мережі з використанням окремих підмереж, VLAN та політик міжмережевого екранування. На периферійному рівні функціонує сукупність гетерогенних пристроїв: сенсорів мікроклімату та стану обладнання, лічильників енергоресурсів, мережевих камер відеоспостереження, зчитувачів і контролерів систем контролю доступу, виконавчих модулів (реле, контролерів освітлення та інженерних систем), мікроконтролерних модулів та спеціалізованих IoT-плат.

Пристрої взаємодіють через дротові підключення Ethernet та виділені бездротові сегменти Wi-Fi для IoT, а в окремих випадках - через енергоощадні технології короткого радіуса дії, інтегровані з периферійними шлюзами. На проміжному рівні розташовано IoT-шлюзи та точки доступу, які виконують агрегування трафіку, конвертацію протоколів, застосування локальних політик безпеки та маршрутизацію даних до серверів ядра мережі. На верхньому рівні функціонують сервери моніторингу, брокер MQTT, веб- та прикладні сервіси, бази даних, а також компоненти, інтегровані з хмарними платформами.

Узагальнена архітектура цього середовища представлена у вигляді трирівневої моделі (рис. 2.1), де виділено рівень IoT-пристроїв, рівень IoT-шлюзів та комунікаційного обладнання, а також рівень центральних серверів і хмарних сервісів, між якими організовано захищені канали зв’язку. Така побудова відповідає сучасним підходам до сегментації IoT-інфраструктур, які передбачають мінімізацію довіри до кінцевих пристроїв, контрольовані точки виходу з IoT-сегмента та централізоване застосування політик безпеки [16].

Послідовне впровадження ізоляції, контролю маршрутів та диференційованих зон довіри створює технічне підґрунтя для подальшого аналізу трафіку та виявлення аномалій, але водночас підкреслює складність середовища: велика кількість пристроїв, різні виробники, протоколи, режими роботи та оновлення призводять до високої варіативності легітимної поведінки.

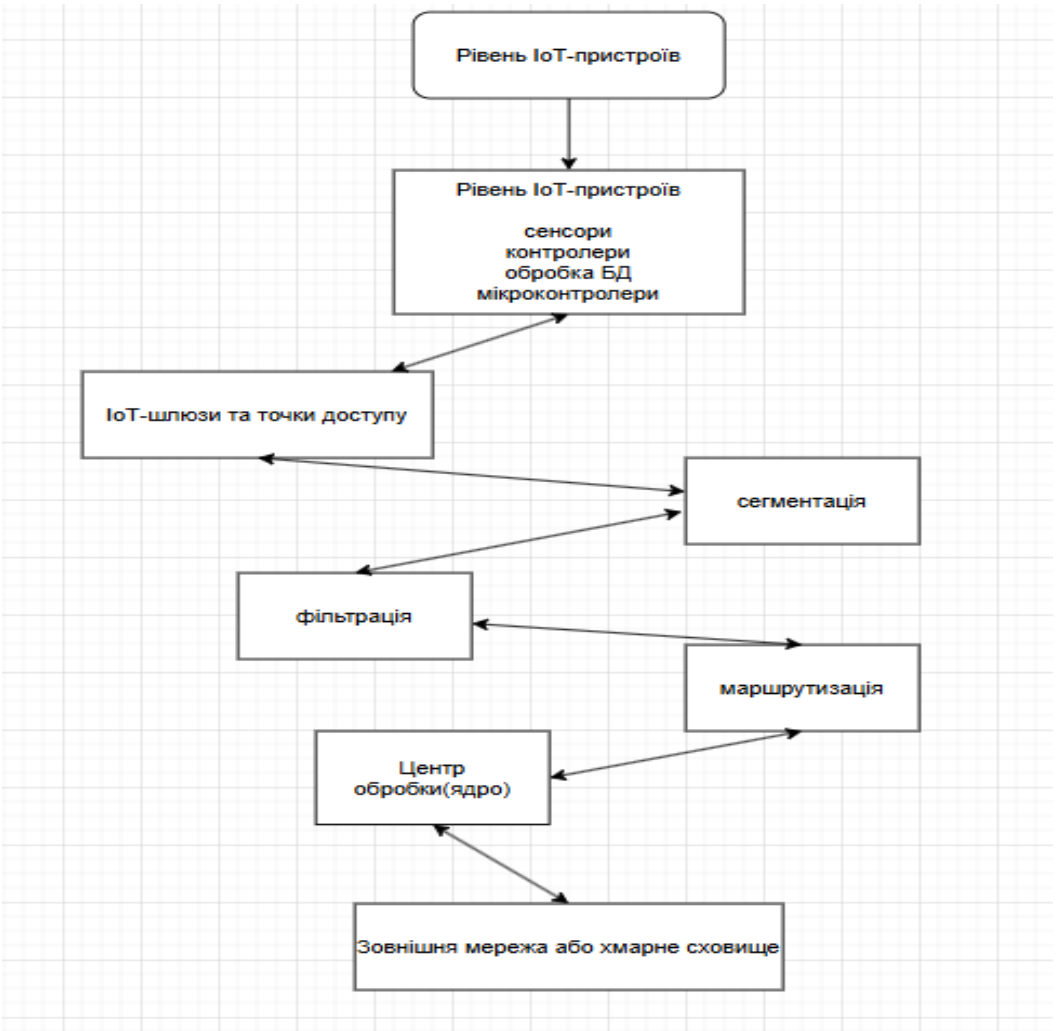


Рис. 2.1 Архітектура IoT-сегмента

Інформаційна база дослідження сформована на основі даних про мережеву активність IoT-сегмента за дворічний період (з січня 2023 року по грудень 2024 року), що дозволяє врахувати сезонні коливання, етапи розширення інфраструктури та зміну політик безпеки. Для подальшого аналізу використовуються: агреговані записи мережевих потоків (у форматі NetFlow/IPFIX або еквівалентному), журнали подій IoT-шлюзів, точок доступу та міжмережевих екранів, логи брокера MQTT та суміжних прикладних сервісів, а також вибіркові фрагменти сирого трафіку, зібрані у контрольні інтервали.

Схематичне відображення інфраструктури збору та консолідації даних наведено на рис. 2.2, де показано основні точки контролю та потоки даних до централізованого сховища. Така конфігурація забезпечує покриття як периферійного, так і магістрального рівнів, що є принципово важливим для коректного відтворення структурних і поведінкових характеристик IoT-мережі.

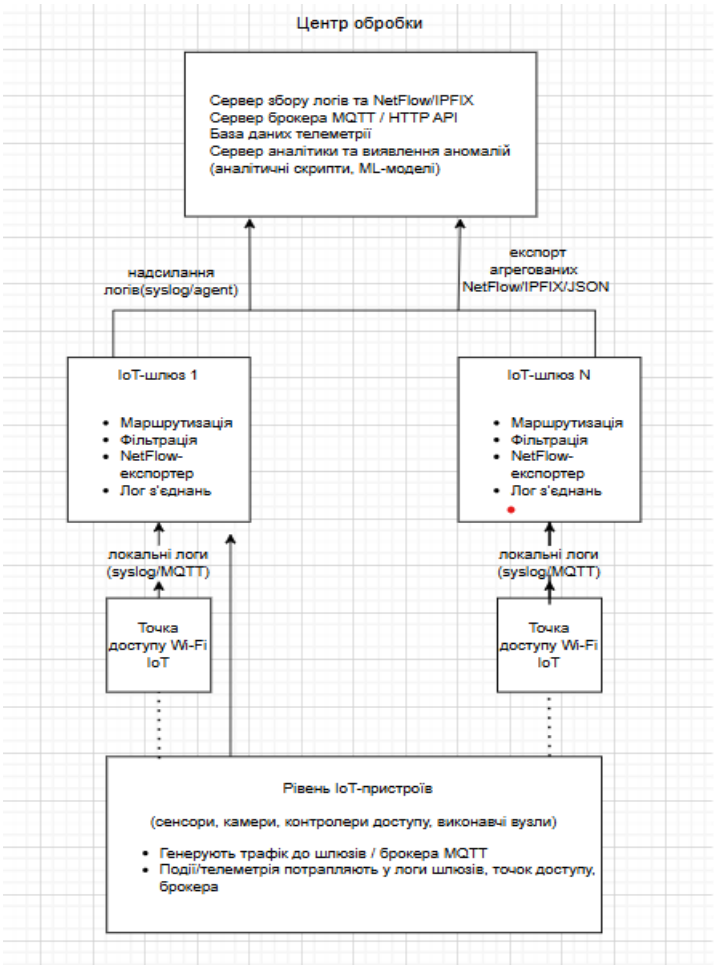


Рис. 2.2 Схема інфраструктури збору та консолідації даних трафіку IoT-мережі

Формування набору даних здійснювалося з урахуванням вимог конфіденційності та коректності подальшого аналізу. Реальні IP-адреси, MAC-адреси, унікальні ідентифікатори пристроїв та чутливі атрибути були піддані псевдонімізації із збереженням стабільних відповідностей у межах досліджуваного періоду, що уможливлює відстеження поведінки окремих вузлів без розкриття їх фактичної належності. Службовий трафік, не пов'язаний безпосередньо з роботою IoT-пристроїв (наприклад, внутрішні технічні резервні копії, службові канали адміністрування), був виключений із подальшого розгляду. В умовах широкого застосування шифрування прикладного рівня основну увагу в дослідженні приділено саме метаданим трафіку - часовим, протокольним, напрямним, сеансовим та топологічним характеристикам, що узгоджується із сучасними підходами до виявлення аномалій в IoT-мережах на основі аналізу поведінкових ознак[23].

З метою забезпечення уніфікованого формату даних для наступних етапів аналізу розроблено процедуру попередньої обробки. Вона включає завантаження експортованих потокових даних, відбір записів, що належать до цільового IoT-сегмента, нормалізацію часових міток, псевдонімізацію ідентифікаторів, а також формування консолідованого набору мережевих подій. Приклад реалізації ключових кроків цієї процедури наведено у лістингу 2.1, де демонструється відбір трафіку виділеної підмережі та заміна реальних адрес на внутрішні ідентифікатори пристроїв.

```
df = pd.read_csv("iot_netflow.csv")

# Відбір записів, що належать до виділеного
# IoT-сегмента (приклад: підмережа 10.10.0.0/16)
iot_df = df[
    df["src_ip"].str.startswith("10.10.") |
    df["dst_ip"].str.startswith("10.10.")
].copy()

# Псевдонімізація IP-адрес для дотримання вимог конфіденційності
ip_map = {}

def pseudo_ip(ip):
    if ip not in ip_map:
        ip_map[ip] = f"dev_{len(ip_map) + 1}"
    return ip_map[ip]

iot_df["src_id"] = iot_df["src_ip"].apply(pseudo_ip)
iot_df["dst_id"] = iot_df["dst_ip"].apply(pseudo_ip)
```

Рис. 2.3 Приклад формування вибірки трафіку IoT-сегмента та псевдонімізації ідентифікаторів пристроїв

IoT-інфраструктура досліджуваного об’єкта являє собою багатокомпонентний комплекс апаратних і програмних засобів, інтегрований у єдиний корпоративний мережевий простір та орієнтований на підтримку автоматизованих сервісів моніторингу й керування. У її склад входять різнорідні кінцеві пристрої, периферійні вузли обробки, комунікаційне обладнання та серверні компоненти, об’єднані загальними політиками доступу і єдиною системою адміністрування. Така інфраструктура сформована поступово, у кілька етапів розгортання, що зумовило наявність обладнання різних виробників, з різними версіями прошивок і наборами підтримуваних протоколів, але з уніфікованими вимогами до їх інтеграції в мережеву та інформаційну архітектуру організації.

Кінцевий рівень IoT-інфраструктури представлений сенсорами, виконавчими пристроями та спеціалізованими модулями, які безпосередньо взаємодіють із фізичним середовищем об’єкта. До цієї групи належать датчики температури, вологості, освітленості, якості повітря, вібрацій і стану обладнання, лічильники теплової енергії, електроенергії та водопостачання, мережеві камери відеоспостереження, зчитувачі і контролери систем контролю та управління доступом, а також виконавчі реле, електронні замки, модулі керування вентиляцією, освітленням та іншими інженерними системами. Значна частина цих пристроїв працює в цілодобовому режимі, генеруючи телеметричні дані з різною інтенсивністю: від періодичних вимірювань з інтервалом у кілька хвилин до безперервних потоків відео або подій доступу. Саме ця різнорідність формує складну мозаїку легітимного трафіку, що потребує індивідуалізованого підходу до визначення нормальної поведінки для кожного класу пристроїв.

Проміжну ланку інфраструктури становлять периферійні IoT-шлюзи, маршрутизатори та точки доступу, які забезпечують зв’язок між кінцевими пристроями та центральними сервісами. До їхніх функцій належать: агрегування трафіку від груп сенсорів і виконавчих модулів, конвертація протоколів з

урахуванням обмежених ресурсів периферійних пристроїв, реалізація політик доступу, базова фільтрація, маркування трафіку для подальшого аналізу, а також маршрутизація даних до ядра корпоративної мережі. Окремі шлюзи виконують роль точок інтеграції з бездротовими технологіями короткого радіуса дії, забезпечуючи прозору взаємодію низькопотужних пристроїв із серверною інфраструктурою. Наявність саме цих проміжних елементів робить їх ключовими позиціями як для забезпечення захищеності, так і для організації моніторингу трафіку та збору статистики.

На верхньому рівні IoT-інфраструктури функціонують сервери, відповідальні за обробку, зберігання та візуалізацію даних, а також за інтеграцію з іншими інформаційними системами організації. До них належать сервер моніторингу та журналювання, брокер MQTT, веб-сервіси для конфігурації та адміністрування IoT-пристроїв, сервери систем контролю доступу, відеореєстратори, аналітичні модулі та бази даних. Частина сервісів реалізована локально у межах корпоративного дата-центру, інша - із використанням хмарних платформ, до яких доступ здійснюється через захищені тунелі та контрольовані точки виходу в Інтернет. Така побудова забезпечує централізоване керування, історичне накопичення даних та можливість впровадження аналітичних рішень, але водночас уводить додаткові вектори потенційного впливу на безпеку через зовнішні сервіси.

З огляду на викладене, IoT-інфраструктура об’єкта характеризується поєднанням високої функціональної насиченості, гетерогенного складу обладнання та жорстких вимог до безперервності роботи. Це створює сприятливе, але водночас складне з аналітичної точки зору середовище для дослідження аномалій у трафіку: будь-яке рішення з виявлення відхилень має враховувати велику кількість типів пристроїв, різні моделі їх поведінки, наявність проміжних вузлів агрегації та багаторівневу структуру мережі. Саме тому подальші підрозділи розділу 2 зосереджуються на формалізованому описі архітектури, джерел даних та характеристик трафіку, що дозволяє перейти від

загального уявлення про інфраструктуру до побудови кількісних моделей нормальної та аномальної активності.

Мережева архітектура досліджуваного IoT-сегмента побудована за принципом багаторівневої сегментації з чітким розмежуванням зон довіри та контрольованими точками перетину з іншими частинами корпоративної інфраструктури. IoT-пристрої об’єднані в окремі підмережі, виділені в рамках VLAN, що дає змогу фізично та логічно відокремити їх трафік від офісних робочих станцій, серверів загального призначення та гостьового доступу. Така ізоляція мінімізує ризики горизонтального поширення атак з менш захищених IoT-пристроїв у критично важливі сегменти мережі та спрощує подальший аналіз трафіку, оскільки потоки зосереджуються у визначених логічних доменах.

Комунікація між IoT-пристроями та центральними сервісами здійснюється переважно через спеціалізовані IoT-шлюзи, які виконують роль пограничних вузлів сегмента. Саме через них реалізується маршрутизація трафіку до серверів моніторингу, брокера MQTT, систем відеоспостереження та служб керування доступом. На цих вузлах застосовуються політики фільтрації, обмеження набору дозволених протоколів і портів, обмеження вихідних з’єднань у зовнішні мережі, а також базові механізми контролю цілісності та коректності трафіку. Доступ до Інтернету для IoT-пристроїв або повністю заборонений, або здійснюється виключно через централізовані шлюзи з глибокою інспекцією трафіку. Така організація дозволяє чітко фіксувати всі міжсегментні взаємодії та зменшує площу потенційних атак.

Ядро корпоративної мережі, до якого підключені IoT-шлюзи, відокремлене від зовнішнього середовища міжмережевими екранами та, за потреби, демілітаризованими зонами, де можуть розміщуватися окремі прикладні сервіси або проксі-компоненти. Взаємодії з хмарними платформами, сервісами віддаленого керування та зовнішніми API здійснюється через захищені канали з попередньо визначених точок виходу, що створює можливість централізованого моніторингу та обмеження цих з’єднань. Схематично така архітектура з відокремленим IoT-сегментом, проміжними IoT-шлюзами, ядром

мережі та зовнішніми сервісами подана на рис. 2.4, який відображає ключові маршрути руху трафіку та позиції для його контролю.

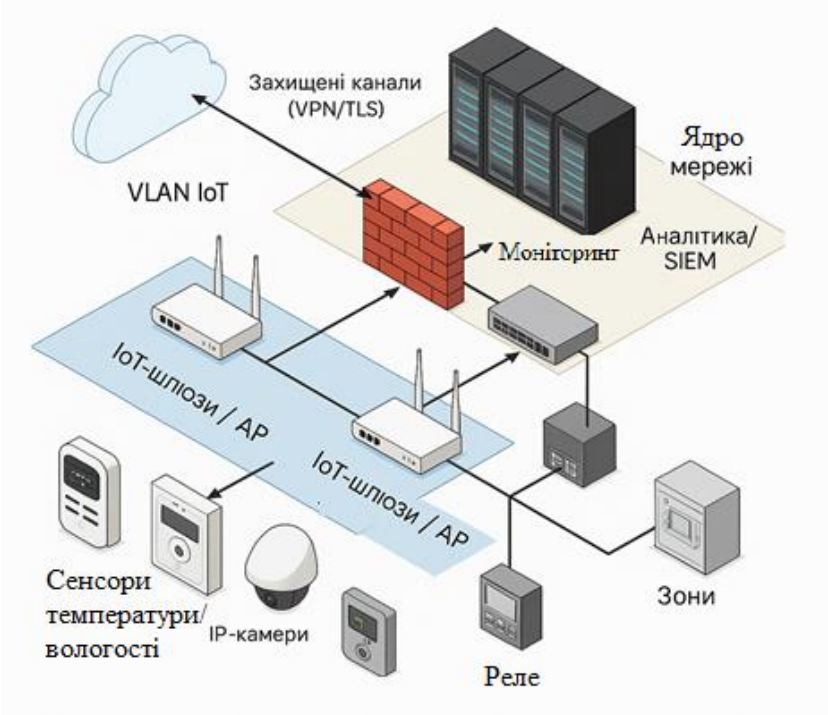


Рис. 2.4 Структура IoT-сегмента корпоративної мережі

Обрані підходи до сегментації й побудови архітектури узгоджуються з актуальними рекомендаціями щодо безпеки IoT-інфраструктур, які передбачають мінімізацію довіри до кінцевих пристроїв, використання окремих доменів для IoT-трафіку, обмеження прямих з'єднань до критичних ресурсів та централізований контроль точок виходу з сегмента [16]. У контексті подальшого дослідження це забезпечує дві ключові переваги: по-перше, підвищується керованість та прозорість потоків даних, по-друге, створюються сприятливі умови для побудови моделей виявлення аномалій, оскільки структура дозволених взаємодій є чітко визначеною і будь-які відхилення від неї можуть інтерпретуватися як потенційно небажані.

Інформаційна підсистема спостереження за IoT-сегментом побудована так, щоб охоплювати як периферійний рівень взаємодій "пристрій → шлюз/точка доступу", так і магістральні маршрути "шлюз → сервер/хмара". Базовим джерелом є агреговані записи мережевих потоків у форматі NetFlow/IPFIX або

їх функціональні аналоги, що експортуються з IoT-шлюзів і прикордонних маршрутизаторів. Потокові записи містять часові мітки початку й завершення з'єднання, ідентифікатори джерела та призначення (у псевдонімізованому вигляді), порти, протокол, тривалість, обсяг переданих байтів і пакетів, а також допоміжні атрибути, зокрема позначки VLAN і TCP-прапорці. Саме така телеметрія забезпечує відтворюваний, узгоджений у часі огляд усіх основних взаємодій усередині сегмента та на його межах, що є критично важливим для побудови статистичних і топологічних ознак подальшого аналізу [18].

Другим класом джерел виступають журнали подій мережевої інфраструктури: syslog-повідомлення від IoT-шлюзів, комутаторів доступу, точок Wi-Fi, а також записи міжмережевих екранів, які фіксують факти дозволу/блокування сесій, зміни правил, збої інтерфейсів, повторні спроби встановлення з'єднань та інші сигнали про якість і стабільність передавання. Для прикладного рівня використовується телеметрія брокера MQTT: події підключення/відключення клієнтів, спроби автентифікації, статистика публікацій і підписок по топіках, ознаки нестандартної активності клієнтів. Ці журнали дозволяють уточнити семантику мережевих подій і співвіднести виявлені відхилення з конкретними сервісами, залишаючись при цьому в межах аналізу метаданих без доступу до змісту зашифрованого корисного навантаження.

Для верифікації протокольних особливостей і калібрування евристик використовуються вибіркові короткі знімки сирого трафіку у контрольні інтервали. Вони не складають основи даних, але допомагають ідентифікувати специфічні шаблони ініціації сесій, характерні тайм-аути та службові послідовності обміну, що важко відновити з агрегованих потоків. З огляду на вимоги конфіденційності, такі знімки зберігаються у захищеному ізольованому сховищі та не містять персональних даних; їх застосування обмежене стадіями налагодження методики.

Інфраструктурно збір даних реалізовано за централізованою схемою. На межових пристроях увімкнено експорт NetFlow/IPFIX із передачею в колектор,

який виконує первинну нормалізацію полів, синхронізацію часових міток за NTP і дедуплікацію записів від різних точок спостереження. Syslog та журнали прикладних сервісів надходять через агентів збору у спільний буфер подій, після чого індексуються в системі централізованого журналювання. Для аналітики часових рядів та побудови ознак використовується сховище з підтримкою зручної агрегації у вікна (time-series/columnar БД або шардований індекс у SIEM-системі). Канали від джерел до колектора захищені, доступ до даних регламентований ролями, а політики зберігання встановлюють різні строки утримання для сирих і агрегованих представлень. Схематичне розміщення компонентів та потоки телеметрії наведені на рис. 2.2.

З позиції змісту даних ключовим є уніфікований словник полів, що забезпечує узгодженість між різними джерелами. Для потоків зберігаються принаймні часові мітки, псевдонімізовані ідентифікатори вузлів-учасників, напрям, порти, протокол, тривалість, обсяги, кількість пакетів, індикатори якості (повторні SYN/RETRANSMIT, FIN/RST), позначки сегментації. Для журналів інфраструктури - тип події, рівень, результат дії (allow/deny), код причини, ідентифікатор пристрою-джерела; для MQTT - clientId, подія (connect/subscribe/publish), topic, короткі лічильники. На етапі консолідації всі часові поля переводяться до єдиного часового стандарту, а ідентифікатори пристроїв зв’язуються зі словником атрибутів класу (сенсор, камера, контролер тощо), що дозволяє надалі будувати профілі “норми” за типами обладнання.

Окремо враховано типові артефакти, що впливають на якість даних: асиметричні маршрути, NAT на межових пристроях, вибірковий семплінг потоків, короткі розриви в телеметрії під час оновлення прошивок. Для мінімізації викривлень застосовано правила об’єднання дублікатів по ключах ”час-пара вузлів-набір портів”, маркування періодів технічних робіт і виключення їх із моделювання “норми”, а також контроль повноти через щоденні звіти про очікувану та отриману кількість записів з кожного джерела. Така дисципліна даних безпосередньо позначається на чутливості детекторів

аномалій та зменшує ризик хибних спрацювань, обумовлених інфраструктурними збоями, а не реальною нетиповою поведінкою вузлів.

У підсумку сформовано консолідований набір метаданих трафіку, що поєднує потокову, інфраструктурну та прикладну телеметрію і придатний для подальшого статистичного та структурного аналізу. Наявність єдиного каналу збору, уніфікованого словника полів і контроль якості даних забезпечують відтворюваність результатів та створюють надійне підґрунтя для побудови простору ознак у підрозділі 2.2 і апробації методів виявлення аномалій у підрозділі 2.3.

Побудова надійної інформаційної бази здійснювалася за відтворюваною процедурою, яка поєднує уніфікацію схем даних, очищення, синхронізацію часу, псевдонімізацію і консолідацію різнорідних джерел у єдиний масив подій. На етапі потокових записів NetFlow/IPFIX, журнали мережевої інфраструктури й телеметрія брокера MQTT переводилися до уніфікованого словника полів із чітко визначеними типами: часові мітки приводилися до UTC з фіксацією відхилень NTP, ідентифікатори вузлів зводилися до внутрішніх ключів, а протокольні атрибути нормалізувалися до узгоджених категорій. Такий підхід зменшує дрейф форматів між різними пристроями та полегшує подальшу побудову багатовимірного простору ознак, орієнтованого на аналіз метаданих, що є рекомендованою практикою для зашифрованих IoT-комунікацій.

Очищення даних передбачало усунення дублікатів і неповних записів, що виникають через одночасний експорт потоків із кількох точок спостереження або через асиметричні маршрути. Дедуплікація виконувалася за композиційним ключем “часовий інтервал - пара вузлів - напрям - набір портів – протокол” з урахуванням похибки синхронізації, після чого залишалися лише репрезентативні записи з найбільш повними лічильниками. Паралельно застосовувалася фільтрація службового трафіку, який не відображає поведінку IoT-пристроїв (резервні копії, внутрішні оновлення, технічні heartbeat-канали адміністративних систем), що дозволяє уникати помилкових інтерпретацій під час побудови профілів “норми”. Для полів із поодинокими пропусками

використовувалася м’яка стратегія: заповнення з контексту джерела або відкидання запису лише за відсутності критичних атрибутів (час, джерело, призначення, протокол).

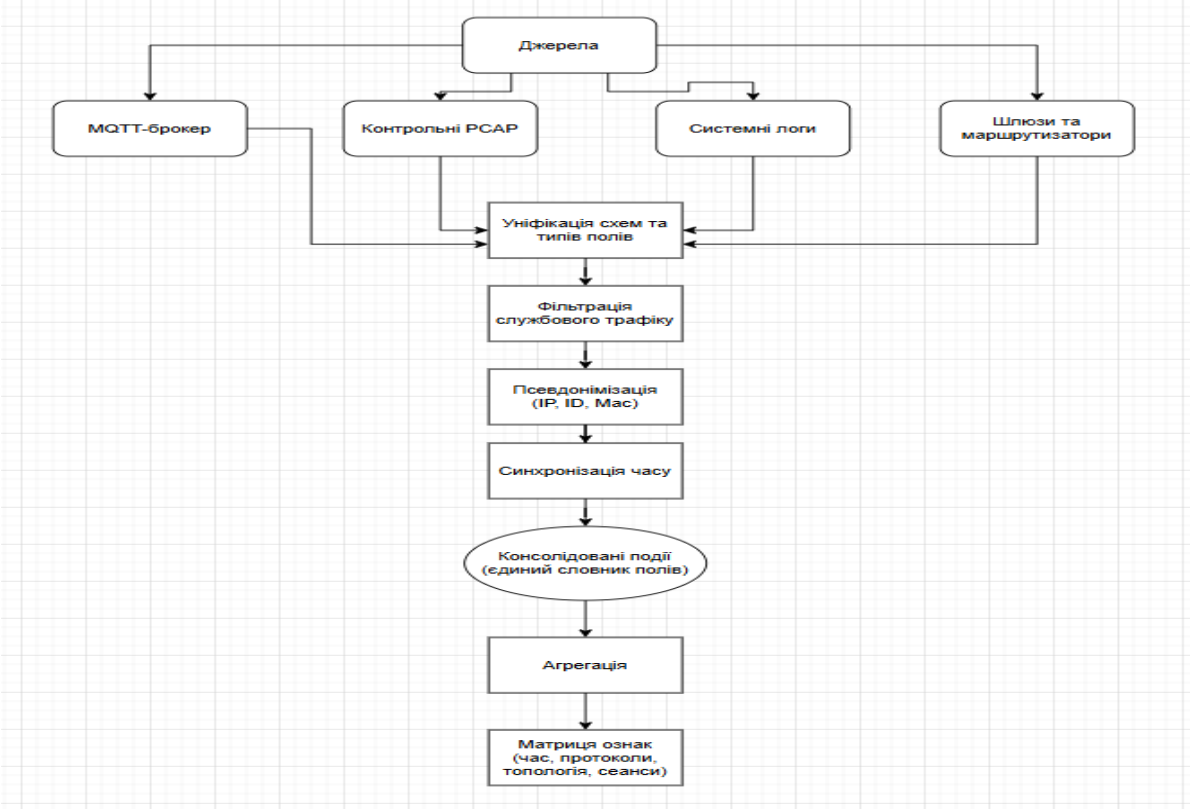


Рис. 2.5 Агрегація подій у часові вікна

Псевдонімізація ідентифікаторів виконувалася детерміновано з підтримкою стабільних відповідностей упродовж всього періоду спостереження: одна й та сама адреса завжди відображалася в один і той самий внутрішній ключ, що дає змогу відстежувати поведінкові траєкторії окремих вузлів без розкриття їх фактичної належності. Додатково формувався довідник класів пристроїв (сенсор, камера, контролер доступу, шлюз тощо), який з’єднувався з консолідованими подіями за внутрішнім ключем; це забезпечує можливість контекстної стратифікації під час навчання профілів “норми” та аналізу відхилень. В умовах високої частки шифрованих каналів саме така робота з метаданими дає найбільшу віддачу з точки зору якості виявлення аномалій при дотриманні вимог конфіденційності.

Синхронізація часу відігравала ключову роль, оскільки навіть незначні розбіжності між джерелами призводять до хибних висновків у часовому аналізі. Для цього використовувалася інформація з NTP-журналів і контрольні “маяки” подій (наприклад, планові перезавантаження обладнання), за якими оцінювалися та компенсувалися систематичні зсуви. Періоди оновлень прошивок, зміни конфігурацій і технічних робіт маркувалися як “нестабільні” та виключалися з формування еталонних профілів; натомість вони зберігалися у базі для подальшої оцінки стійкості алгоритмів до структурних змін.

Після очищення і нормалізації події переводилися до єдиного часово-віконного подання, що слугує “будівельним матеріалом” для подальших ознак. Для кожного вікна (у роботі застосовувалися погодинні та, за потреби, п’ятихвилинні інтервали) обчислювалися агреговані кількісні характеристики на рівні вузла або групи вузлів: сумарні обсяги, кількість сесій, число унікальних напрямків взаємодії, частки протоколів, індикатори якості передавання (частота повторних спроб, обірваних сесій), а також елементи топологічної структури (ступінь вузла, зміна ролі вузла-концентратора тощо). Таке віконування згладжує випадкові піки, вирівнює різні масштаби потоків і дозволяє чітко порівнювати поточний стан з історичним контекстом, як це рекомендується у практичних дослідженнях з детекції аномалій у мережевих середовищах [19].

Якість консолідованого набору контролювалася через щоденні звіти повноти та узгодженості: для кожного джерела фіксувалася очікувана і фактична кількість записів, частка відкинутих подій, середній і максимальний часовий зсув, частота дублікатів. У випадку деградації показників запускалася процедура повторної консолідації за відповідний період із уточненими параметрами NTP-вирівнювання та дедуплікації. Результатом усіх кроків стала стандартизована таблиця подій і відповідне віконне представлення, готове до статистичного й структурного аналізу у підрозділі 2.2 та до пілотної апробації детекторів у підрозділі 2.3. Така дисципліна підготовки даних мінімізує ризик технічних артефактів, підвищує відтворюваність результатів і створює належне підґрунтя

для подальшого моделювання “норми” та виявлення відхилень у трафіку IoT-мережі[23].

2.2 Аналіз характеристик IoT-трафіку

У межах дворічного періоду проаналізовано часові ряди сумарного обсягу трафіку IoT-сегмента та кількості активних пристроїв із погодинною агрегацією й подальшим згладжуванням ковзним середнім (вікна 24 год для добових закономірностей і 7 днів для тижневої сезонності). Під “активним” розуміємо пристрій, що у вибраному інтервалі сформував принаймні один мережевий потік або подію прикладного рівня (наприклад, MQTT-публікацію). Таке визначення забезпечує однаковий підхід до різнорідних класів вузлів і коректну порівнянність динаміки навантаження з динамікою популяції пристроїв.

Декомпозиція часових рядів на тренд, сезонну й залишкову компоненти виявила стабільну тижневу періодику: нижчі значення у вихідні та підвищення у робочі дні, що особливо виразно проявляється для підсистем контролю доступу та відеоспостереження. На добовому горизонті спостерігаються виражені “денні хвилі” активності з піками у ранкові та вечірні проміжки; сенсорні підсистеми формують рівномірніший фон із невеликими амплітудами. Структурні злами тренду (step-зсуви) збігаються з періодами розширення інфраструктури: збільшенням кількості шлюзів, підключенням нових камер або введенням додаткових сенсорних ліній. Такі епізоди марковано як “режими зміни”, і вони надалі не використовуються для навчання еталонних профілів “норми”, аби уникнути упередженості.

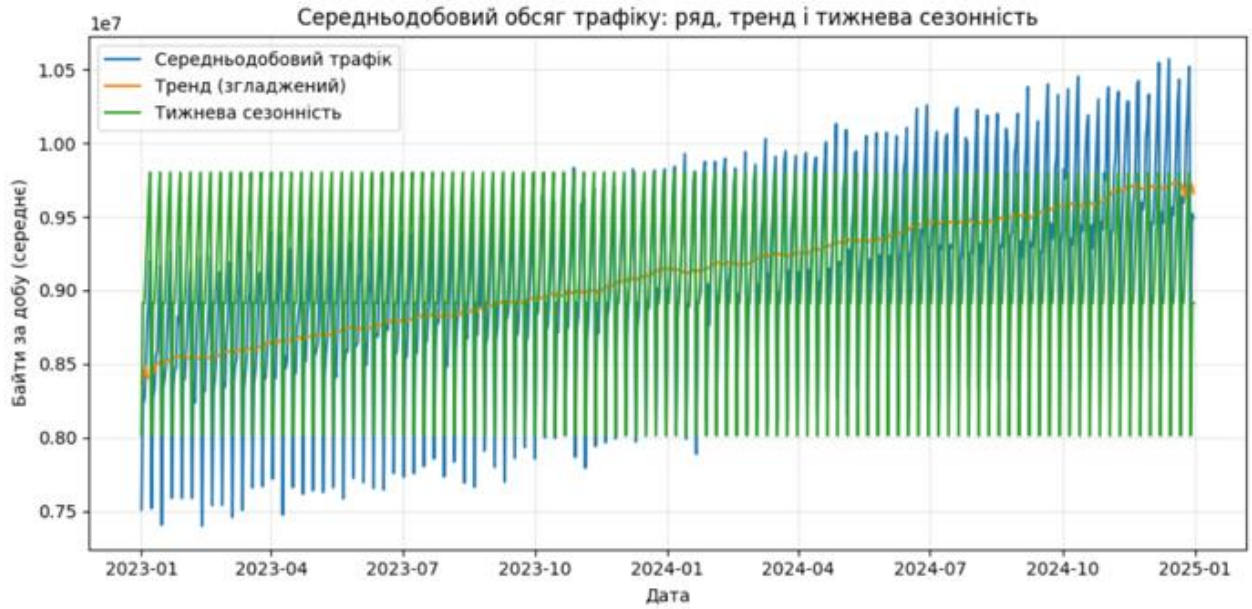


Рис. 2.6 Графік середньодобового обсягу з трендом і тижневою сезонністю

Кореляційний аналіз між “обсягом трафіку” та “числом активних пристроїв” показав очікувану позитивну залежність із різною силою для різних підсистем. Для сенсорів інкремент кількості вузлів частіше призводить до майже лінійного зростання обсягу (кожен додатковий пристрій дає невеликий, але стабільний внесок), тоді як для відеопідсистем навіть незмінна кількість камер може спричиняти значні коливання трафіку через зміни роздільності, частоти кадрів чи політик зберігання. Це підкреслює необхідність контекстної нормалізації: порівнювати навантаження слід не лише у абсолютних величинах, а й у перерахунку на активний вузол відповідного класу (трафік на пристрій, кількість сесій на пристрій, частка “важких” потоків у межах класу).

Рис. 2.7 Кількість активних пристроїв із 7-денним згладжуванням

Залишкова компонента після вилучення сезонності слугує індикатором “незвичної” поведінки. Короткочасні позитивні відхилення часто відповідають подієвим сплескам (масові оновлення прошивок, переналаштування з’єднань, короткі переривання зв’язку з подальшими ретрансляціями), тоді як негативні - технічним вікнам або деградації каналу. Важливо, що однотипні відхилення мають різну “нормальність” у різних підсистемах: для SCUD короткі піки у години пік є типовими, тоді як для сенсорних мереж вони радше нетипові. Саме тому у подальшому формуватимемо профілі “норми” окремо за класами пристроїв і часовими контекстами, а оцінка аномальності враховуватиме як абсолютну величину відхилення, так і відносну - щодо очікуваного рівня для відповідної підгрупи.

Додатково проаналізовано зміну відношення “трафік/активний пристрій” та інтенсивність сесій у часі. Для стабільних інтервалів це відношення має низьку дисперсію й виступає надійним “якорем” для швидкого моніторингу режиму; значне розширення розкиду свідчить або про внутрішню неоднорідність класу (поява нових моделей пристроїв із іншими характеристиками), або про зміну політик обміну даними. Такі ефекти важливо відокремлювати від справжніх

інцидентів: спершу перевіряти на предмет планових змін, а вже за їх відсутності розглядати як кандидата на аномалію.

Методологічно така комбінація декомпозиції та порівняння рядів “навантаження” і “активності” відповідає рекомендованим підходам до первинної діагностики аномалій у мережевих IoT-середовищах, де критичною є відокремленість тренду, сезонності та залишкових збурень.

Часова поведінка трафіку в IoT-сегменті характеризується стабільними добово-тижневими хвилями, які відрізняються між класами вузлів: сенсори формують майже регулярну телеметрію з невеликою дисперсією інтервалів, відеопідсистема породжує навантаження у робочі години, а системи контролю доступу мають подієвий характер із піками на початку та наприкінці змін. Для кількісного опису часові ряди агрегуються в погодинні та добові вікна з подальшою декомпозицією на тренд, сезонність і залишок; еталонні профілі будуються окремо для буднів і вихідних, а також для сезонів експлуатації. Оцінювання відхилень здійснюється не від глобального середнього, а від очікуваного рівня для конкретної години та типу дня; у такому поданні залишкові коливання більш наочно відбивають нетипову активність.

Підвищення стійкості до повільних змін режимів досягається експоненційним згладжуванням профілів із ковзним оновленням. Практична візуалізація у вигляді теплокарт “година ^{x} день тижня” та індексів циркадності (співвідношення денна/нічна активність) дає можливість швидко локалізувати смуги нетипової активності у класах пристроїв, не залучаючи вміст пакетів[19].

Протокольна структура визначається на основі агрегованих потоків із нормуванням часток трафіку за службами та транспортами. Класифікація базується на портово-протокольному підході з верифікацією на вибіркових PCAP і, для TLS, уточнюється через SNI/ALPN у початковому рукостисканні. Типовий розклад має дві домінанти: “легкі” службові та телеметричні взаємодії (DNS/NTP, MQTT/CoAP, короткі HTTPS-виклики) з великою кількістю коротких сесій і “важкі” мультимедійні канали (RTSP/RTP), де небагато тривалих з’єднань генерують левову частку байтів. Для формальної діагностики

формується нормований вектор протокольної підписки для кожного класу обладнання; мірою відхилення використовують, зокрема, дивергенцію Дженсена–Шеннона або χ^2-відстань від еталонного центроїда. Потоки з невизначеною семантикою агрегуються в категорію “other” з подальшим групуванням за сесійними ознаками та ручною атрибуцією найбільших кластерів. Агрегована методика узгоджується з моделлю потокової телеметрії IPFIX, що задає уніфікований спосіб експорту ознак потоків для подальшої атрибуції та аналізу [20].

Показники якості передавання оцінюються на транспортному та прикладному рівнях із мінімальною залежністю від інспектування вмісту. Для TCP відстежуються частка повторних передавань, поява дублікатних ACK і RST, успішність встановлення сесій; затримка наближено оцінюється за різницею між SYN і першим ACK з подальшою робастною агрегацією. Для UDP та відеопотоків ключовими є стабільність бітрейту і джитер, що оцінюються за розкидом секундних/хвилинних сум байтів та інтервалів між пакетами. Додатково беруться прикладні маркери: частка NXDOMAIN і медіанний час відповіді в DNS, успішність TLS-рукостискань, дисбаланс publish/subscribe і частота повторних під’єднань у MQTT. Щоб забезпечити порівнюваність, метрики нормуються відносно кількості активних пристроїв класу та очікуваного рівня для конкретної години; порогування виконується за медіанно-процентильними правилами або робастним z-score. Концептуально це відповідає підходам стандартизації показників якості в телеком-мережах, де базові метрики затримки, втрат і варіації затримки формалізовано в рекомендаціях щодо оцінювання продуктивності пакетних мереж [21].

Топологія взаємодій моделюється у вигляді графів “пристрій–сервіс” з фіксацією напрямів та ваг ребер за кількістю сесій або обсягом трафіку. Для виявлення центральних вузлів та змін ролей застосовуються показники ступеневої та посередницької центральності, а також виявлення спільнот. Вузли-концентратори, що несподівано збільшують “фан-аут” зв’язків або переносить центр ваги на зовнішні домени, позначаються як аномальні з погляду структури

взаємодій. Динамічні аспекти представляються як часові послідовності графів з порівнянням до еталонної добово-тижневої картини; стійкі перестановки контурів зв’язності виділяються окремо від короткочасних імпульсів. Методологічне підґрунтя оцінювання на графах спирається на усталені підходи аналізу мережевих систем, що довели ефективність у виявленні ключових ролей і структурних зрушень у складних мережах [22].

Профілі “норми”» формуються як сукупність еталонних шаблонів за часом, протоколами, якістю передавання та топологією. Для кожного класу обладнання визначається опорний набір дескрипторів: добово-тижневі хвилі з довірчими інтервалами, вектор протокольної підписки, робастні квантилі QoS-метрик і вектори центральностей у графі взаємодій. Порівняння поточного стану з опорними профілями здійснюється у координатах залишкових відхилень із контекстною нормалізацією за годиною та типом дня; така стратифікація суттєво знижує хибні спрацьовування в неоднорідному середовищі IoT і забезпечує інтерпретованість кожного сигналу відхилення.

2.3 Апробація методів детекції

Базова лінія детекції реалізована на моделі випадкової ізоляції (Isolation Forest), що працює з метаданими мережевого трафіку. Дані подано у вигляді погодинних спостережень для множини IoT-вузлів. Після попередньої обробки побудовано 10 ознак: ковзні агрегати (експоненційне середнє за 24 години, медіана вікна), сезонні залишки відносно еталонного рівня, інтерквартильний розмах, перша різниця, а також часовий контекст через гармоніки години доби та дня тижня (sin/cos). Сукупний обсяг становить 526 110 рядків; часовий розподіл - приблизно 70 % на тренування (368 280×10) та 30 % на тест (157 830×10), що унеможливлює витік у майбутнє. Масштабування здійснювалося робастним перетворенням (RobustScaler) для зменшення впливу важких хвостів[24].

```
    from sklearn.preprocessing import RobustScaler, MinMaxScaler
    from sklearn.ensemble import IsolationForest

scaler = RobustScaler()
Xtr = scaler.fit_transform(X_train)
Xte = scaler.transform(X_test)

IF = IsolationForest(
    n_estimators=500, max_samples=2048,
    contamination=0.02, random_state=0, n_jobs=-1, verbose=1
)
IF.fit(Xtr)
s_tr = -IF.score_samples(Xtr)
s_te = -IF.score_samples(Xte)
thr = float(np.quantile(s_tr, 0.99))
```

Рис. 2.8 Налаштування IF і вибір порога

Гістограма розподілу скорів для тренувальної й тестової вибірок із позначеним порогом 0.605 демонструє компактну основну масу значень та тонкий правий «хвіст», який перетинає лінію порога. Саме елементи цього «хвоста» утворюють множину кандидатів на аномалії для подальшої перевірки з протокольними та QoS-індикаторами. Комбінація квантильного порогування з робастним масштабуванням дає керовану інтенсивність тривог у середовищі без розмітки та легко переоцінюється після змін режимів роботи мережі.

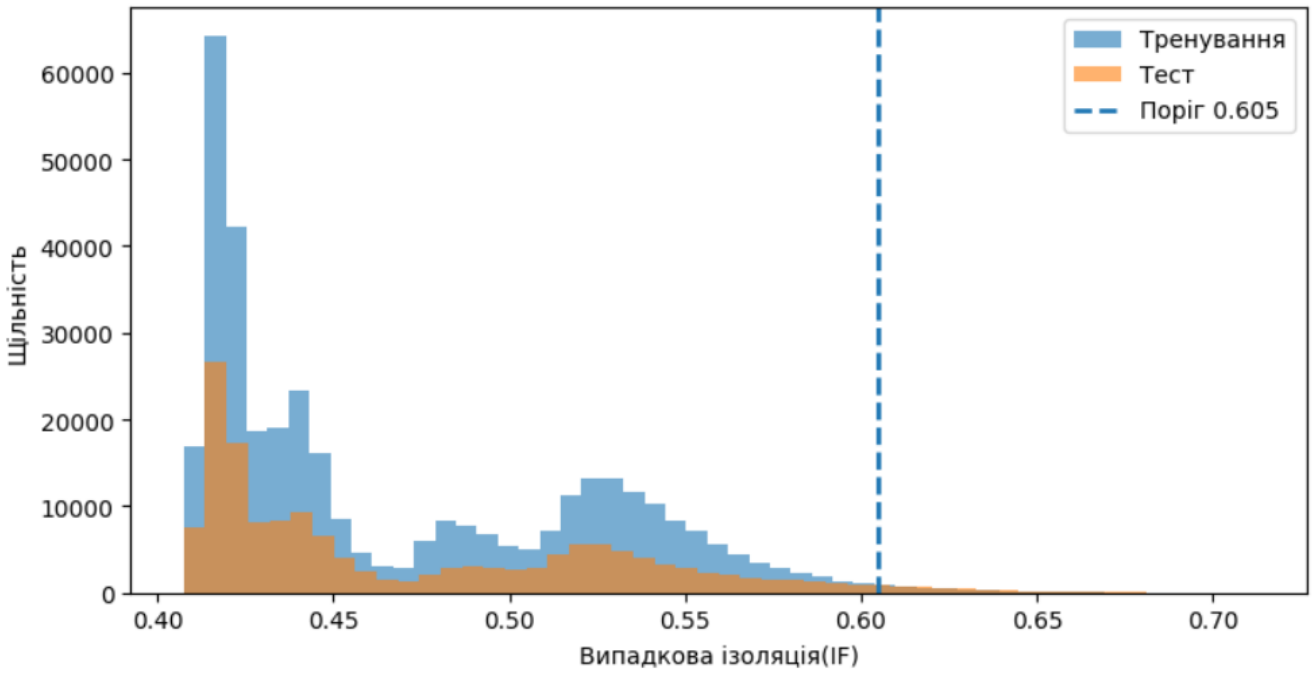


Рис. 2.9 Вибір порога за розподілом IF-скору

Базова ідея полягає у тому, що рішення “норма/аномалія” приймається, спираючись лише на метадані трафіку: агреговані обсяги в одиницю часу, залишки від очікуваних добово-тижневих профілів, частки протоколів та сервісів, робастні показники якості передавання (ретрансмітованість, успішність рукостискань, варіація міжпакетних інтервалів), а також прості топологічні дескриптори взаємодій. Такий підхід знімає залежність від інспектування вмісту пакетів, зменшує обчислювальні витрати на edge-вузлах і полегшує узгодження з корпоративними політиками приватності.

Для кожного пристрою формується коротке вікно ознак (24–48 годин із перекриттям), у якому часовий контекст (година доби, день тижня) забирається у “очікуваний рівень”, а моделі працюють із залишковою динамікою. Вектор ознак масштабується робастним скейлером окремо в межах класу обладнання, щоби не змішувати різнорідні режими (сенсори проти камер тощо). На цьому представленні застосовуються кілька безнаглядних детекторів із різною чутливістю: дерева ізоляції добре “відловлюють” рідкісні викиди у багатовимірному просторі; локально-щільнісні методи, подібні до LOF, реагують на скупчення нетипових точок довкола локально “незвичних” сусідів; методи межі підтримки (one-class) задають гладку оболонку “норми” і стабільно відсікають тривалі зсуви. Глибинні автоенкодери, навчені відтворювати короткі послідовності із “чистих” інтервалів, додають чутливості до колективних збоїв (наприклад, систематичне просідання бітрейту вночі поза типовими рамками).

Скори різних детекторів приводяться до спільної шкали й агрегуються робастно (медіана або перцентильне ранжування), що знижує залежність від окремої моделі та робить рішення стійкішим до дрібних змін у вхідних ознаках. Порогування задається квантилем на “еталонному” тренувальному інтервалі в межах класу пристроїв; додатково вводиться контекстна персистентність: спрацювання підтверджується, якщо поріг перевищено у кількох суміжних вікнах або за інтегральним індикатором. Така схема різко зменшує випадкові імпульсні тривоги, характерні для шумних мережевих процесів.

Інтерпретація події будується на розкладанні ансамблевого скору за групами ознак. Користувачеві подається коротке пояснення у вигляді “трійки» причин із найбільшим внеском: зсув сезонного залишку, перестановка протокольних часток (наприклад, аномальне зростання “other/tcp” замість очікуваного MQTT/HTTPS), погіршення QoS (стрибок ретрансмітованості, нестабільний бітрейт), нетипова топологічна роль вузла (раптове розширення “фан-аут” на зовнішні домени). Завдяки контекстній нормалізації пояснення завжди прив’язане до “звичного” для цієї години й типу дня режиму, а не до глобального середнього.

Операційна інтеграція передбачає дві лінії сповіщень. “Warning” використовується для м’якого порога та одиничних відхилень без явної персистентності; такі події накопичуються у дашборді для ретроспективного перегляду та кореляції. “Critical” вмикається при стійкому перевищенні жорсткого порога або при одночасному спрацьовуванні кількох незалежних груп ознак (наприклад, протокольний зсув разом із деградацією QoS), що є характерною ознакою інциденту. Для зменшення “шуму” схожі події агрегуються у кластери за близькістю у часі та подібністю пояснень.

Якість безнаглядної детекції відстежується непрямими показниками. Перше - стабільність частки тривог у “тихі” періоди та її зростання в моменти, коли зовнішні системи вже фіксують зміни (оновлення прошивок, ремонтні роботи, перебудова сегментації). Друге - середній “час до підтвердження” інциденту після спрацювання критичного правила. Третє - частка повторних спрацювань за тією самою причиною на тому самому вузлі, що сигналізує про системну проблему, а не про разовий сплеск. За потреби пороги коригуються автоматично за контрольними квантилями у ковзному вікні з виключенням явних аномальних днів із бази “норми”.

Узгодження з ресурсними обмеженнями робиться двоступеневим. Легкі агрегати й прості детектори виконуються на периферії (edge), що дозволяє миттєво отримувати попередні скор-сигнали та локальні “warning”. Важчі обчислення - batch-перерахунок ансамблю, навчання автоенкодера, оновлення

еталонних профілів - виконуються у хмарі за розкладом, після чого нові параметри поширюються на периферію. Така організація не потребує DPI, масштабується до великої кількості вузлів і зберігає інтерпретованість результатів для інженерної експлуатації.

Система виявлення має працювати з метаданими мережевого трафіку - потоками, агрегатами, базовими показниками якості передавання та простими топологічними дескрипторами - без аналізу вмісту пакетів. Базовою організаційною ідеєю є двоступенева обробка: легкі перетворення на периферії для швидкого попереднього сигналу та централізований перерахунок і калібрування в хмарі. Такий підхід дає керовані вимоги до ресурсів, спрощує масштабування та полегшує дотримання політик приватності.

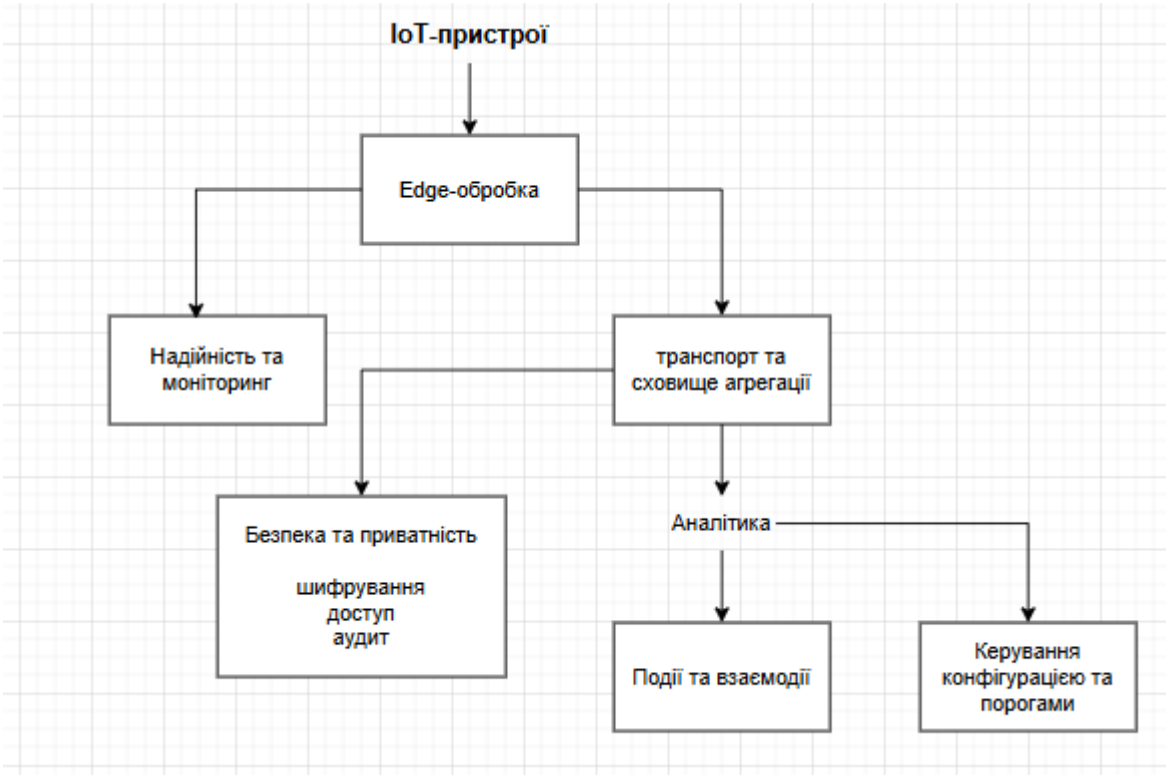


Рис. 2.10 Узагальнена архітектура та потік обробки.

Вхідні дані надходять потоково з пристроїв і мережевих елементів у уніфікованому поданні з нормалізованими часовими мітками. Потоки синхронізуються, агрегуються у погодинні або хвилинні вікна, після чого з них обчислюються ознаки: ковзні статистики, сезонні залишки, частки протоколів і

робастні QoS-індикатори. Для різнорідного парку обладнання ознаки масштабуються окремо в межах класів на кшталт сенсорів, камер, систем контролю доступу чи актуаторів - це запобігає ”змішуванню” режимів. На периферії формується попередній скор, у центральному середовищі - ансамбль моделей із порогуванням та періодичним донавчанням.

Вимірювана якість даних критично важлива: контролюється повнота записів і стабільність частоти, перевіряється монотонність лічильників, обробляються прогалини без попадання у профіль “норми”. Чутливість і навантаження операторів регулюються порогами, що калібруються на “чистих” відрізках часу окремо для кожного класу пристроїв; рішення супроводжується коротким контекстом - ідентифікатор вузла, часовий інтервал, ключові зсуви у групах ознак.

Продуктивність проєктується так, щоб затримка на edge-вузлах лишалася близькою до реального часу, а важчі процедури - навчання, оновлення еталонних профілів, узгодження порогів - виконувалися пакетно в центрі без впливу на онлайн-телеметрію. За зростання кількості вузлів і потоків конвеєр не повинен створювати вузьких місць ні на кроці агрегації, ні на кроці генерації подій.

Надійність забезпечується ідемпотентним процесингом та буферизацією на випадок тимчасових збоїв доставки. Сервіси самі відстежують “здоров’я” конвеєрів і переходять у деградований режим, якщо джерело починає “хитатися” (падає частота спостережень, зміщується час). Технічні аномалії телеметрії відокремлюються від контентних інцидентів, аби не плутати причини реагування.

Система має пристосовуватися до дрейфу режимів. Профілі “норми” та пороги оновлюються за ковзним вікном, а періоди масових змін - оновлення прошивок, перебудова сегментації - не входять до бази калібрування. Версії моделей і конфігурацій зберігаються, передбачено контроль змін і швидкий відкат. Оператори можуть вносити зауваження до інцидентів; цей зворотний

зв’язок використовують для точнішого налаштування порогів і правил персистентності спрацювань.

Пояснюваність - обов’язкова властивість. Для кожного сигналу подається зрозуміле резюме внеску груп ознак: чи це зсув сезонного залишку, нестандартна перестановка протокольних часток, погіршення QoS або нетипова мережева роль вузла. Дашборди показують динаміку скорів і порогів, частку тривог у “тихі” періоди, карти протоколів і еталонні профілі. Близькі за змістом події агрегуються у сюжети, а правила оповіщення підтримують м’який і критичний рівні, щоб зменшити шум.

Інтеграція відбувається через стабільні API для подачі телеметрії, читання агрегатів і подій та керування конфігурацією. Підтримуються стандартні формати експорту й підключення до SIEM. Канали та сховища шифруються, доступ надається за принципом найменших привілеїв із ролями та аудитом, персональні ідентифікатори мінімізуються або псевдонімізуються. Механізми життєвого циклу моделей включають регулярні звіти якості, ін’єкції контрольних відхилень для перевірки чутливості та повну відтворюваність експериментів.

Узгоджені таким чином вимоги відповідають обмеженням IoT-середовища, забезпечують стійку роботу на великій кількості вузлів і дозволяють масштабувати виявлення відхилень без розкриття вмісту трафіку.

3 КОНСТРУКТИВНІ РІШЕННЯ ТА ПРОЄКТ ВПРОВАДЖЕННЯ

3.1 Архітектура та конвеєр даних

Цільова архітектура зосереджена на роботі з метаданими трафіку та зменшенню навантаження на канали та вузли периферії. Початковий збір даних здійснюю шлюзи(gateway) в межах локальних сементів. Вони приймають телеметрію від IoT-пристроїв, нормалізують часові мітки, здійснюють базову фільтрацію службових потоків та одразу групують події у стислі інтервали.

Завдяки впровадженню легкої аналітики на рівні шлюзів, наприклад, застосування простих порогових правил, що дає дає змогу швидко реагувати на аномальну активність без пересилання повного трафіку до центру. Це знижує чутливість системи до затримок, підтримує дотримання базових рекомендацій щодо безпечного налаштування конфігурації пристроїв і шлюзів(паролі, новлення ПЗ, зменшення відкритих шлюзів) та полегшує сегментацію мережі, які є базовими вимогами для споживчих IoT-екосистем[25].

Рис. 3.1 Архітектура рішення

На наступному етапі дані надходять до транспортного рівня, який реалізовано через черги повідомлень, де вони розмежовуються на продуктивний та аналітичний контури. Підсистема зберігання даних(конвеєр) виділяє два типи сховища. Першим є часові ряди, які використовуються для оперативних агрегованих показників. Другий тип сховища це об’єктне сховище, воно слугує для “повних” фрагментів даних(сніпети) та довгострокових архівів.

Обмін організовується як система подій із чіткими контрактами на схеми даних та частоти їх надходження, що дає змогу безпечно масштабувати споживачів і повторно відтворювати потоки у разі відмов. Уздовж усього маршруту передавання додаються маркери якості, туди входять і частка втрачених записів, і різкі зміни значень лічильника, і зрушення часових міток. Вони фіксуються окремо від змістових інцидентів, щоб технічні проблеми телеметрії не плутались з реальними відхиленнями в поведінці вузлів.

Формування ознак виконується в окремому конвеєрі побудови фіч. На його вхід надходять агреговані метрики(байти/пакети за інтервалом, частки протоколів/портів, базові топологічні характеристики взаємодії, показники якості передавання). Далі до цих метрик послідовно застосовуються стабільні трансформації(робастні відхилення від добових чи тижневих профілів, компоненті , які пов’язані з часом доби).

Через різнорідність обладнання, ознаки нормалізують окремо в межах класів пристроїв, щоб уникнути змішування режимів камер, систем контролів доступу, сенсорів та актуаторів. Керованість цього рівня забезпечується підходами, характерними для промислових ML-конвеєрів, описами схем, автоматичними перевірками сумісності, контролем статистики та версіонуванням як перетворень, так і сформованих наборів ознак. Така дисципліна дозволяє безпечно оновлювати моделі та перевидавати фічі без порушення робочих процесів[17]. Фрагмент коду наведено на рис. 3.2

```
import pandas as pd, numpy as np
from sklearn.preprocessing import RobustScaler
from sklearn.ensemble import IsolationForest
import matplotlib.pyplot as plt

X = pd.read_csv("artifacts/features_hourly.csv",
    usecols=["bytes","ewm24","resid","roll_med","iqr","diff1","hour_sin","hour_cos","dow_sin","dow_cos"]
).astype("float32").values

X = RobustScaler().fit_transform(X)
IF = IsolationForest(n_estimators=300, max_samples=4096, contamination=0.02, random_state=0).fit(X)
scores = -IF.score_samples(X); thr = np.quantile(scores, 0.99)

plt.hist(scores, bins=50); plt.axvline(thr, ls="--"); plt.xlabel("Скор аномальності (IF)"); plt.tight_layout()
```

Рис. 3.2 Фрагмент навчання Isolation Forest та вибір порога

Над шаром ознак розташовується детекторний контур. Його мінімальна конфігурація включає швидку базову лінію для фільтрації сезонних та трендових коливань, та один некотрольований детектор багатовимірних метаданих. Рішення зводиться до швидкого виявлення аномалій та винесення бінарного вердикту при контрольованому пороговому значенні. Результат подається разом з коротким поясненням які групи ознак дали найбільший внесок у рішення, також у якому контексті це було зроблено(клас пристрою, час, день чи ніч). Окремий процес відслідковує стабільність порогів та профілів “норми” і виконуємягке оновлення у ковзному вікні, щоб уповільнити дрейф, і він не призводив до великої кількості хибних спрацювань.

Результати роботи детекторів надходять до підсистеми керування подіями. Окремі спрацювання групуються в інциденти за часом та схожістю, після призначається приорітет, а вже далі вони відображаються на панелі моніторингу разом із динамікою скорів і порогових значень. Для оператора є два основні режими взаємодії. Перший це режим м’яких попереджень для подальшого аналізу подій, що відбулися та другий канал – канал критичних тривог, що вимагає негайної реакції. Залишений оператором зворотній зв’язок для надзвичайних ситуацій з спрацюваннями зберігається й використовується для тонкого налаштування порогів та інцидентів, за наявності маркувань, також для напівкерованого покращення моделей.

Питання захисту приватності вирішується на всьому шляху даних. Спочатку індентифікатори вузлів змінюють назву зі збереженням сталих відповідностей у межах періоду аналізу, після цього канали зв’язку захищають шифруванням та після цього доступ до журналів подій та конфігурації здійснюється за ролями з мінімальними привілеями та обов’язковими логами аудиту.

На периферійному рівні застосовується максимально обмежена політика відкриття інтерфейсів та запроваджується обов’язкові оновлення програм. Тоді як у центральній частині діє інший життєвий цикл, який схвалює, випробовує та поетапно розгортає. Саме таке поєднання вимог до пристроїв, процесів та шлюзів узгоджується з актуальними галузевими рекомендаціями для споживчих IoT-систем і водночас не заважає можливостям масштабування рішень[25].

Завершальною складовою є забезпечення експлуатаційної спостережуваності самого конвеєра. Відстежуються затримки на ключових стадіях (приймання даних, формування ознак, агрегування, детекція), частка втрачених або ж повторювальних записів, наватнаження CPU/RAM на вузлах периферії та у центральних сервісах. Усі ці артефакти починаючи від описів схем до наборі ознак і порогових налаштувань – версіонізуються, що в свою чергу дозволяє відтворювати експерименти, порівнювати результати резних релізів та безпечно виконувати відкати змін. Такі підходи до керування даними й моделями є необхідною передумовою стабільної роботи конвеєра у промисловій експлуатації та підтримується інструментами класу виробництва ML-платформ [17].

3.2 Методика виявлення та адаптації

Перед оцінюванням відхилень, необхідно відокремити регулярну складову трафіку від випадкових коливань. Для цього погодинні агрегати байтів по вузлам приводять до єдиної шкали часу, синхронізуються за часовими поясами, усувають дублікати та артефакти лічильників, такі як раптові “провали”

через переповнення. А прогалини заповнюються короткою спеціальною встакою (інтерполяцією) з подальшою перевіркою, що відновлення не спотворює локальну динаміку. На цьому етапі окремо фіксується технічні показники якості даних(частота перепусток, тимчасові зрушення та різкі зрушення), щоб не плутати збої телеметрії з поведінкою аномалій.

Стабілізацію виконують через явне моделювання нормальної поведінки, як поєднання повільного змінювання тренду та періодичних сезонних коливань. У задачах IoT зазвичай домінують добові та тижневі цикли, тому для кожного вузла або класу пристроїв будується робастна сезонно-трендова декомпозиція, яка згладжує повільні зміни навантаження. Прикладом такого можна назвати поступове зростання потоку від камер після оновлення ПО та відображає типовий профіль активності за годинами доби та днями тижня. Візуальний приклад такого розкладу навів на рис. 3.3, що дає змогу перевірити коректність відокремлення регулярної компоненти від коливань.

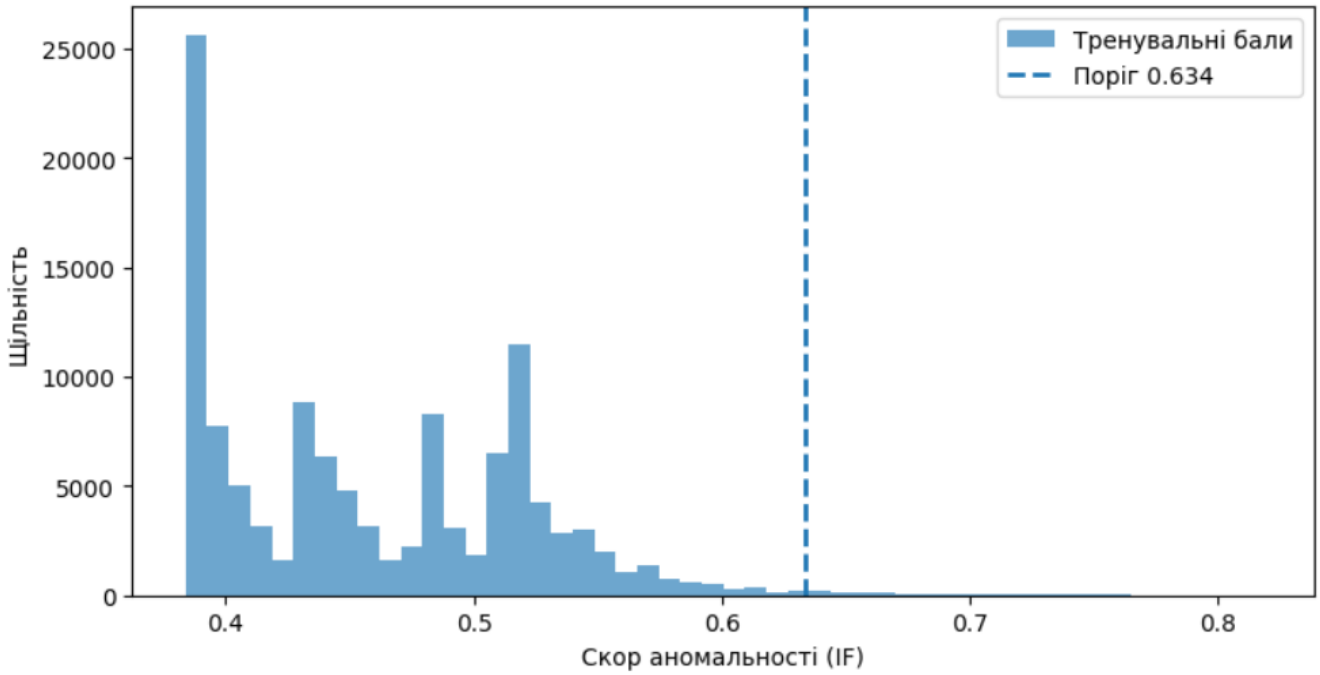


Рис. 3.3 Результати Isolation Forest із порогом

Щоб модель “норми” не реагувала на поодиночні різкі сплески, компоненти оцінюють за допомогою стійких до викидів методів. Можна виділити три методи: ковзний медіан, квартильних інтервалів (IQR) для

вимірювання варацій та експоненційно згладжених середніх з довгим вікном. Добові та тижневі цикли описують через синусоїдальні базисні функції або еквівалентні сезонні профілі, які навчають на інтервалі з плином часу із мінімальною кількістю інцидентів. Для різних типів обладнання ця процедура виконується окремо в межах групи. Групами є камери, системи контролю доступу та сенсори, що не дає розмити характерні профілі через відмінності у масштабах та режимів роботи.

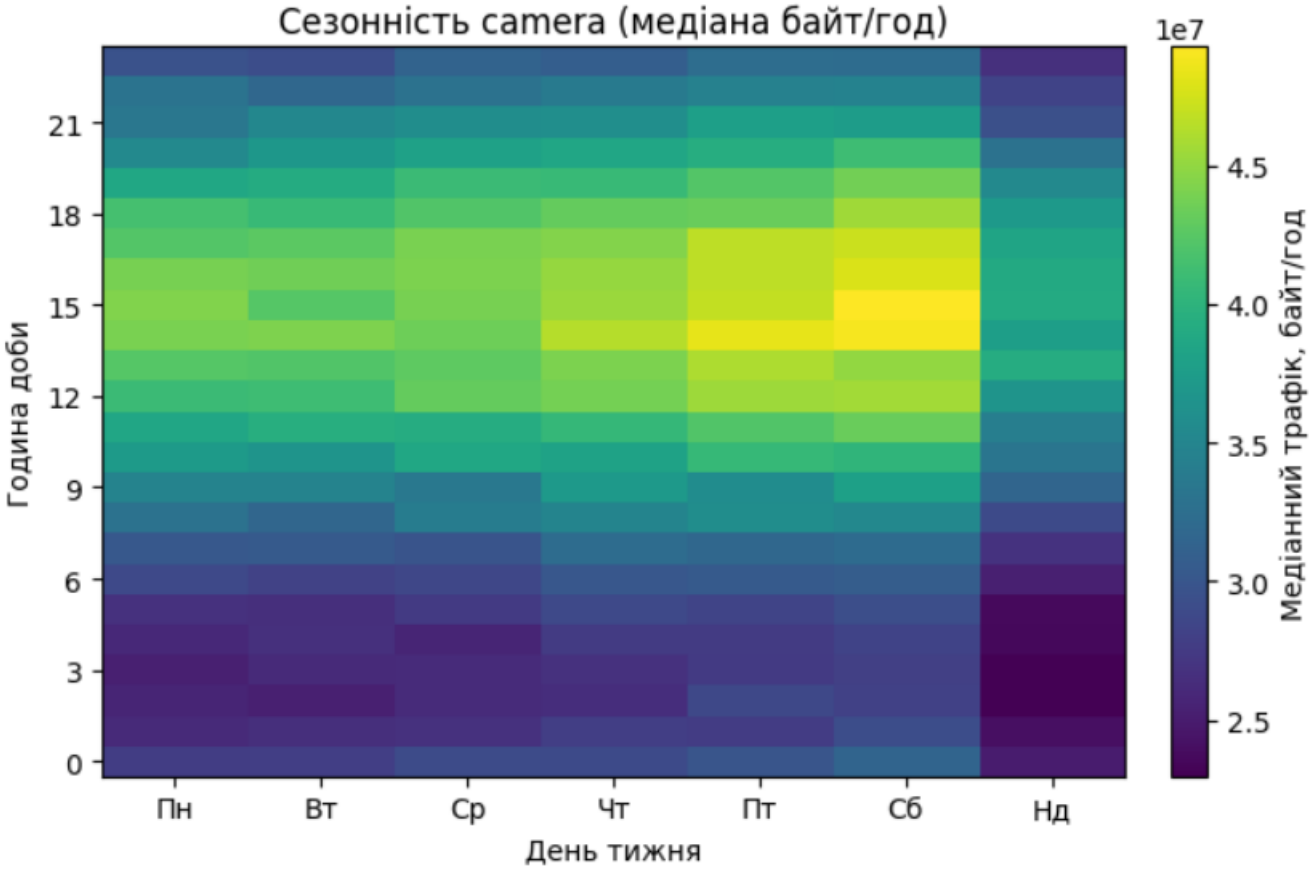


Рис. 3.4 Теплова карта медіанного трафіку

У підсумку сезонно-трендової декомпозиції формується залишкова компонента з різницею між фактичним значенням та передбаченою нормою. Саме цей залишок виступає основним носієм інформації про нетипові відхилення, оскільки вже очищений від регулярної циклічності та повільних змін зони. Щоб підвищити стійкість подальшого аналізу, залишок додатково нормалізують у межах відповідного класу пристрою. Прикладом є використання робастного масштабування відносно медіани та IQR. Після цього роширюють

опис локальною динамікою і часовим контекстом. Такий набір ознак допомагає моделі краще відрізняти очікувані пікові навантаження від справді аномальних сплесків.

Особливу роль відіграє правильний вибір інтервалу для калібрування. Якщо історія містить відомі періоди з інцидентами, моделі “норми” навчають на відносно спокійних ділянках, а перевірку виконують на окремих відрізках із типовою сезонністю. За відсутності маркувань застосовується самокалібрування з відкриттям хвостів розподілу залишку та повторним навчанням до того моменту, поки статистики не стабілізуються. Такий підхід дозволяє сформувати надійні та стабільні профілі навіть у системах зі змінними навантаженнями.

Параметри сформованих профілів це довжини вікон, коефіцієнти згладжування, амплітуди сезонних компонент. Вони версіонуються та зберігають разом із метаданими якості даних, що критично для відтворення результатів та контрольованого оновлення. У режимі промислової експлуатації профілі “норми” оновлюються в рамках ковзного вікна. Нові спостереження поступово додаються до оцінки тренда та сезонності, але з обмеженою швидкістю, щоб короткочасні збурення не призводили до їх деградації. Завдяки цьому система м’яко підлаштовується до довготривалих змін і водночас зберігає чутливість до справжніх аномалій.

Мета побудови простору ознак полягає в тому, щоб перетворити сирі погодинні вимірювання трафіку на такі представлення, які не ламаються через періодичність та різний масштаб навантаження між окремими пристроями, а аномальність у них чітко відокремлена від звичайних коливань. Вихідною точкою слугує залишок після вилучення трендової та зонної складової, адже саме він містить інформацію про нетипову поведінку. Різниця поточним значенням та фоновою оцінкою утворює залишок. Локальні зміни кодуються першою різницею, що робить модель чутливою до різних зсувів за декілька годин без прив’язки до рівня трафіку.

Часовий контекст задають у циклічному вигляді, щоб модель коректно сприймала та могла розрізнити робочі дні та вихідні. Для цього годину доби та

день тижня відображають у просторі пар синусів на косинусів, що прибирає штучні розриви на межах періоду та дає змогу навчитися в безперервному кутовому просторі. Такі ознаки не піднімають сезонну нормалізацію, а працюють разом із нею. Вона однакова за величиною амплітуди залишку в спокійні години та в пікові години навантаження, вони мають різну інформативність. Саме часові синусоїди дозволяють моделі коректно врахувати різницю.

Щоб зробити вектори порівнюваними між пристроями різних класів, використовують робастне масштабування в межах класу. Кожна ознака нормується за медіаною та IQR, як аналог RobustScaler. Вона зменшує вплив крайніх значень і забезпечує стабільну шкалу навіть на наявності поодиноких викидів[27]. Нормування здійснюється за параметрами, обчисленими на чистому відрізку даних та надалі в потоці використовують саме ці фіксовані оцінки. Такий підхід унеможливлює витік інформації та забезпечує відтворення результатів. Для неоднорідної кількості пристроїв процедуру нормування виконує окремо в межах груп (камери, системи контролю доступу, актуатори та сенсори), щоб не спотворювати масштаб через змішування різних режимів роботи.

У результаті базовий вектор ознак на одну часову точку містить небагато компонентів - рівень трафіку та його робастну оцінку фону, залишок і його варіативність у вікні, першу різницю, а також пару синус/косинус для години та дня. Такий компактний набір добре масштабується, не потребує значних обчислювальних ресурсів, придатний для потокової обробки та забезпечує чутливість до відхилень, що тривають від однієї години до кількох діб. Саме на цих ознаках надалі навчається детектор, а обрані пороги інтерпретуються в узгодженій, стійкій шкалі.

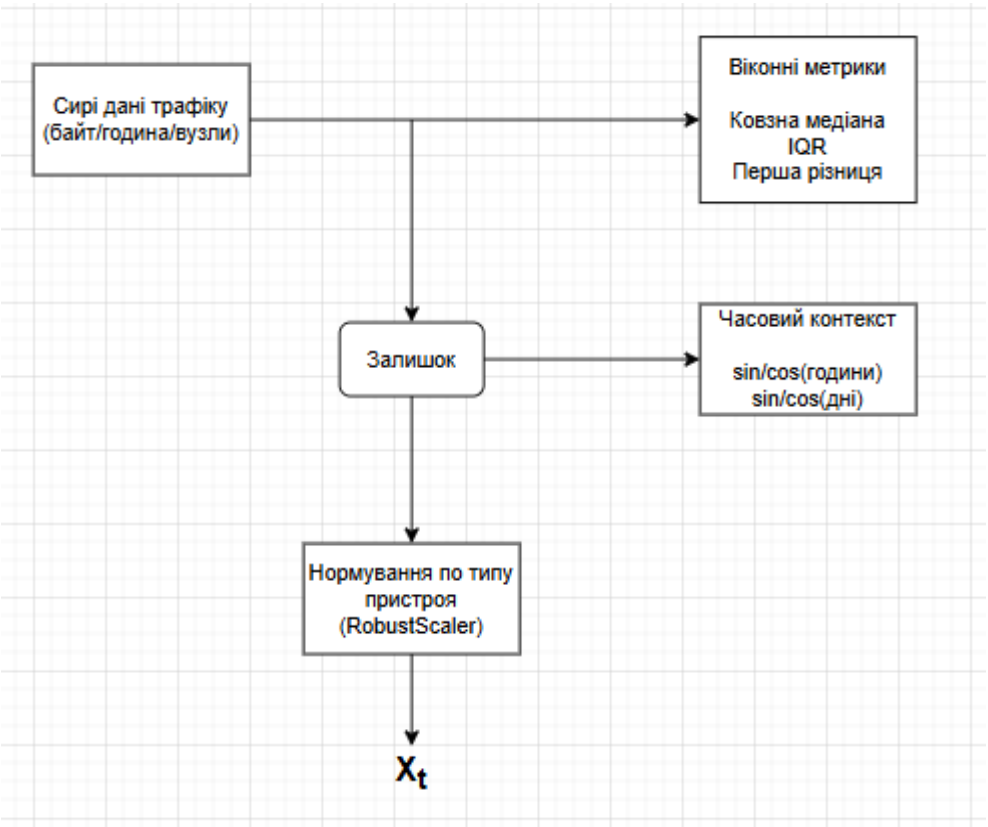


Рис. 3.5 Структура простору ознак

Потік “сирих” спрацювань від детектора перетворюється на осмислені інциденти поетапно. Спершу значення аномальності згладжують коротким експонтенційним середнім, аби відсіяти випадкові імпульси. Відкриття події фіксують не за одиничним перевищенням порога, а за правилом накопичення у ковзному вікні з M годин подія стартує, якщо принаймні K точок мають скор не нижчий за робочий поріг. Закриття налаштовують з гістерезисом - інцидент вважається завершеним лише після стабільного повернення скорів нижче зниженої межі, що усуває ”миготіння” поблизу порога та скорочує кількість зайвих сповіщень.

Далі застосовується часово-просторова агрегація. Спрацювання того самого вузла, розділені невеликим інтервалом спокою, об’єднують у один інцидент із сумарною тривалістю; паралельні події на різних вузлах з однаковим профілем ознак (стрибок залишку у нетипову годину) групують на рівні підмережі, VLAN або класу пристроїв. Для середовищ із чіткою топологією додатково враховують близькість за L2/L3-сегментами та спільні шлюзи

доступу, що дозволяє відрізнити локальну відмову датчика від мережевого збою, який одночасно вражає цілу групу елементів.

Кожному інциденту нараховується інтегральна вагомість, яку зручно визначати як комбінацію пікового скору, тривалості, охоплення та контексту критичності обладнання. Практична форма індексу важливості може мати вигляд:

$$Severity = w\_1 \cdot max\ s\_t + w\_2 \cdot log(1 + duration) + w\_3 \cdot share\_affected + w\_4 \cdot criticality$$

де:

- max s_t — інтенсивність відхилення за час інциденту (з нормованим скором моделі).
- log(1+duration) — тривалість із насиченням (щоб 12 год не були рівні)
- share_affected — частка уражених вузлів у класі/сегменті (масштаб впливу).
- criticality — ваговий коефіцієнт важливості активу.

Щоби зменшити “шум” повідомлень, застосовують пригнічення дублювань і керування частотою сповіщень. Інциденти однакового типу, що повторилися в межах короткого вікна, не створюють нових алертів, а лише оновлюють стан існуючого запису. Для трафіку задають максимальну швидкість генерації подій на клас пристроїв; перевищення ліміту переводить нові спрацювання у безпечний буфер з агрегованим звітом наприкінці інтервалу.

Операційну корисність підсилює коротка діагностика причин відхилення. Разом із кожним інцидентом зберігаються ключові ознаки, що зробили найбільший внесок у рішення (підвищений залишок у тиху годину, різкий ріст IQR, розрив добового патерну), а також стислий текстовий опис. Це скорочує час аналізу черговою зміною і підвищує відтворюваність рішень. Різні оператори бачать однакові пояснення та діють за єдиним регламентом.

Післяобробка завершується нормалізацією життєвого циклу інциденту. Відкриття, підтвердження, ескалація, гасіння та закриття відмічаються явно; усі стани версіонуються з часовими мітками, а артефакти (витяги з логів, знімки графіків, конфігураційні зміни) прикріплюються до картки інциденту[28]. Події синхронізуються з системою заявок, що забезпечує трасованість від спрацювання до виконаних дій і дозволяє збирати розмітку “істинних/хибних” спрацювань для подальшого тонкого налаштування порогів і перенавчання моделей.

Нарешті, для стабільності в довгій перспективі вбудовується контроль дрейфу потоку сповіщень. Відстежуються частота інцидентів, середня тривалість, частка помилкових спрацювань, розподіли скорів і пауз між подіями. Виявлений зсув це сигнал до перегляду параметрів агрегації, порогів або самої моделі. Завдяки такому конвеєру “спрацювання → інцидент” система утримує керовану кількість оповіщень, забезпечує операторам стислий і пояснюваний контекст і дає відтворювані підстави для прийняття рішень у реальному часі.

Щоб сповіщення були не лише своєчасними, а й придатними до дії, кожне виявлене відхилення супроводжується коротким поясненням “чому саме спрацювало”. Логіка проста - операторові важливо побачити не абстрактний бал аномальності, а конкретні ознаки, які «потягнули» подію вгору. Для цього аномальний рахунок розкладається на локальні внески ознак у точці часу t — залишок після вилучення тренда та сезонностей, ковзна медіана, IQR, перша різниця й циклічні компоненти години та дня тижня. В результаті формується компактна картка інциденту це найбільший внесок дав різкий стрибок залишку у нетипову годину; вторинний вплив - розширення IQR протягом останнього вікна.

Практично локальну пояснюваність обчислюють двома взаємодоповнюваними шляхами. Перший - безпосередньо для моделі Isolation Forest через методи розкладання внесків на деревних ансамблях або через сурогатну регресію, а далі навчається легка пояснювана модель, що наближує залежність у локальному околі точки; ваги такої моделі читаються як важливості

ознак у конкретному випадку. Другий шлях - перестановка зниження якості: для кожної ознаки створюють локальні ”випадання” й вимірюють, як змінюється рахунок; чим помітніший приріст або спад, тим вагоміша ознака для рішення в цій точці. Обидва підходи не потребують розмітки атаки/норми і працюють на метаданих трафіку, що зручно в умовах широкого шифрування прикладного рівня.

Пояснення подаються у двох формах - числовій та візуальній. Числова форма - це вектор із кількома найбільшими за модулем внесками, нормованими до зрозумілої шкали, разом із короткою текстовою анотацією. Візуальна форма це невелика горизонтальна діаграма внесків деяких ознак, де позитивні значення штовхають рішення до «аномалії», а негативні - у бік норми. Такий міні-звіт одразу відповідає на три запитання оператора: що саме зламалося у звичному патерні, чому система вважає це суттєвим і в якому контексті часу це сталося.

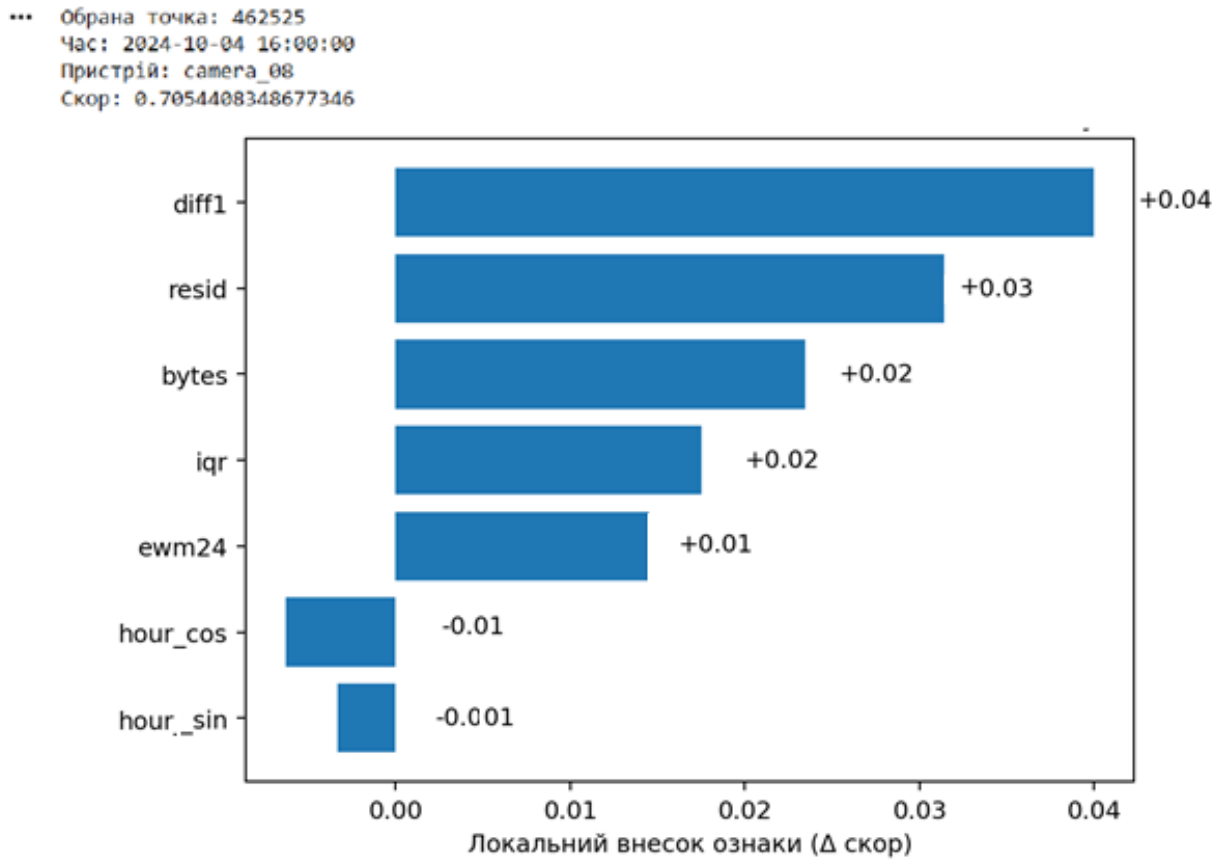


Рис. 3.6 Локальна важливість ознак для інциденту

Щоб пояснення залишалися корисними у великому парку пристроїв, їх обмежують ”бюджетом”: не більше 3–5 найсильніших факторів на інцидент і не більше кількох рядків тексту, з пріотету доменно значущих ознак. Додатково впроваджується валідація пояснюваності для вибірки інцидентів перевіряють узгодженість причин із сирими рядами та залишками (контрприклади відкидаються), оцінюють стабільність ранжування ознак при малих збуреннях даних і зберігають приклади в репозиторії кейсів для навчання нових операторів.

У виробничому режимі пояснення обчислюються там, де це доцільно з погляду ресурсів: на краю (edge) зберігається тільки топ-k внесків і короткий текст, а повні розкладання та графіки формуються у хмарі для інцидентів із високою важливістю або на вимогу. Це дозволяє не перевантажувати кінцеві вузли, але мати глибший розбір тоді, коли він справді потрібен. Для підвищення відтворюваності всі версії моделей і правил формування пояснень версіонуються, а прикріплені до інцидентів артефакти - діаграми внесків і фрагменти рядів - зберігаються в єдиному журналі подій.

Потоки IoT-трафіку змінюються з часом, змінюються режими роботи обладнання, з’являються нові прошивки, переїжджають сервіси, і навіть типові добово-тижневі патерни поступово дрейфують. Щоб детектор не старів і не втрачав чутливість, у конвеєр вбудовується керована адаптація. У практичних термінах розрізняють дрейф ознак (covariate drift, коли змінюється розподіл вхідних фіч), дрейф класів (prior drift, коли змінюється частка аномалій у потоці подій) і концептуальний дрейф (concept drift, коли змінюється сама залежність між ознаками та рішенням). Для кожного типу застосовують мінімально достатній рівень втручання: від простої перекалібровки порога до часткового або повного перенавчання моделі.

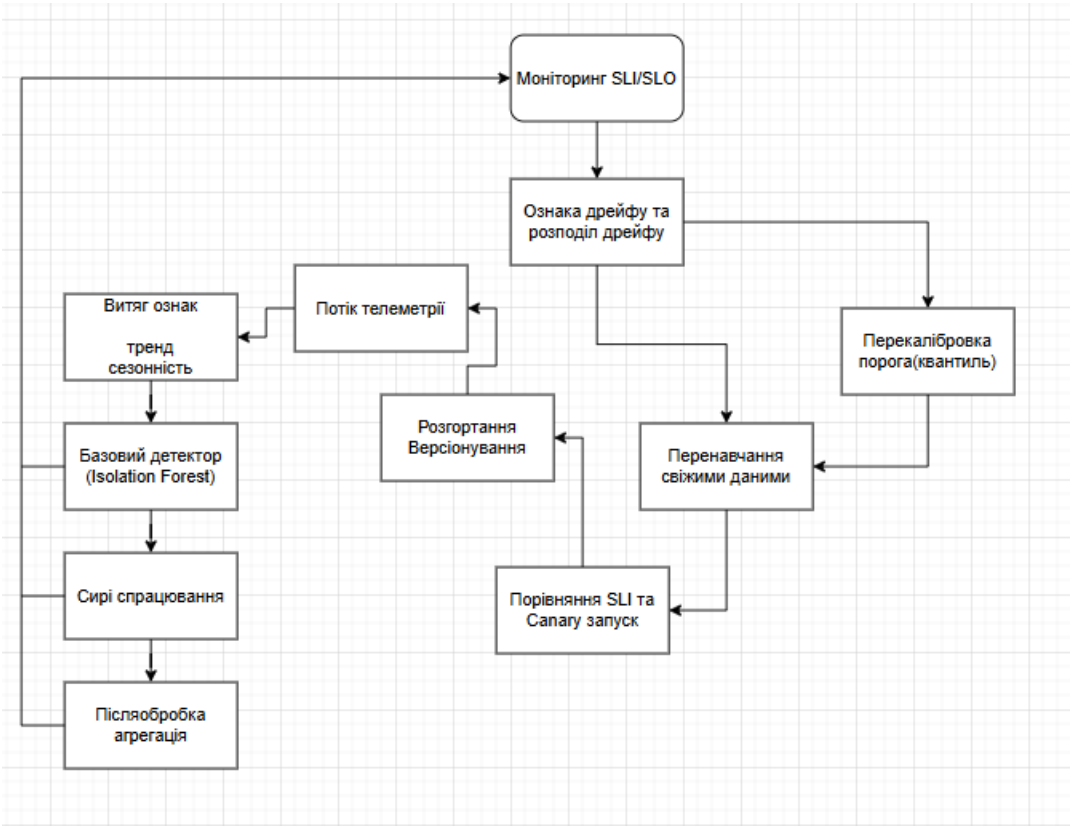


Рис. 3.7 Контур адаптації до дрейфу

Базова лінія витримується за рахунок ковзної перекалібровки порога на основі свіжого «чистого» вікна спостережень. Розподіл скорів на цьому вікні порівнюють з еталонним; за стабільності відхилень поріг оновлюють як заданий квантиль (наприклад, 0.99), зберігаючи бюджет хибних спрацювань. Якщо показники стабільності погіршуються, вмикається глибша адаптація: обмежене перевчання на останніх даних зі ”захистом від інцидентів” (вікна з масовими відхиленнями виключаються), а при виявленні концептуального дрейфу - навчання кандидата-моделі у режимі shadow або canary й лише потім - контрольоване переключення.

Ознаки дрейфу контролюються кількісно. Для фіч застосовують індекси стабільності на кшталт PSI (Population Stability Index) або дивергенції між еталонним і поточним вікном; для моделі - моніторять розподіли скорів, частоту перетинів порога та часові інтервали між спрацюваннями. Відхилення вище попереджувальних порогів запускають автодіагностику, перевіряється якість

даних, уважно розглядаються зміни у Flow побудови ознак (оновлення прошивок, нові типи пристроїв, реконфігурації мережі). Адаптація завжди версіонується, параметри профілів норми, пороги, дати ввімкнення нових моделей та їхні контрольні звіти зберігаються для відтворюваності.

Рис. 3.8 Моніторинг дрейфу (PSI) з порогами

Експлуатаційна придатність рішення фіксується через узгоджені індикатори рівня сервісу (SLI) та цільові значення (SLO)[29]. До якості детекції відносять частоту хибних тривог у контрольні періоди, частку підтверджених інцидентів за відгуками операторів, середню затримку виявлення від моменту фактичної події до спрацювання, а також стабільність порога в часі. Для самого конвеєра важать and-to-and затримка обробки від телеметрії до сповіщення, доступність сервісу за годину/добу/місяць, свіжість даних у буфері та споживання ресурсів на edge і в хмарі. Виробнича цінність додатково вимірюється індексом важливості інцидентів, частка сповіщень рівня Critical і Major, середній час до ескалації та закриття, навантаження на чергу опрацювання.

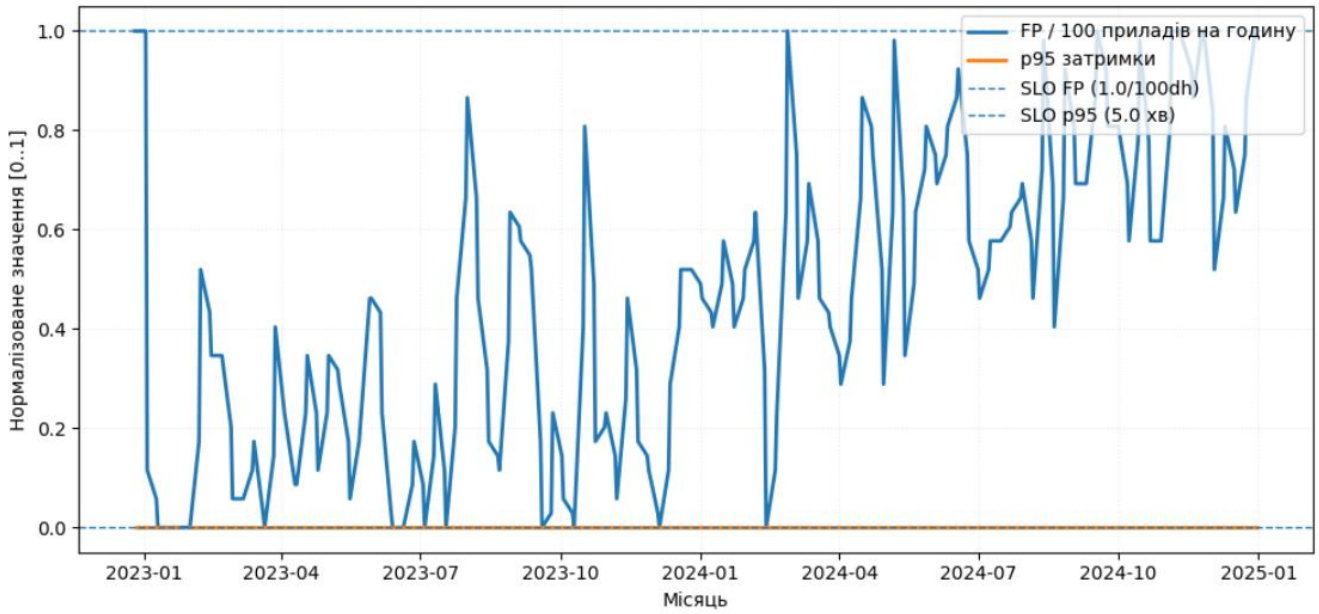


Рис. 3.9 Тенденції якості: частота хибних спрацювань та p95-затримка

Цільові пороги задаються під доменні обмеження: наприклад, не більше двох хибних тривог на 100 пристроїв за годину; медіана затримки детекції - до кількох хвилин; and-to-and затримка конвеєра - не більше за 95-й перцентиль у межах погодженої секундо-хвилинної норми; доступність сервісу 99.9% на місяць; частка підтверджених інцидентів - не нижче узгодженого мінімуму. Для дрейфу встановлюють жовті та червоні пороги для PSI/дивергенцій, а також ліміти на добову зміну робочого порога, щоб модель не плавала надто агресивно. Порушення SLO не просто логуються - вони зменшують ”error budget” сервісу і ініціюють план дій, від фіксації дефектів у флоу ознак до перегляду розкладу перенавчання.

Перевипуск моделі або порога відбувається за правилами безпечного розгортання. Кандидат запускається у shadow-режимі паралельно з чинною версією; збираються SLI-метрики та порівнюються з референсними; допускається короткий canary на частці вузлів або сегментів мережі; автоматичне перемикання дозволяється лише за умови, що різниця за ключовими SLI належить до коридору прийнятності. Якщо кандидат погіршує хоча б один критичний індикатор, він відкочується без впливу на продуктивний контур. Усі

рішення супроводжуються коротким звітом, щоб забезпечити трасованість і накопичувати знання для наступних ітерацій.

Зворотний зв'язок операторів інтегрується в цикл покращення. Підтвердження або відхилення інцидентів, причини помилкових спрацювань, наявність відомих розкладів робіт або тестових прогонів - усе це збагачує навчальну вибірку для тонкого налаштування порогів і ваг ознак у майбутніх релізах. Паралельно відстежують «здоров'я» даних: пропуски, дублікати, зсуви часових міток, розсинхронізацію edge-буферів і контроль доступності джерел. Такий цикл «моніторинг → діагностика → адаптація → валідація" гарантує, що система зберігає потрібну чутливість до реальних відхилень і водночас залишається керованою в експлуатації на довгій дистанції.

3.3 Впровадження і надійність сервісу детекції

Експлуатаційна стійкість детектора в IoT-мережі визначається не тільки вибором алгоритму, а передусім керованим життєвим циклом: від формування datasets і ознак до серіалізації моделі, версіонування порогів та безпечного виведення у прод. У динамічному середовищі, де змінюються типи пристроїв, прошивки і профілі навантаження, головний ризик - поступовий дрейф якості; отже, конвеєр має забезпечити відтворюваність експериментів, простежуваність артефактів і контроль схеми даних.

Практична організація включає репозиторій із чітким розділенням коду детектора, конвеєра побудови ознак та конфігурацій; артефакти "дані → ознаки → модель → пороги" фіксуються з контрольними сумами й метаданими, а середовища збірки описуються як код для гарантованої відтворюваності. Перевірки якості даних запускаються на вході (часові мітки, діапазони, щільність), а на етапі побудови ознак контролюється відсутність витоку майбутньої інформації та паритет між офлайн/онлайн-трансформаціями. Валідація моделі виконується в часовому розрізі; до експлуатаційних індикаторів, що відразу співвідносяться з практикою, віднесено FP/100 device-

hours і p95 затримки детекції. Кандидати розміщуються в реєстрі моделей із правилами промоції (champion/challenger), тіньовим прогоном та чіткими воротами якості - така організація відповідає загальноприйнятому життєвому циклу MLOps і знижує ризик деградації під час релізів [15].

Механізм оновлення поєднує періодичний retrain за розкладом і подійні тригери: перевищення порогів дрейфу розподілів ознак, падіння експлуатаційних метрик або розсинхронізація онлайн/офлайн. Перед увімкненням нової версії результати порівнюються з поточною моделлю на ковзному вікні останніх тижнів; промоція дозволяється лише за стабільної переваги або принаймні відсутності регресії у визначених метриках.

Впровадження виконують у гібридній топології edge та хмара: поблизу джерела даних працює легкий агент із мінімальним набором перетворень і правил, у центрі - сервіс агрегації, калібрування порогів і кореляції подій. Контракт передбачень фіксується через стабільний API, а сумісність версій забезпечується міграціями схем і версіонуванням конфігурацій. Щоб мінімізувати ризик, застосовуються поступові стратегії оновлення (canary, blue-green) з автоматичними критеріями промоції і миттєвим відкатом. Надійність сервісу формалізується за підходом SRE: для ключових індикаторів якості (частота хибних спрацювань у нормованому вимірі, p95 затримки детекції, свіжість даних, покриття) задаються SLO-цілі, вмикаються сповіщення з відповідними runbook, а швидкість релізів узгоджується з бюджетом помилок (error budget) [30].

Помаранчева п95 це лінія показує , що кількість помилок зменшується система спостережуваності поєднує метричні панелі, журнали та трасування. Окремо відслідковується якість даних: контроль схеми, частки пропусків, а також індикатори дрейфу (на кшталт PSI) для ключових ознак. Виявлені відхилення автоматично створюють завдання на ретюнінг порогів або retrain, а у разі порушення SLO запускають сценарії деградаційного режиму (зменшення чутливості, локальне вимкнення правил, відкат моделі) до відновлення стабільності.

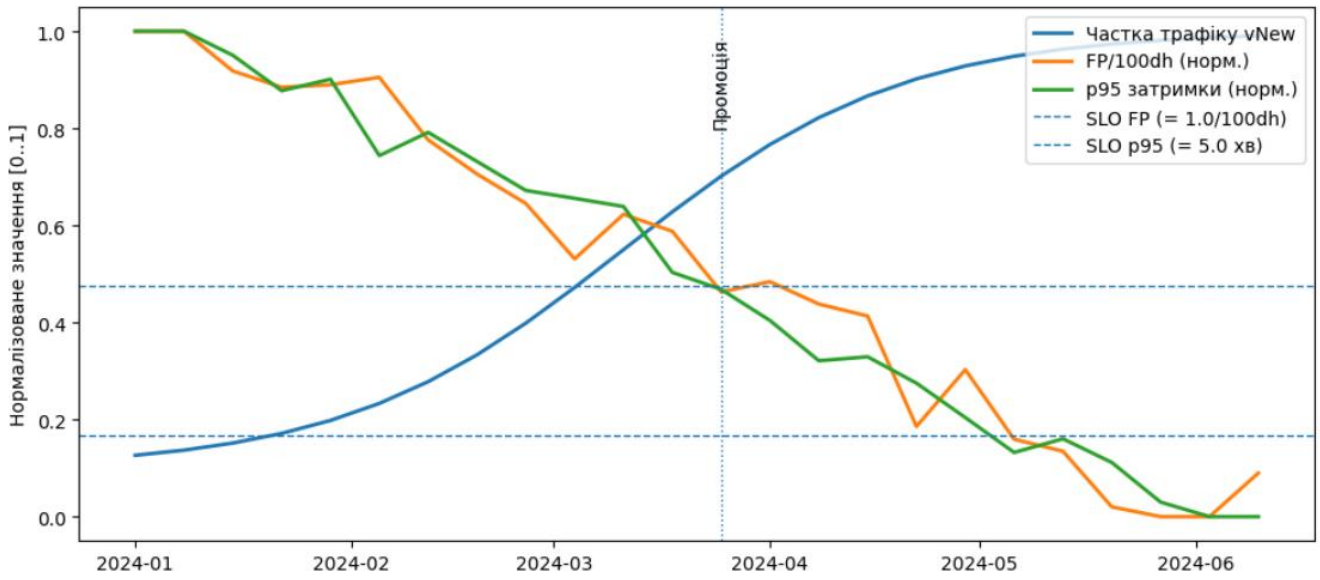


Рис. 3.10 Канарейкове розгортання: частка трафіку нової версії та SLI

Оскільки телеметрія містить чутливі ідентифікатори, застосовується мінімізація і псевдонімізація даних із контрольованими термінами ретенції; доступ здійснюється за ролями з журналюванням усіх адміністративних дій. Ланцюжок постачання захищається підписаними образами та SBOM, секрети зберігаються у сховищі, трафік і сховища шифруються; мережеві політики ізолюють середовища навчання від продакшену.

Масштабування досягається поєднанням autoscaling у хмарі, обмеженням частоти детекції в піках і винесенням частини перетворень на периферію. Для керування вартістю впроваджуються ліміти ресурсів, кешування ознак і профілювання інференсу; планування ємності прив’язується до темпів приросту пристроїв і очікуваного рівня спрацювань. Надійність підсилюється резервуванням черг і сховищ, реплікацією критичних компонентів та регулярним тестуванням сценаріїв аварійного відновлення; коригують пороги й оновлюють інструкції реагування.

Підсумовуючи, інтеграція налагодженого конвеєра даних, спостережуваності та заходів безпеки трансформує запропоновану методику детекції з лабораторного рівня у промислову, забезпечуючи контрольовану

еволюцію моделі, прозору оцінку її якості та відновлюваність сервісу в умовах пікової навантаженості.

ВИСНОВКИ

Дослідження було спрямоване на розроблення та апробацію методів виявлення аномалій у трафіку IoT-мереж із використанням методів машинного навчання. Під час роботи враховано обмежені обчислювальні ресурси вузлів, поширене шифрування прикладного рівня та вимоги до надійності промислового сервісу детекції. Запропоновано цілісний підхід, що поєднує модель трафіку, матрицю ознак на основі метаданих і виробничий MLOps-конвеєр для підготовки даних, навчання та моніторингу моделей.

У ході виконання роботи проаналізовано специфіку IoT-трафіку в корпоративному сегменті та показано, що навіть за умов шифрування корисного навантаження трафіку практично цінну інформацію для детекції дають метадані: часові характеристики, інтенсивність і обсяги обміну, протокольні “підписи”, показники стабільності з’єднань і зміни структури взаємодій між вузлами. Значну увагу приділено підготовці даних, оскільки саме якість консолідації, синхронізація часу, псевдонімізація ідентифікаторів та коректне віконування визначають стабільність подальших висновків. У результаті сформовано узгоджений набір ознак, який дає можливість порівнювати поведінку різних класів IoT-пристроїв без “змішування” режимів роботи сенсорів, камер чи контролерів доступу, а також враховувати сезонність і регулярні цикли навантаження.

Практична апробація підходу підтвердила, що безнаглядні методи можуть бути ефективною базовою лінією для реального середовища, де повна розмітка інцидентів зазвичай відсутня. Обрані принципи це робота з залишковою динамікою, робастне масштабування, порогування за розподілом скорів та перевірка кандидатів на аномалії додатковими індикаторами дозволяють отримувати керовану кількість спрацювань і роблять систему придатною до експлуатації. У підсумку робота показує, що задачу виявлення аномалій у трафіку IoT-мереж можна розв’язувати практично й масштабовано, якщо

поєднати правильну інженерію даних і ознак із керованим контуром розгортання та супроводу моделей.

Подальший розвиток теми доцільно пов’язати з накопиченням зворотного зв’язку від експлуатації (позначення істинних/хибних спрацювань), переходом до напівкерованих підходів, кращою адаптацією до дрейфу та “легітимної новизни” після оновлень пристроїв, а також із поглибленням пояснюваності результатів, щоб спрацювання одразу підказували оператору, що саме змінилось у поведінці вузла чи сегмента і де шукати першопричину.

ПЕРЕЛІК ПОСИЛАНЬ

1. ENISA Threat Landscape 2023 | ENISA. Home | ENISA. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023
2. Ляшенко О. С., Великий І. А., Знайдюк В. Г., Журило О. Д. Модель та методи виявлення широкомасштабної атаки в середовищі IoT. 2024. URL: https://www.researchgate.net/publication/378451177_MODEL_TA_METODI_VIAVLENNA_SIROKOMASSTABNOI_ATAKI_V_SEREDOVISI_IOT.
3. Кібербезпека. Методи захисту інформації в технологіях IoT. 2025. URL: https://www.csecurity.kubg.edu.ua/index.php/journal/article/view/705.
4. Internet Of Things (IoT): Origins, Embedded Technologies, Smart. URL: https://www.researchgate.net/publication/381689333_Internet_Of_Things_IOT_Origins_Embedded_Technologies_Smart_Applications_and_its_Growth_in_the_Last_Decade.
5. Ashton K. That ‘Internet of Things’ Thing. RFID Journal. 2009. URL: https://www.rfidjournal.com/articles/view?4986.
6. IoT Trends 2024. N-iX. URL: https://www.n-ix.com/iot-trends.
7. Fog computing: Principles, architectures, and applications. Wikipedia. URL: https://en.wikipedia.org/wiki/Fog_computing.
8. Dauda A., Flauzac O., Nolot F. A Survey on IoT Application Architectures. Sensors. 2024. URL: https://doi.org/10.3390/s24165320.
9. RFC 7452. Architectural Considerations in Smart Object Networking. URL: https://www.rfc-editor.org/rfc/rfc7452.
10. ENISA. Guidelines for Securing the Internet of Things. 2020. URL: https://www.enisa.europa.eu/publications/guidelines-for-securing-the-internet-of-things.
11. NISTIR 8228. Considerations for Managing IoT Cybersecurity and Privacy Risks. URL: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8228.pdf.
12. Antonakakis M. et al. Understanding the Mirai Botnet. USENIX Security Symposium. 2017. URL:

https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf.

13. Hyndman R., Athanasopoulos G. Forecasting: Principles and Practice. Chapter “ARIMA models”. URL: https://otexts.com/fpp3/arima.html.

14. Machine learning. Wikipedia. URL: https://en.wikipedia.org/wiki/Machine_learning.

15. ENISA. Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures. URL: https://www.enisa.europa.eu/sites/default/files/publications/WP2017%20O-1-1-2%201%20Baseline%20Security%20Recommendations%20for%20IoT%20in%20the%20context%20of%20CII_FINAL.pdf.

16. NISTIR 8259A. IoT Device Cybersecurity Capability Core Baseline. URL: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259A.pdf.

17. Gama J., Žliobaitė I., Bifet A., Pechenizkiy M., Bouchachia A. A Survey on Concept Drift Adaptation. ACM Computing Surveys. 2014. URL: https://dl.acm.org/doi/10.1145/2523813.

18. Mazhar M. T. et al. Characterizing Smart Home IoT Traffic in the Wild. 2020. URL: https://arxiv.org/abs/2001.08288.

19. Sivanathan A. et al. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. IEEE Transactions on Mobile Computing. 2019. URL: https://www2.ee.unsw.edu.au/~vijay/pubs/jrnl/18tmc.pdf.

20. RFC 7011. Specification of the IP Flow Information Export (IPFIX) Protocol. URL: https://www.rfc-editor.org/rfc/rfc7011.

21. ITU-T Recommendation Y.1540. Network performance objectives for IP-based services. URL: https://www.itu.int/rec/T-REC-Y.1540.

22. Barabási A.-L. Network Science (online edition). URL: http://networksciencebook.com.

23. Підгорний П., Лаврик Т. Виявлення аномалій у зашифрованому мережевому трафіку за допомогою глибокого навчання. Кібербезпека: освіта,

наука, техніка. 2025. URL: https://csecurity.kubg.edu.ua/index.php/journal/article/view/897.

24. Goldstein A., Uchida S. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. PLOS ONE. 2016. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0152173.

25. Google SRE Book. Service Level Objectives (SLI/SLO/SLAs). URL: https://sre.google/sre-book/service-level-objectives/.

26. Hyndman R., Athanasopoulos G. Forecasting: Principles and Practice. Exponential smoothing. URL: https://otexts.com/fpp3/expsmooth.html.

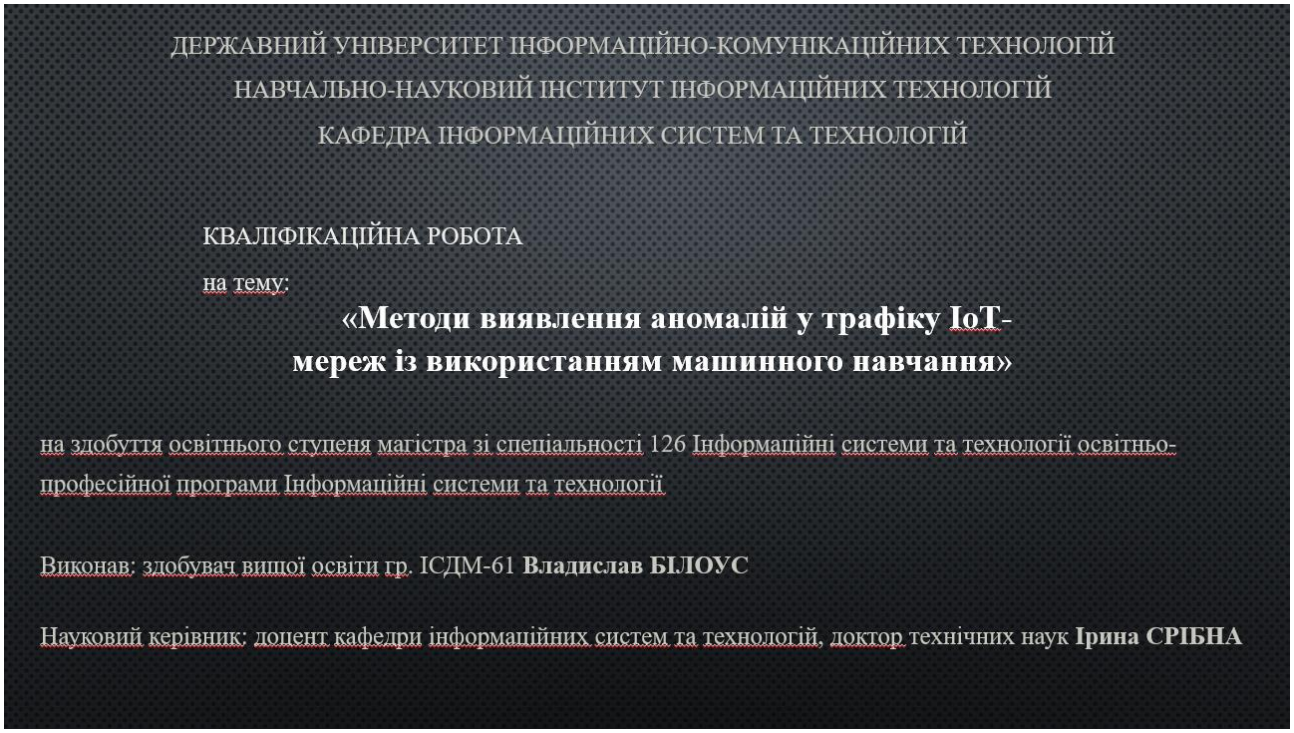
27. scikit-learn. RobustScaler. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html.

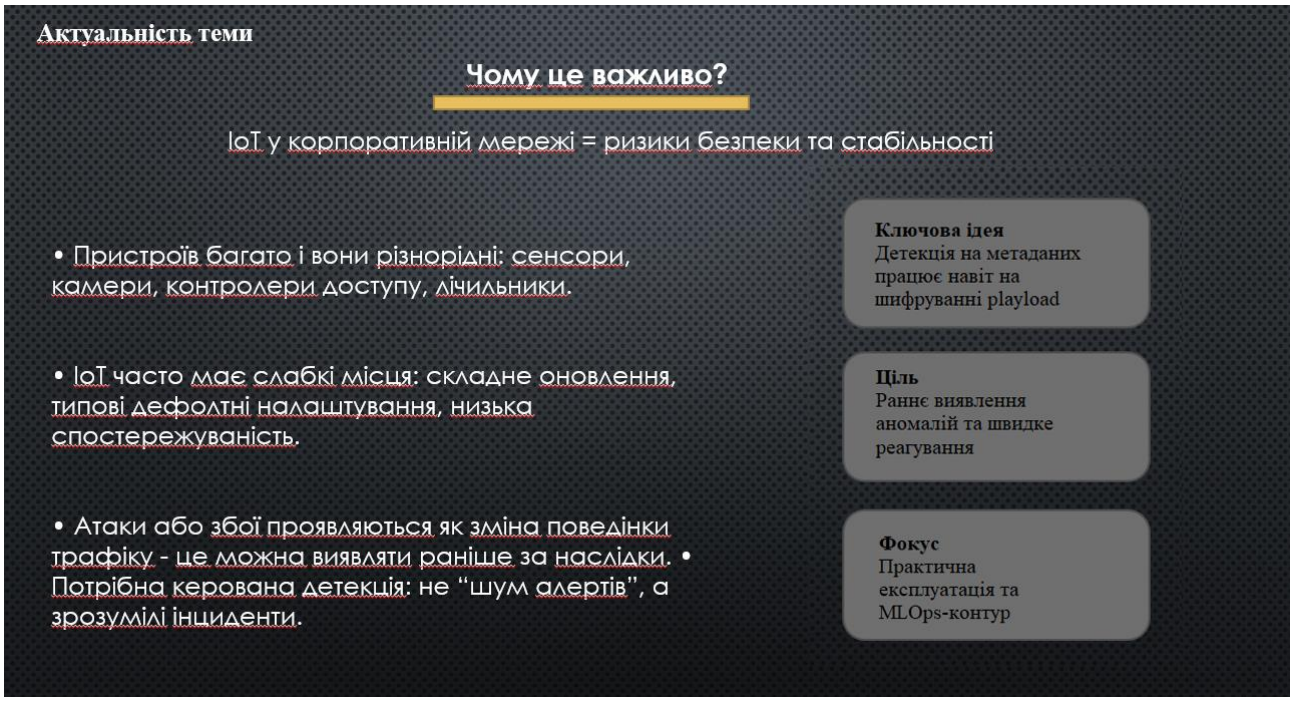
28. ADTK. Anomaly Detection Toolkit. URL: https://adtk.readthedocs.io/en/stable/.

29. scikit-learn. Permutation feature importance. URL: https://scikit-learn.org/stable/modules/permutation_importance.html.

30. Zaharia M. et al. Accelerating the Machine Learning Lifecycle with MLflow. 2018. URL: https://people.eecs.berkeley.edu/~matei/papers/2018/ieee_mlflow.pdf.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)





Постановка задачі

Мета, об'єкт, предмет, завдання

Мета роботи

Розробити та апробувати методи виявлення аномалій
у трафіку IoT-мереж із використанням машинного навчання
та виробничого підходу MLOps.

Об'єкт дослідження	Трафік та телеметрія IoT-сегмента корпоративної мережі.
Предмет дослідження	Методи детекції аномалій за метаданими трафіку та їхня експлуатація в контурі MLOps (пороги, дрейф, якість сервісу).

Основні завдання

- Описати модель трафіку IoT та вимоги до сервісу детекції
- Сформувати матрицю ознак із метаданих та виконати передобробку
- Порівняти статистичні та ML-підходи (зокрема Isolation Forest)
- Запропонувати контур моніторингу та адаптації до дрейфу

Обмеження реального середовища

Чому "просто навчити модель" недостатньо?

Шифрування
Часто недоступний
вміст пакетів (payload)
→ ставка на метадані

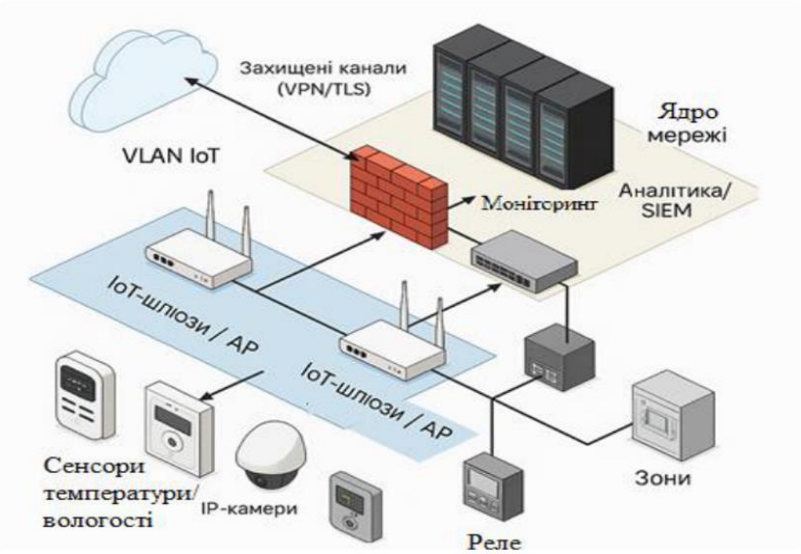
Мало розмітки
Інциденти рідкісні
та не завжди задокументовані
→ безнаглядні методи

Надійність сервісу
Керована кількість спрацювань,
агрегація у інциденти,
моніторинг

Ресурси
Обмеження edge-вузлів
та каналів зв'язку

Архітектура IoT-сегмента

Сегментація, точки контролю, канали зв'язку



Структура IoT-сегмента корпоративної мережі (рис. 2.4)

Збір та консолідація даних

NetFlow/IPFIX, логи шлюзів, MQTT та центральна аналітика

Джерела даних

- Агреговані записи мережевих потоків (NetFlow/IPFIX або еквівалент)
- Журнали подій IoT-шлюзів, точок доступу та міжмережевих екранів
- Логи брокера MQTT та суміжних прикладних сервісів
- Дворічний період спостереження (2023–2024) для сезонності та змін політик

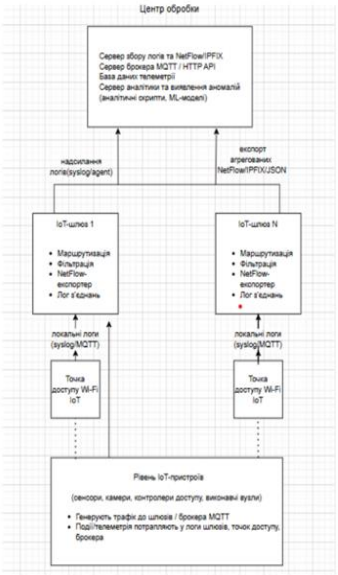


Схема інфраструктури збору та консолідації (рис. 2.2)

Класифікація аномалій

Як систематизуються відхилення у трафіку IoT

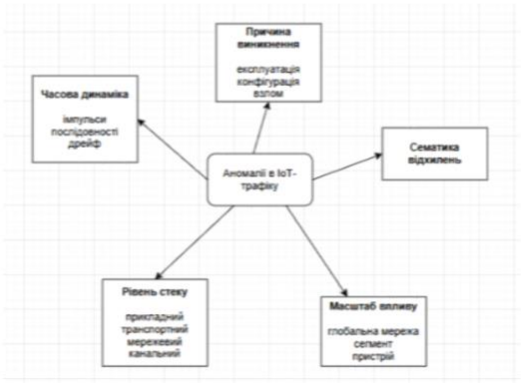


Рис. 1.2 Узагальнена класифікація аномалій у трафіку IoT

Зрізи класифікації

- Часова динаміка: імпульси / послідовності / дрейф
- Рівень стеку: канальний / мережевий / транспортний / прикладний
- Семантика: відхилення у поведінці та протоколах
- Причина: експлуатація / конфігурація / злом
- Масштаб: пристрій / сегмент / глобальна мережа

Базові статистичні підходи

Гістограми та ARIMA як "карта здоров'я" метрик

Ідея

- Гістограми — швидка перевірка "хвостів" та змін форми розподілу
- ARIMA — моделювання сезонності/тренду та аналіз залишків
- Комбінація допомагає відрізняти аномалії від природної динаміки

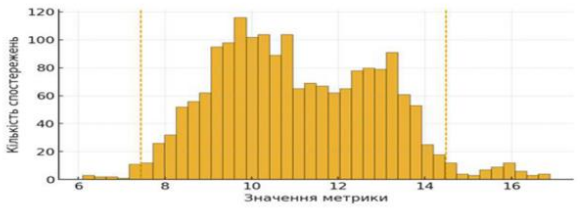
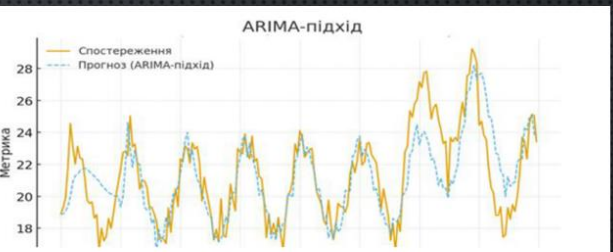


Рис. 1.3 Гістограма метрики трафіку IoT

Приклад гістограми метрики (рис. 1.3)



ARIMA: факт vs прогноз (рис. 1.4)

Isolation Forest

Ключове

- Ненаглядний (unsupervised) детектор — працює без розмітки атак
- Дає скор аномальності для кожного інтервалу/об'єкта
- Поріг підбирається так, щоб контролювати хибні спрацювання
- Далі — постобробка: агрегація, контекст, стабілізація порога

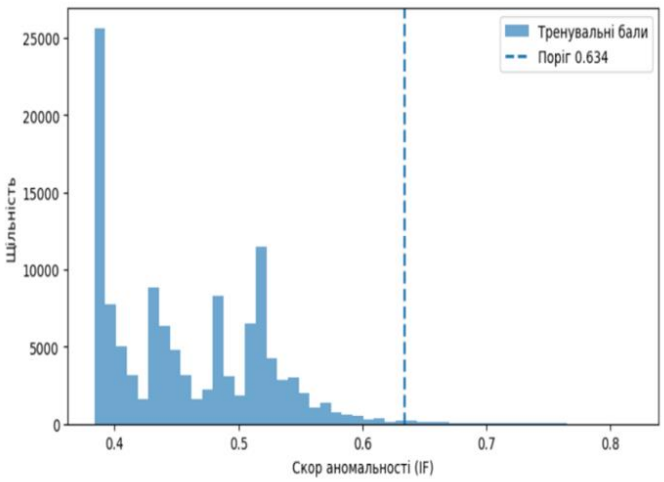


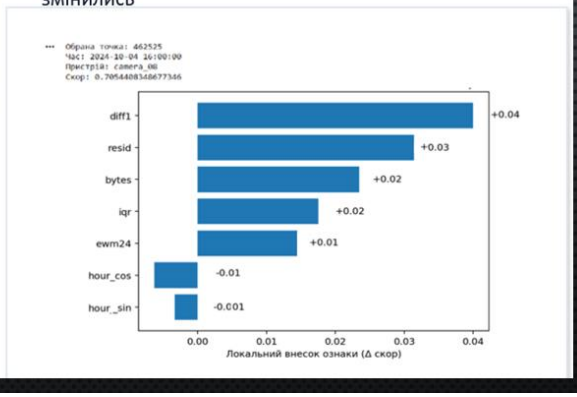
Рис. 3.3 Результати Isolation Forest із порогом

Пояснюваність та дрейф

Що вплинуло на інцидент і як контролюється стабільність

Пояснюваність

- Локальні внески ознак показують: "що саме зламалось" у патерні
- Дає оператору контекст: чому скор виріс і які метадані змінились



Контроль дрейфу (PSI)

- Моніторинг розподілів ознак/скорів у часі
- Пороги PSI сигналізують про концептуальні зміни
- За потреби — перекалібровка порога або перенавчання

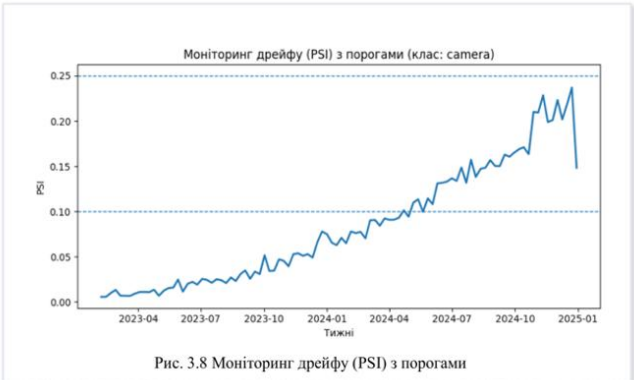
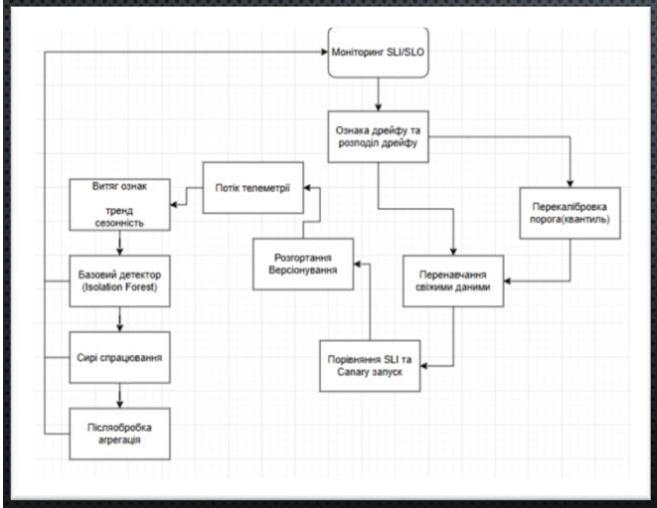


Рис. 3.8 Моніторинг дрейфу (PSI) з порогами

Впровадження та MLOps

Моніторинг якості, адаптація, версіонування



Практичні принципи

- SLI/SLO як “рамка” для прийняття рішень про зміни
- Перекалібровка порога (квантиль) на «свіжому» вікні
- Перенавчання на нових даних із захистом від інцидентів
- Запуски shadow/canary перед повним перемиканням
- Версіонування моделей, порогів і контрольних звітів

Апробації

Білоус В.В., Білошицький М.П., Срібна І.М., Данильченко В.М. “МОДЕЛІ МАШИННОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ ТА ЗАХИСТУ ДАНИХ У СИСТЕМАХ МЕДИЧНОГО АДМІНІСТРУВАННЯ НА ОСНОВІ ІОТ”.

Стаття подана до друку у загальногалузевий науково-виробничий журнал “Зв’язок” - м.Київ

Білоус В.В. “ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ У ПОБУТІ І ПРОМИСЛОВОСТІ” та “БАГАТОРІВНЕВА АІ ТА ML-АРХІТЕКТУРА ПОБУТУ ТА ПРОМИСЛОВОСТІ”

Теза у ІІІ Всеукраїнська науково-технічна конференція "Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу" – м. Київ 11 листопада 2025р.

Дякую за увагу!